PRACTICAL MODEL BASED SOFTWARE TESTING

A MASTER'S THESIS

IN

INFORMATION TECHNOLOGY SERVICE MANAGEMENT

ATILIM UNIVERSITY

BY

RENGİN ÇOŞKUN

MAY 2015

PRACTICAL MODEL BASED SOFTWARE TESTING

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

RENGİN ÇOŞKUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCES OF PHILOSOPHY

IN

THE DEPARTMENT OF INFORMATION TECHNOLOGY SERVICE MANAGEMENT

MAY 2015

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

_____

Prof. Dr. K. İbrahim AKMAN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. Murat KOYUNCU

Head of Department

This is to certify that we have read the thesis "Practical Model Based Software Testing" submitted by "Rengin ÇOŞKUN" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Asst. Prof. Dr. Deepti MISHRA

Supervisor

Examining Committee Members

Assoc. Prof. Dr. Korhan Levent Ertürk _____

Asst. Prof. Dr. Meltem Eryılmaz _____

Asst. Prof. Dr. Deepti Mishra _____

Date: 12.05.2015

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Rengin ÇOŞKUN

Signature:

ABSTRACT

PRACTICAL MODEL BASED SOFTWARE TESTING

Çoşkun, Rengin

M.S., Information Technology Service Management Department

Supervisor: Asst.Prof.Dr. Deepti Mishra

May 2015, 172 pages

In this thesis, it is proposed a new approach using both manual and automated testing for MBT. The goal is to find successful test cases. Many different research papers are studied and UML sequence diagram examples of these research papers as a model are used in this thesis. Two new algorithms are written for manual testing. It is designed a new tree technique using depth first search path algorithm to find test path. It is written a new java junit testing code to find test paths automatically in automated testing. It is used same operation for both manual and automated testing and also, it is examined these two methods with same sequence diagram examples. However, effective results are not achieved because of determination of same results and errors in automated testing. Therefore, automated testing (java junit testing) code is improved using different methods. It is compared manual, improved automated method and research papers methods using same examples. Finally, it is obtained more successful test paths and test cases from the improved automated testing method according to the manual and research papers testing methods.

Keywords: Model Based Testing, UML Sequence Diagram, Algorithm, Intermediate Model, Depth First Search, JUnit, XML, Test Paths, Test Cases.

ÖZ

PRATİK MODEL TABANLI YAZILIM TESTİ

Çoşkun, Rengin

Yüksek Lisans, Bilgi Teknolojileri ve Hizmet Yönetimi Bölümü

Tez Yöneticisi: Yrd.Doç.Dr. Deepti Mishra

Mayıs 2015, 172 sayfa

Bu tezde, model tabanlı test için, hem otomatik hem de manuel test kullanarak yeni bir yaklaşım üretilir. Amaç, başarılı test durumları bulmaktır. Bu tezde, birçok farklı araştırma makalesiyle çalışıldı ve bir model olarak bu araştırma makalelerinin sıralı diyagram örnekleri kullanıldı. Manuel test için iki yeni algoritma yazıldı. Test yollarını bulmak için, derinlik öncelikli arama yol algoritması kullanılarak yeni bir ağaç tekniği tasarlandı. Otomatik testte, test yollarını otomatik olarak bulmak için, yeni bir java birim test (junit) kodu yazıldı. Hem otomatik, hem de manuel test için, aynı işlem kullanıldı ve ayrıca aynı sıralı diyagram örnekleri ile bu iki metod test edildi. Fakat, aynı sonuçların saptanması ve otomatik testteki hatalardan dolayı, etkili sonuçlar elde edilemedi. Dolayısıyla, farklı metodlar kullanılarak, otomatik test kodu (java birim test) geliştirildi. Aynı örnekler kullanılarak, manuel, gelişmiş otomatik test metodu ve araştırma makalelerindeki metodlar karşılaştırıldı. Sonuçta, araştırma makaleleri ve manuel test metodlarına göre, geliştirilmiş otomatik test metodundan daha başarılı test yolları ve test durumları elde edildi.

Anahtar Kelimeler: Model Tabanlı Test, UML Sıra Diyagramı, Algoritma, Ara Model, Derinlik Öncelikli Arama, JUnit, XML, Test Yolları, Test Durumları.

To My Parents

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Asst.Prof.Dr. Deepti Mishra for her guidance, help and insight throughout the research. Thanks also, to my parents, I offer sincere thanks for their continuous support and patience during this period.

TABLE OF CONTENT

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

MBT - Model Based Testing

UML - Unified Modeling Language

DFS - Depth First Search Algorithm

BFS - Breath First Search Algorithm

JUnit - Java Unit Testing

GA - Genetic Algoritm

SG - Sequence Graph

SDG - Sequence Dependency Graph

EFSM - Extended Finite State Machine

SDT - Sequence Dependency Table

SUT - System Under Testing

SQL - Structured Query Language

XML - Extensible Markup Language

GUI - Graphical User Interface

CHAPTER I

INTRODUCTION

1.1 Statement of the Problem

Software tests have an increasingly effective and important role in terms of whole system in software systems. Also, it is necessary in terms of some criterias of the system such as successful, quality, cost, availability etc. The problem of the thesis, large systems has very complex structures and also finding of test cases based on a model are a difficult step in testing. Therefore, finding of test cases need more time and effort. Tools are very practical to find test cases in automated methods. Also, automated testing can be done by improving source code for software testing. Although manual testing method has some advantages, this method is more difficult than automated methods. Also, manual testing is more risky than automated testing. However, researchers prefer manual or automated testing methods according to their studies.

1.2 Scope and Outline of the Thesis

The aim of this thesis is to propose a solution for the above problem. There are lots of approaches and studies about this issue. Many researchers propose a new method or an algorithm, and find test cases of the system with their methods and algorithms. On the other hand, some researchers use tools that are existing. The purpose of this thesis is to support MBT with a new technique and to prove which testing method is better. In this thesis, it is used two general testing methods with a new approach. Both manual and automated method are used for finding good test paths and also, test cases generation. In manual testing, it is produced a new algorithm based on a sequence diagram model. Then, a new technique is proposed according to the new

algorithm. In automated testing, it is used the same operation with manual testing because of showing whether test paths results same or different in terms of both manual and automated testing. The aim is whether test paths results are the same or test paths results include errors, manuel or automated testing should be improved.

In **chapter 1**, what model based testing is and how this testing is used with UML sequence diagram are explained. Also, scope and statement of the thesis are defined. It will be mentioned about two general methods that are manual and automated testing in MBT. It will be explained the produced technique or method, algorithms, codes and tools in **chapter 2**. In **chapter 3**, it is shown which methods will be used and how the methods will be used. Furthermore, it will be explained test paths and test cases of the proposed approach with examples. In **chapter 4,** the proposed approach will be improved, and the proposed approach results will be compared with manual method results and research papers methods results. It is proved the success of improved test methods results. The final step **chapter 5** is the conclusion of the thesis. It will be explained all results of the thesis.

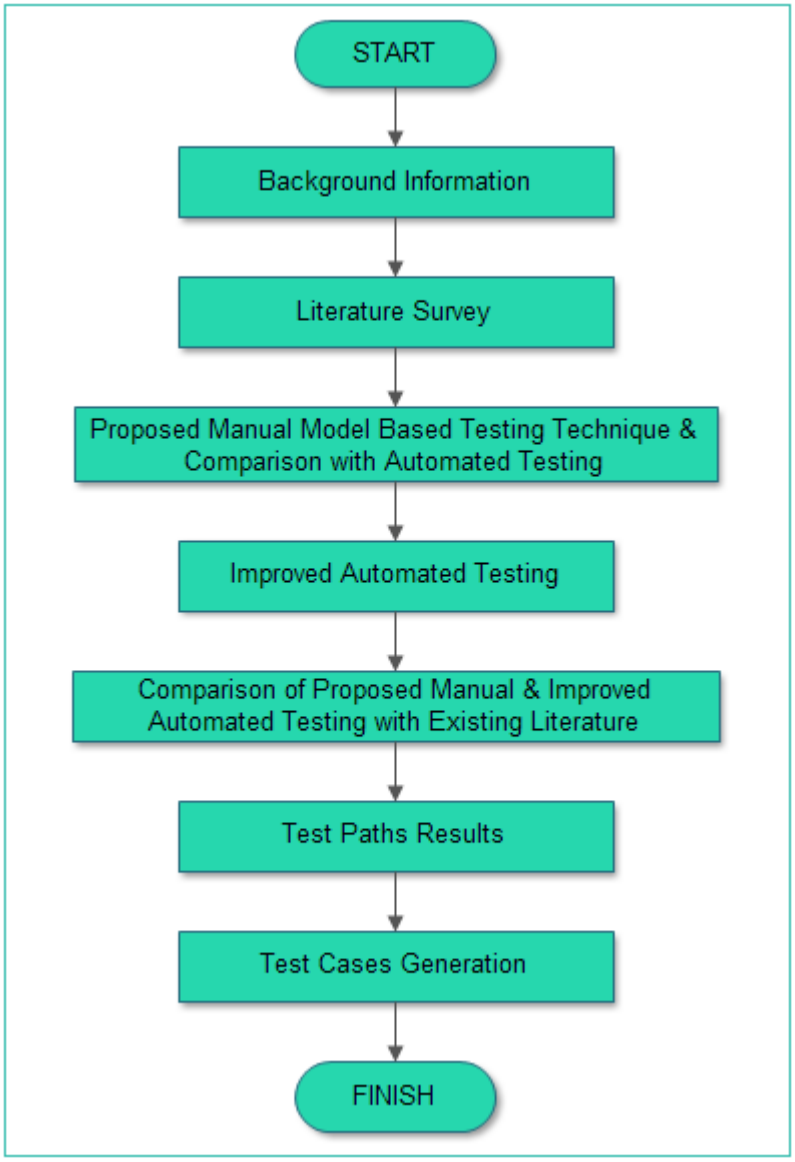


Figure 1: Flow Diagram of the Outline of the thesis

1.3 Software Testing

Software testing is a testing model of system under testing. System under testing is a phase of maturity of the software testing for correct operations. Software testing executes all process of the whole system and also this testing provides finding errors. Software testing is based on validation and verification processes. The validation process involves requirement documents which describe what the software is supposed to do, whereas, the verification process determines whether the software is correct with regard to its design. [22]

1.3.1 Software Testing Methods

Software testing methods can be categorized as level of testing and method of testing.

➢ ***The level of Testing*** [22]

- *Function/Unit testing* is a unit program code of an software system.
- *Module testing* is whole modules or components of software system.
- *Integration testing* assesses the software about its subsystem design.
- *System testing* is an architectural design testing of the complete system.
- *Acceptance testing* is measurement of the system satisfies performed. Acceptance criterias that operational, process and business requirements are very important for this testing.

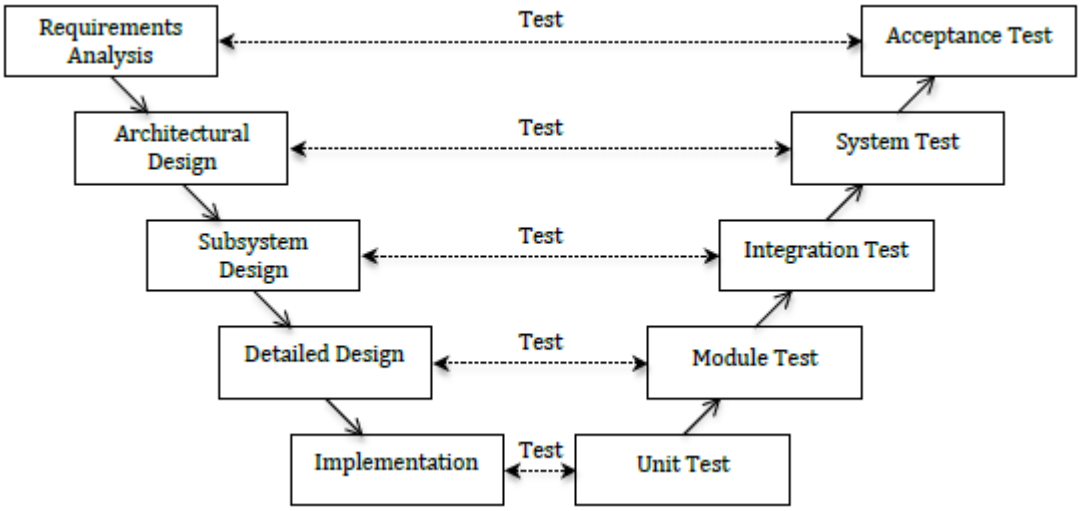


Figure 2: Software development activities and testing levels - the "V-Model" as in [22]

➢ ***The method of Testing***

- ***White-box*** *testing* is the actual source code for software. Tests design code of system with various algorithms and some structures of the source code. This type of testing is also referred to as structural testing. As in [22]
- ***Black-box*** *testing* is a system-under-test that is considered as inputs and the outputs of the system. Also, this testing is referred to as functional testing. As in [22]

1.4 Model Based Testing

The model-based testing is defined as a kind of test application where test cases are generated based on a model to perform system testing. Model-based testing is a one of the software testing where test cases are derived from a model as whole or part. This testing can describe some functional aspects of the system under test (SUT). The test cases derived from models are a process of functional and structural tests of the model. MBT is a black box testing method that test cases are occured automatically using models. In other words, model based testing is a behavior model. The behavioral model specifies system requirements or conditions with the system under testing. This behavior models include requirements of system such as inputs, outputs and condition of the system (pre-condition and post-condition). A behavior model explains test cases in a model based testing.

According to [23], there are four main approaches known as model-based testing:

1. Generating of test input data from a domain model
2. Generating of test cases from an environment model
3. Generating of test cases with oracles from a behaviour model
4. Generating of test scripts from abstract test

Generating of test input data from a domain model is related to input data of model in model-based testing. The test input information involves combination of a subset of input data values to produce test values.

Generating of test cases from an environment model uses a different model to describe the expected environment. These environment models can generate sequences with the system under test. This environment models can not predict the output values because the environment models are not behaviour models. Therfore, these environment models are very difficult to specify whether any testing has passed or failed.

Generating of test cases with oracles from a behaviour model includes oracle information for executable test cases. These are expected output values or automated check on the actual output values of the system under testing. This model is very complex models of model based testing. This models has feedback, however do not check the results. This model should identify behaviour relationship between inputs and outputs of the system under testing.

Generating of test scripts from abstract test gives abstract description of a test case, such as a UML sequence diagram, and this testing is related to transforming of

abstract test case for low-level test script. In this approach, the model includes information about the structure and API (Application Programming Interface) of the system under testing.

1.4.1 Model Based Testing Process

Figure 3 shows steps of MBT processes. According to [22], MBT includes five processes.

The first step is related to phases of model. A model can be an environment of the system-under test, and it is not complex than the actual implementation. The model includes requirements, design specifications or specification documents and test plan of the system.

The second step is test case generation from abstract models. This test cases is related to key aspects for testing and also many details. Also, this process follows model coverage criteria and keeps requirements for the test cases.

The third step is called implementation of tests. This step focuses on transformation of the abstract test cases with various templates and mappings. This process is supported with tools by programmers.

The fourth step is related to test execution and it needs to be adapted to execute. This process is created by using a tool commonly called as test adapter. The adapter adds specific information to the tests and tests can be executed either in online or in offline mode. This test results are directed by test adapter tool.

The fifth step is the analysis phases of the test results. This testing phase is similar to traditional test analysis. Also, this testing phase is related to test case failure and a fault in the system.

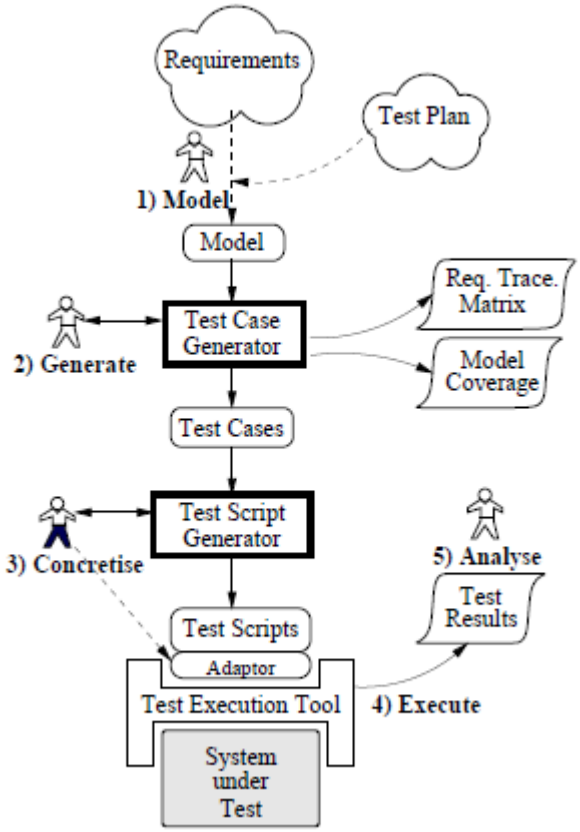


Figure 3: Model Based Testing Process as in [12]

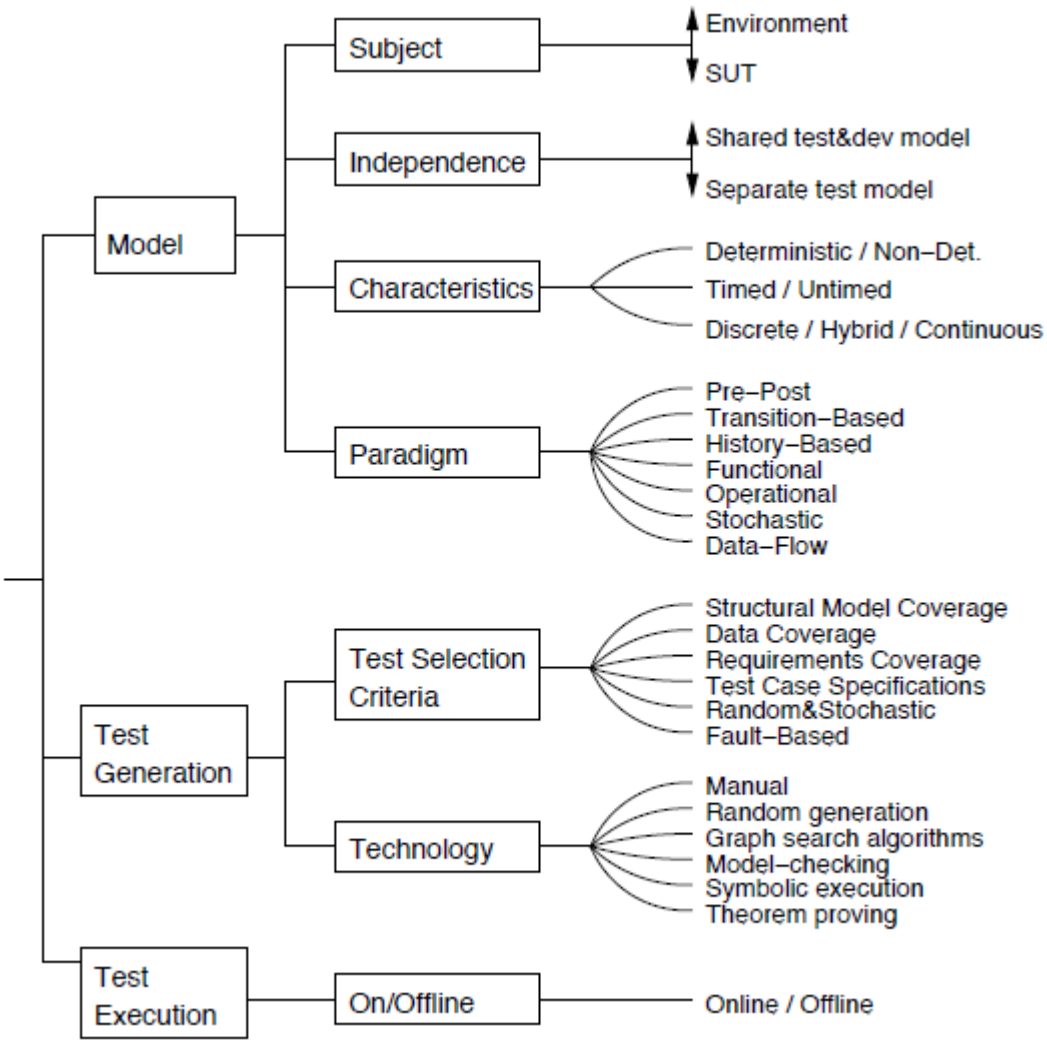


Figure 4: Taxonomy of Model-Based Testing as in [13]

1.4.2 Model Based Testing Advantages and Disadvantages

Advantages of MBT

✓ MBT provides high level automation for testing.

✓ MBT provides improvement of product quality in testing system. Also, MBT is a systematic testing to help understanding the behavior of the system. MBT provides improving quality with this property.

✓ Detail testing can find test cases with MBT.

✓ Test case changing can be managed and followed with MBT.

✓ MBT provides direct connection with the model so, model requirements can be updated.

✓ MBT provides high effective test coverage, and this testing is applied for all the functionalities of the system.

✓ MBT reduces cost of testing. This testing includes fault detection and finding error, so cost reduces as automatically.

✓ MBT reduces time of testing, and avoids from the system complexity. From this property, it can be reached testing results shortly with MBT.

✓ MBT provides traceability for model requirements and test cases.

Disadvantages of MBT

➢ If model complexity increases, then MBT becomes more difficult.

➢ In MBT, adequate skill is necessary to construct an effective model for abstract and design model.

➢ If any changing occurs in the model, then a different test set can occur.

➢ MBT can not quarantee to find test cases differences from the model.

1.5 Flow Diagram

Flowchart or flow diagram is a diagram that uses graphic symbols, represents algorithms, and steps of all process order by their connecting each other. In this thesis, it is used flow diagram to represent all process with the proposed approach. Flowchart is an easy diagram in terms of both drawing and understanding. It is tried to show all steps of the system in general with this flowchart.

CHAPTER II

BACKGROUND INFORMATION AND LITERATURE SURVEY

Testing has a significance role for software systems. Large systems of software have very complex structures, and finding test cases are a difficult step for testing. Finding test cases is a very difficult process in terms of time, effort and cost. Many researchers use lots of different approaches, and they use different methods and algorithms in model based testing as manual or automatic technique. Besides, researchers prefer existing tools in some studies. Tools have many good properties because they run automatically. Although manual testing is more difficult than automated testing, many researchers prefer manual testing for their studies. The most important reason of this situation is that many tools can not meet the requirements of the studies.

In chapter 2, it will be given some information about contents of this thesis. Manual and automated testing are defined as test automation in MBT. Both manual and automated testing are included in this study. Also, it is explained techniques, algorithms and tools in this approach. UML sequence diagram is used as a model in MBT and it is worked with sequence diagram in different areas. UML Sequence diagrams are very common models in terms of model based testing approach. Sequence diagrams in MBT provide test path sequences with various intermediate models, graphs, tables and flow graphs. An another method for sequence diagrams in MBT is that various algorithms provide generation of test cases with unit testing. Sequence diagrams can transform from model-to-model. xUnit is a collective name of unit testing frameworks such as JUnit, SUnit etc. Test cases can find with these test code methods. Furthermore, complexity and test coverage criteria of test cases can be defined with MBT.

2.1 Testing Types

Manual and automated tests are the most common testing types in the software. These testing types cover some specific testing methods such as black box testing, white box testing, integration testing, system testing, performance testing, load testing, unit testing and acceptance testing. Some of these testing are very suitable for manual testing, but some testing are better for automated testing. **Manual testing** is executed by human without using any automated tools or scripts. A test plan document or test scenario is prepared by testers for manual testing. **Automated testing** is executed with automated tools, scripts and software. Testers can use a suitable programming language for system and also, testers can improve system for any operation in automated testing. Manual and automated tests have some pros and cons.

2.1.1. Manual Testing Pros & Cons

PROS

1. The most important feature is flexibility.

2. More likely to find real user issues [15]

3. Short-term cost is lower. [15]

4. Manual testing will be useful when the test case only needs to run once or twice [17]

5. To execute the test cases every time tester requires the same amount of time [17]

6. Manual tests are very much helpful in UI testing [17]

7. To execute the Build Verification Testing (BVT) is very mundane and tiresome in manual testing [17]

8. To execute the test cases first time using manual testing will be very much useful. But it is not sure that it will catch the regression defects under frequently changing requirements [17]

CONS

1. Certain tasks are difficult to do manually [15]

2. Manual tests are unstimulating.

3. Can’t reuse manual tests [15]

4. Manual tests are less reliable.

5. Non- programmable [16]

6. Time consuming and tedious [16]

7. Using manual testing,testing on different machine with different OS platform combination is not possible, concurrently. To execute such task different testers are required [17]

2.1.2. Automated Testing Pros & Cons

PROS

1. Automated testing is more reliable.

2. Automated tests are very fast.

3. Less investment in human resources [16]

4. Programmable [16]

5. Can execute easily and effectively.

6. Can be cost effective [15]

7. More interesting [15]

8. Results can see quickly.

9. Automation testing will be useful to execute the set of test cases frequently [17]

10. After making Automation test suites, fewer testers required to execute the test cases [17]

11. Automation testing can also be done on different machine with different OS platform combination, concurrently [17]

12. Some time it is not helpful in UI testing [17]

13. Automation testing is very useful for automating the Build Verification Testing (BVT) & it is not mundane and tiresome [17]

14. Automation testing is very much helpful regressions in testing where code changes frequently [17]

CONS

1. Cost of tools can be expensive.

2. Use of tools can take time consuming.

3. Tools can include various limitations.

2.2 Test Techniques

It is used two test techniques in this thesis. First technique is manual testing and second technique is automatic testing in the MBT. These techniques are explained with following subtitles.

2.2.1 Manual Test Technique

In manual testing, it is preferred intermediate model such as formal models, graphs, trees, and other hybrid types to create test case generation. These intermediate model approaches are broadly classified into four major categories: specification based testing, graph theoretic testing, heuristic testing, and direct UML specification processing. [18] However, it is used only tree approach among intermediate model structures in this paper. Tree model approach depends on graph theoretic testing methods.

Graph theory is an area of mathematics that helps deriving test cases from design models in many different ways. [18] Graphs are the structures that are used the most for abstract test cases. UML models represent graphical structures of a system properties. However, UML models are not used directly to create test cases so, UML models need some intermediate models such as trees and graphs for test case generation.

In this study, two new algorithms are proposed using UML sequence diagram model for manual test technique in MBT. First algorithm is necessary to develop a tree. The tree is created from the first algorithm using UML sequence diagram. Then, test paths are read from the tree using second algorithm. This second algorithm includes DFS algorithm. The reasons for choosing of DFS algorithms is that this algorithm focuses on to generate a set of test scenarios or basic test paths. Finally, it is derived a set of test paths and also test cases or scenarios from this test method.

2.2.1.1 Tree Method

A tree is a connected acyclic simple graph. An alternate sequence of vertices and edges beginning at the root and ending in a leaf node is of interest for a scenario representation. [18]

Lots of researhers benefit from graphs, but it is used a tree technique instead of graph. Also, trees are an undirected graphs. Every tree consists of nodes and edges. In this study, it is preferred tree, becuase tree is a very practical and easy method in terms of manual testing. Also, this method is very available to find test paths immediatelly.

➢ **Theorem:**

Tree is T = (V, E)

- V = nodes. (number of message)
- E = edges between pairs of nodes. (message dependency)
- Captures messages relationship between objects.

Let T be an tree on n nodes that message numbers.

- T is connected.
- T does not contain a cycle.

2.2.1.2 Depth First Search Algorithm

In manual testing, an algorithm is generated to find test paths in test case generation. The algorithm includes also DFS algorithm. It is preferred DFS algorithm, because DFS algorithm is very practical and common for tree tehnique in the software engineering. The other reason of using DFS algorithm is that DFS algorithm is more suitable than other existing algorithms for the proposed tree technique.

Depth first search is one of the simplest algorithms for searching a graph, and this algorithm uses also for tree structures. This algorithm controls all neighboor of a node, and provides searching deeper. DFS discovers all vertices. If any undiscovered vertices remain, then depth-first search selects one of them as a new source, and it repeats the search from that source. The algorithm repeats this entire process until it has discovered every vertex.[14]

```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)

DFS-VISIT(G, u)
1  time = time + 1              // white vertex u has just been discovered
2  u.d = time
3  u.color = GRAY
4  for each v ∈ G.Adj[u]        // explore edge (u, v)
5      if v.color == WHITE
6          v.π = u
7          DFS-VISIT(G, v)
8  u.color = BLACK              // blacken u; it is finished
9  time = time + 1
10 u.f = time
```

Figure 5: Illustrates the progress of DFS on the graph as in [14]

2.2.2 Automatic Test Technique

In this thesis, it is used some tools to generate set of test paths for automatic test technique in MBT. It is used java as programming language. Therefore, new codes in eclipse are written. Also, junit for testing is used. Junit is better than the other unit testing frameworks so, junit is preferred.

2.2.2.1 Supporting Tool

It is used some open source tools for this approach in MBT. Especially, these tools are necessary for automatic testing. An integrated system is used for automatic testing. It is prefered java as programming language because java is a very common, usable and preferable programming language in the software engineering. In this thesis, it is used eclipse for java language and also, plug-in tools for eclipse platform is used. For automated testing, test paths and test cases are generated using unit testing.

Eclipse is a very common, easy and useful tool for software engineering. It is a unique model for Open Source Development. Also, the programming language of eclipse is Java, C and C++ so, many users prefer eclipse with this feature. The Eclipse is a non-commercial tool. Also, this tool can be integrated with other tools easily. Eclipse can run both Windows and Unix platforms.

- **Junit Testing Features**

Unit testing is used for automatic testing. JUnit is an open source framework for java programming language. Junit is an automatic language that write and run. Junit is a unit test case part for java code. Also, junit provides achievment of all desired test results quickly. Unit test case includes two test cases results as positive and negative. These two test cases results provides usable code for testing requirements.

xUnit is general name of various unit testing frameworks. It is compared most popular unit testing framewors in the following Table 1. It is prefered junit testing as the most popular unit testing. Junit is better than the other unit testing frameworks because of its good properties. Also, junit provides an automatic testing for developing codes and applications. It is very useful and easy for many programmers. It provides easy understanding of code, and it is very fast for code development. Therefore, it does not cause loss of time. JUnit reduces the error rate in the code, increases code quality, and detects errors quickly. Errors are found very easy and fast in automated codes or systems. Also, junit has porting properties in software engineering. **Porting** is a process of adapting software. Porting is an adapting of an executable program to different environments for software / hardware.

Unit Test Framework Name	JUnit	NUnit	CppUnit	PHPUnit
xUnit	✓	✓	✓	✓
Automated Run	✓	✓	✓	✓
Atomic Result PASS/FAIL	✓	✓	✓	✓
Language	Java	C# / .Net	C++	PHP
Integrated	✓	✓	✓	✓
Open Source	✓	✓	✓	✓
Common	✓	✓	✓	✓
Port / Based on	-	Junit	Junit	Junit
License	Eclipse Public License	Zlib License	BSD	LGPL

Table 1: Comparison of Most popular Unit Test Frameworks

2.3 UML

The Unified Modelling Language (UML) is a visual modelling and also a general programming language of the software system. In 1994-1995, Grady Booch, Ivar Jacobson and James Rumbaugh was developed this modeling language. UML was designed to incorporate current best practice in modelling technologies and software engineering. [23] This modeling language includes various behavior and structure diagrams.

✓ **Behavioral diagrams** represent specific processes of a system. And also, behavioral diagrams are very common diagrams. These diagrams are called as activity, use case, state machine, and interaction diagrams.

✓ **Interactional diagrams** are more than one diagram. These diagrams includes a subset of behavioral diagrams. These diagrams are called as sequence, communication, interaction overview, timing diagrams.

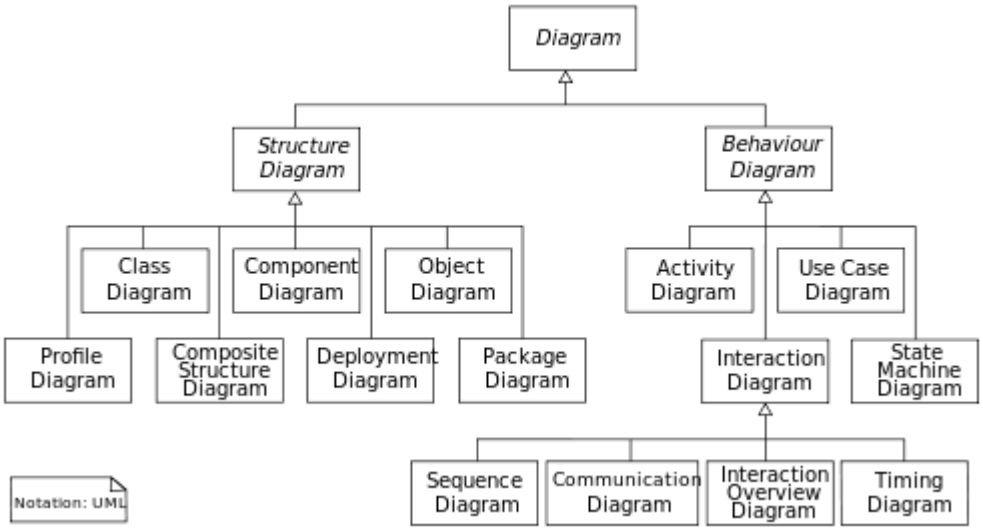


Figure 6: UML Diagrams as in [19]

2.3.1 Sequence Diagram

A sequence diagram is an interaction diagrams of UML behavior diagrams. This diagrams consists of a specific number objects and messages. Sequence diagram is shown with vertical lines and these lines consist of objects of the diagram. Sequence diagram messages are among objects, and they act according to a sequence. Sequence diagrams are based on a scenario. Every message is send or received according to the scenario.

In MBT with UML model use various diagrams such as UML activity, use case, sequence diagrams, etc. Activity, sequence and state-machine diagrams are commonly used for MBT.

In this study, it is used sequence diagram as UML diagram for MBT because sequence diagrams are very common in UML behavior models. Sequence diagram is very suitable for scenario-based testing in MBT. In scenario-based testing, a scenario is an abstract test case. Scenario-based testing tests are usually different from test cases in that test cases are single steps and scenarios cover a number of steps. [11] Scenarios are useful to use in terms of model test requirements and test cases. Also, this testing works the best in terms of very complex system or events. In MBT, using sequence diagram can give more effective results to find test cases according to other diagrams.

2.3.1.1 Sequence Diagram Elements

- **Object :** Objects defines classes of a sequence diagram, and they shows dependency of messages.

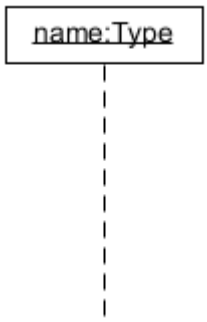


Figure 7: Lifeline of UML Sequence Diagram

- **Messages :** A message has a name and a parameter in a sequence diagram. Messages consist of various names such as synchronous messages, asynchronous messages, return messages, instantaneous messages and self messages.

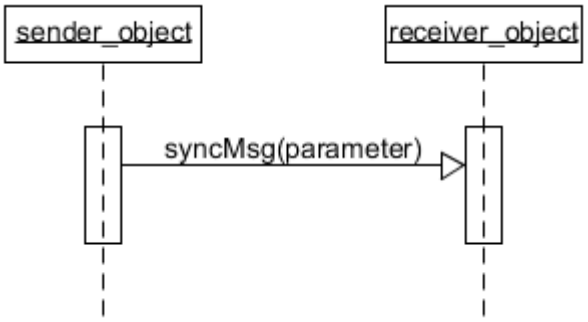


Figure 8: Synchronous message of UML Sequence Diagram

A synchronous message sends message between sender object and receiver object. The sender object waits to finish processing message of the receiver object.

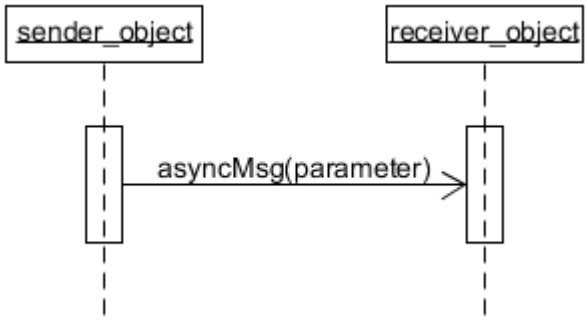


Figure 9: Asynchronous message of UML Sequence Diagram

The sender object does not wait to finish processing message of the receiver object in an asynchronous message. An asynchronous message does not have a return message.

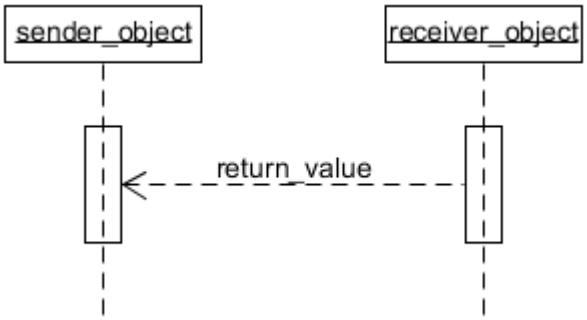


Figure 10: Return message of UML Sequence Diagram

Return messages are solutions of sending messages and they depends on an optional.

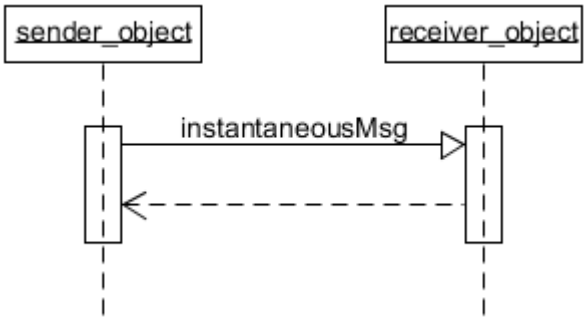


Figure 11: Instantaneous message of UML Sequence Diagram

The sending message arrives to receiving message, but the time is negligible in the instantaneous mesages.

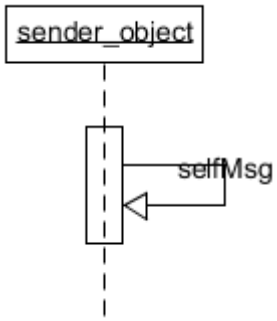


Figure 12: Self message of UML Sequence Diagram

An object sends message to itself that called self message.

- **Interaction Fragment (Conditional Interaction) :** The following table defines most useful operators of sequence diagrams.

alt	Divides fragment into groups and defines condition for each group - only the one whose condition is true will execute .
opt	Defines condition to a single call - the call will execute only if the supplied condition is true . Equivalent to an alt with only one trace.
par	Defines that the calls within the fragment run in parallel.
loop	Defines that the calls within the fragment run in a loop.
region	Defines that the calls within the fragment reside in a critical section, i.e. the fragment can have only one thread executing it at once.

Figure 13: Interaction Fragments of UML Sequence Diagram [20]

In this thesis, it is studied with UML sequnce diagram models in model based software testing. It is studied with sequence diagrams in different areas because the aim is to achieve a different perspective in terms of test results. It is shown different results for both manual and automated testing using models in different areas. Also, according to research papers, it is achieved more power results of this approach with different sequence diagram in MBT. In this thesis, sequence diagram topics are *'Banking System', 'Bank ATM System (Withdraw Money)', 'Deal Cards Scenario', 'Medical Consultation System', 'FaceBook Authorization', 'Bookstore Composite Web Service', 'Library Management System', 'Operation getRegistered()'.*

2.3.1.2 Supporting UML Tool

It is chosen Umlet as UML tool. Umlet has some good features such as free or open source, easy to use, multiple diagram, export diagram or file. This tool can be integrated and it has an easy GUI. This tool is a UML plug-in tool. Also, it is appropriate for UML 2 Graphical Model.

It is preferred Umlet UML tool instead of other UML tools. Umlet UML tool can be integrated easily with Eclipse. It is studied with XML files in this thesis. UML diagrams can be converted and read XML file with Umlet easily. This property provides very usable XML files. Umlet is very flexible to find test paths of a diagram with this property.

2.4 Automation Level or Complexity

Automation level or complexity measures and analyses rate of easiness and difficulty of the system. Automation levels change according to manual or automated systems. According to [21], In this thesis, automation level complexity criteria means:

•**Low (L):** Low level represents automated or involve a simple task. Tool is available.

•**Medium (M) :** Medium level represents intermediate model modeling or test data definition.

• **High (H):** High level represents manual translation between models. Tool does not support.

This thesis covers high, medium and low automation levels in MBT. The proposed manual testing covers high and medium complexity criterias because, manual testing contains intermediate model, and it does not use tool. The proposed automated testing contains low complexity criteria because it includes automatic running code. Also, this automated testing uses a tool.

2.5 Test Coverage Criterias

In MBT, models have different structural notations. These notations include structural model coverage criterias. These criteria specify the selection of tests from models in terms of coverage of the structural elements of the models. [22]

In this studies, it is used history-based model notation both manual and automated testing. This model notation includes all messages and message path coverage as structural model coverage. This criteria is evaluated all message for testing from a model. This approach is scenario-based testing and this approach supports functional system testing depending on inputs and outputs of the system. The following Table 2 shows all notations of model and structural model coverage in MBT.

Modeling Notation	Structural Model Coverage Example
State-based (Pre/Post)	Cause-Effect coverage, Disjunctive Normal Form (DNF) of post-conditions [12, page 138].
Transition-based	Various graph coverage criteria e.g., all-states, all-transitions, all-transition-pairs and all-cycles [12, page 32].
History-based	Each Message (EM) or Message Path (MP) coverage [63].
Functional	Coverage of axioms.
Operational	Similar to graph coverage e.g., transitions, places, cycles etc. [72].
Stochastic	Coverage from usage profile in terms of probabilities of states and transitions [90].
Data-flow	Data-path coverage, which is similar to code coverage criteria e.g., all-definitions, all-uses, all-definition-use pairs of data variables [41].

Table 2: Modeling notations and Structural Model coverage examples as in [22]

CHAPTER III

PROPOSED MANUAL MODEL BASED TESTING TECHNIQUE & COMPARISON WITH AUTOMATED TESTING

3.1 Proposed Approach

Many researchers have produced many approaches for MBT. These studies are different from each other in terms of techniques and algorithms. Some researhers prefer using tool that using automated testing, and some researhers prefer using their algorithms with manual testing to achieve test paths in MBT. Also, some researhers prefer using their own methods. They write their own automated codes using tools and programming language, or they develop their own algorithms using manual methods. The proposed approach is based on scenario-based testing. It is designed a novel approach with a new system using UML sequence diagram models for MBT, and two testing methods are used as manually and automatically for testing in this thesis.

3.2 Manual Testing

In this thesis, it is designed a new technique and algorithm as manually for model based software testing. The following subtitles explain the proposed technique and algorithms. A new technique that is called '*The Single Message Sharing Tree*' is developed for sequence diagram message system, and two algorithms for this system is written. The first algorithm that is called '*The Single Message Sharing Tree Algorithm*' explains what message tree is and how a message tree is created. Second algorithm that is called as '*Distribution of The Message Path Algorithm'* includes DFS path algorithm to read all test paths. Finally, all test paths for all test cases are reached in this thesis. In summary, desired results of test cases of the system are achieved.

The whole system is defined using a flow diagram. The following flow diagram shows all process of proposed manual testing.

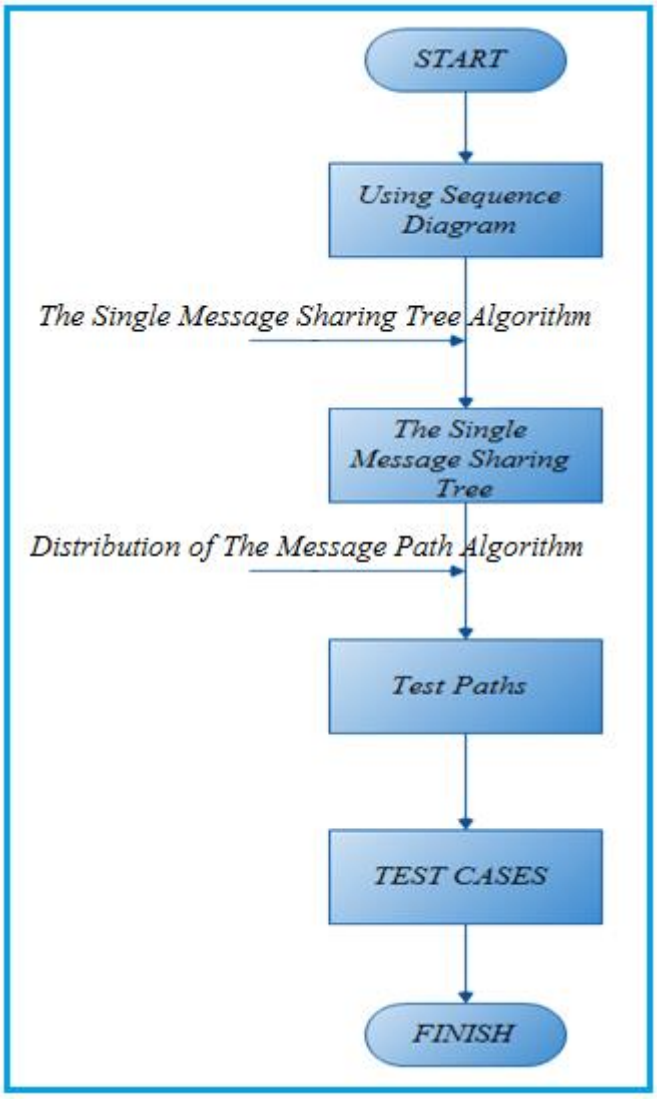


Figure 14: General Flow Diagram of Tree Testing Process

3.2.1 The Single Message Sharing Algorithm for Tree

UML sequence diagram is used as a model for the system. This model is a. A new approach using messages of the sequence diagram is produced. The proposed technique is a tree. A road algorithm is written that is called as ***'Algorithm 1'*** and then, the tree is generated according to algorithm 1. Algorithm 1 explains steps of tree. This algorithm generates nodes of tree. Also, all test paths of the tree are reached with this algorithm. Every step explains node dependency of tree according to messages of sequence diagram.

Algorithm 1: *Generate The Single Message Sharing Tree Algorithm*
Input: Sequence Diagram (SD)
Output: The Single Message Sharing Tree (SMST)

1) *Begin.*
2) *For each mc open a new node.* // mc is current message.
3) *For mc! = mf.* // continue until current message mc is not equal to mf that the final message. mc & mf are blue circles.
4) *mc = mi.* //mc start with initial message mi.
5) *Each mc=mi+1.* // Each mc connected the other message respectively that each message follow the other. (ex; 1-2-3,…)
6) *IF mc= R THEN open new node from S.* // every mc is equal to return message with respectively (ex; 6. message) then open a new node from sending message (S is sending message ex; 5). Sending message is equal to fork node. // R is return message that red circles(RM).
8) *Again continue step 5.*
9) *End.*

Figure 15: *Generate The Single Message Sharing Tree Algorithm*

3.2.2 Finding Test Paths Algorithm of Tree

The following algorithm is ***'Algorithm 2'*** that finds all test paths of the tree. Also, algorithm 2 benefits from depth first search path algorithm.

Algorithm 2: *Generate Distribution of The Message Path*
Input: The Single Message Sharing Tree (SMST)
Output: Distribution of The Message Path (DMT)

1) *Begin.*
2) *Traverse the SMST.*
3) *nc= Root Node.* *//Start node is equal to root node or initial message node of tree.*
4) *Repeat step 5until nc!= nf* *// nc is current node and nf is final node.*
5) *Using DFS Path Algorithm (Depth-First-Search).*
DFS (T, rn) *// T is tree and root node rn is the vertex where the search starts of tree.*
- *Initialize all nodes to ready state.*
Stack *S := {};* *// Create an empty stack S.*
for each vertex v, set visited[v]:=false;
push S,rn; *// Push root node to S.*
while(S is not empty) do *// while S is not empty.*
v :=pop S; *// Pop an item v from S.*
if *(not visited [v]) then* *// if v is not visited.*
visited [v]:= true;
for each unvisited neighbour n of v
push S,n; *// push n to S.*
end if
end while
- ***END DFS()***

6) *End.*

Figure 16: *Generate Distribution of The Message Path*

3.3 Automated Testing

In this study, a second technique is designed as automatically for model based software testing. The following subtitles explain this second technique. First of all, a sequence diagram is drawn using UMLet tool inside eclipse, and sequence diagram is converted to XML file using eclipse. Secondly, a new algorithm that is called as *'Table'* is written using java code to show all test data. All data of sequence diagram is extracted from an XML file. After then, all data is stored into a jtable with this ***'Table'*** code. Thirdly, a junit unit testing code that is called as ***'JUnitCode'*** is written for testing of all data. The junit testing code benefits from jtable. Furthermore, this code provides to find and show all test paths of sequence diagram in the java console. Finally, all test cases with these test paths are reached in this thesis.

The whole system is defined using a flow diagram. The following flow diagram Figure 17 shows all process of proposed automated testing. In Figure 18, well-formed of all XML files are shown. This form shows structure of all xml files.

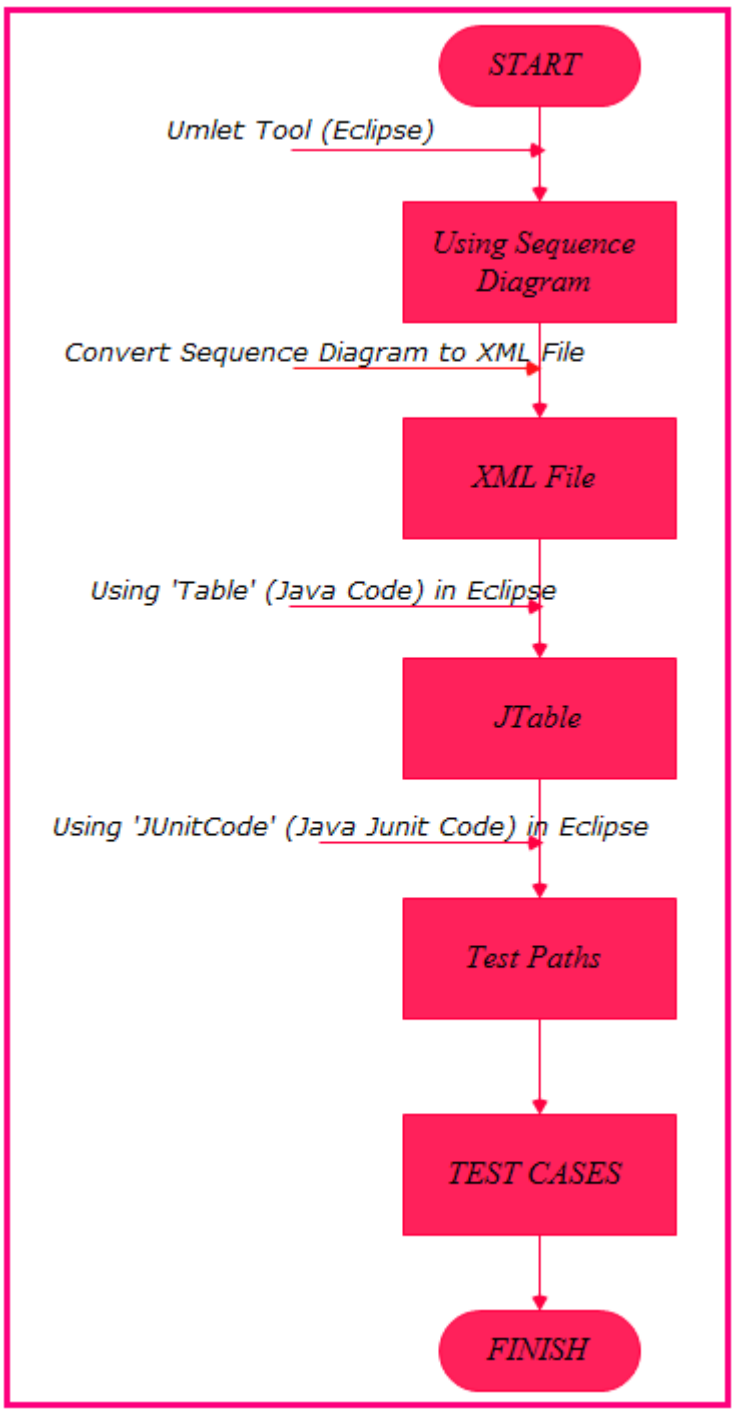


Figure 17: General Flow Diagram of JUnit Testing Process

Figure 18: Example of a well-formed XML File

Figure 19 shows the ***'Table'*** java code. It is used parser method to convert from XML file to jtable in the *'Table'* java code. The XML file is in **Appendix K**. In this study, java jtable method is used to extract all necessary data of sequence diagram, and all data is stored into a jtable with this code.

In chapter 3, a new testing code is called as ***'JUnitCode'*** is in **Appendix L**. The purpose is to find all test scenario cases of sequence diagram from test paths with junit testing. This junit testing code incudes some test methods for test requirements. First method is *testPossibleTestPaths()* method. This method is necessary to find all possible test path sequence. Second method is called *testAllPaths()* to read names of sequence diagram messages according to test paths. Third method is called *testReadMessages()* to show all messages and also number of message of sequence diagram. Final method is called *testBestPath()* to show best test path of the sequence diagram. All necessary test path of the sequence diagram are found using jtable data with junit testing code in proposed automated test method.

```
package test;
import org.w3c.dom.*;
import javax.xml.parsers.*;
import java.awt.BorderLayout;
import java.util.*;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.*;

public class Table extends AbstractTableModel{
/****/
private static final long serialVersionUID = 1L;
Vector<String> row_identify = new Vector<String>();
Vector<String> column_identify = new Vector<String>();
String[] columns_name = {"Case_ID", "Activity_Name"};
public String getColumnName(int index) {
return columns_name[index];
}
public Table () {
DocumentBuilderFactory doc_factory = DocumentBuilderFactory.newInstance();
try {
DocumentBuilder doc_builder = doc_factory.newDocumentBuilder();
Document xml = doc_builder.parse("File.xml");
Element root_element = xml.getDocumentElement();
NodeList first_element = root_element.getElementsByTagName("id");
NodeList second_element = root_element.getElementsByTagName("panel_attributes");
NodeList element_name = root_element.getElementsByTagName("element");
for (int j = 0; j < element_name.getLength(); j++) {
String element_one = first_element.item(j).getFirstChild().getNodeValue();
Stringelement_two = second_element.item(j).getFirstChild().getNodeValue();
if (element_one.equals("Relation") && element_two.equals("lt=.") ||
element_two.equals("lt=-")) {
row_identify.remove("lt=.");
row_identify.remove("lt=-");}
else if (element_one.equals("UMLGeneric") && element_two.equals("//Line")) {
row_identify.remove("//Line");}
else{
row_identify.addElement(element_one);
row_identify.addElement(element_two);}}
column_identify.add("");
column_identify.add("");
} catch (Exception ex) {
ex.printStackTrace();
System.out.println("The System is ERROR!... ");}}
public static void main(String argv[]) throws Exception {
Table data = new Table();
JTable jtable = new JTable();
jtable.setModel(data);
jtable.setPreferredScrollableViewportSize(new java.awt.Dimension(700, 600));
JScrollPane pane = new JScrollPane(jtable);
JPanel j_panel = new JPanel();
j_panel.add(pane);
JFrame frame = new JFrame("...MESSAGE CASE TABLE for TEST PATHS..."); // This part
is table name.
frame.add(j_panel, BorderLayout.CENTER);
frame.pack();
frame.setVisible(true);
System.out.println("The JTable load successfully... ");}
public int getRowCount() {
return row_identify.size() / getColumnCount();}
public int getColumnCount() {
return column_identify.size();}
public Object getValueAt(int rowIndex, int columnIndex) {
return (String) row_identify.elementAt((rowIndex * getColumnCount()) +
columnIndex);}
}
```

Figure 19: '*Table*' java code for JTable

3.4 An Example For Proposed Approach in MBT

The proposed manual and automated test method are explained with a good example. UML sequence diagram in Figure 20 is used as a model. This model includes a hospital management system. The following parts explain testing results of the proposed manual and automated methods.

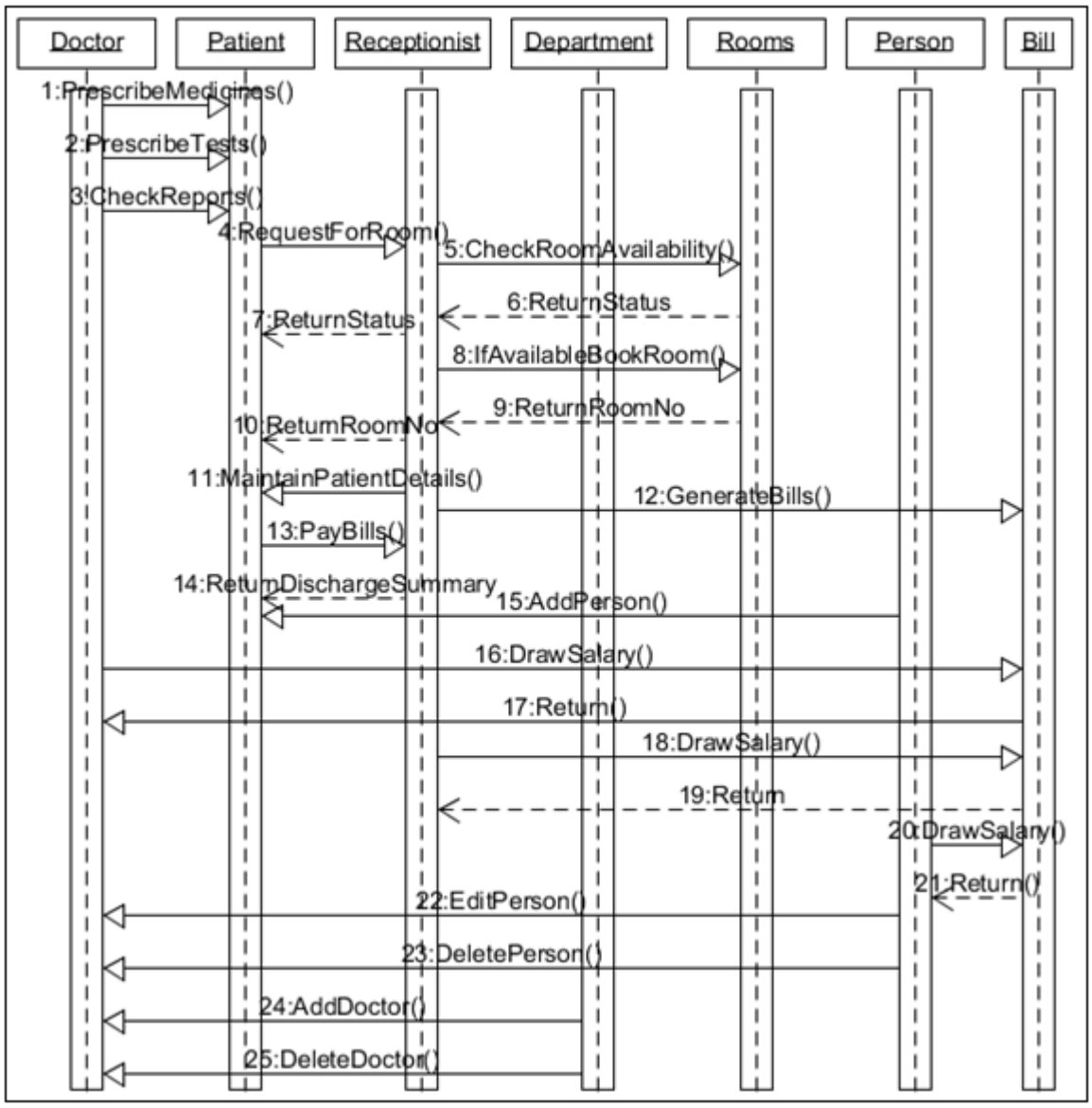


Figure 20: Example Sequence Diagram of the Hospital Management System as in [24]

3.4.1 Tree & Test Paths For Proposed Manual Testing

In Figure 21 shows the proposed tree. Tree starts root node, and connects nodes to edges from start message to end message respectively. However, after every RM, a new node from SM opens (for example; if 5. message is SM & 6. message is RM

node, a new node from SM opens for 7. message) and continues. RM node is called return message. Also, the final node is called the final message. Table 3 and 4 show all possible test paths from the tree. The purpose is to evaluate all sending and return messages. Figure 21 is the proposed message tree as a technique, and *Algorithm 1* is the algorithm of this tree technique.

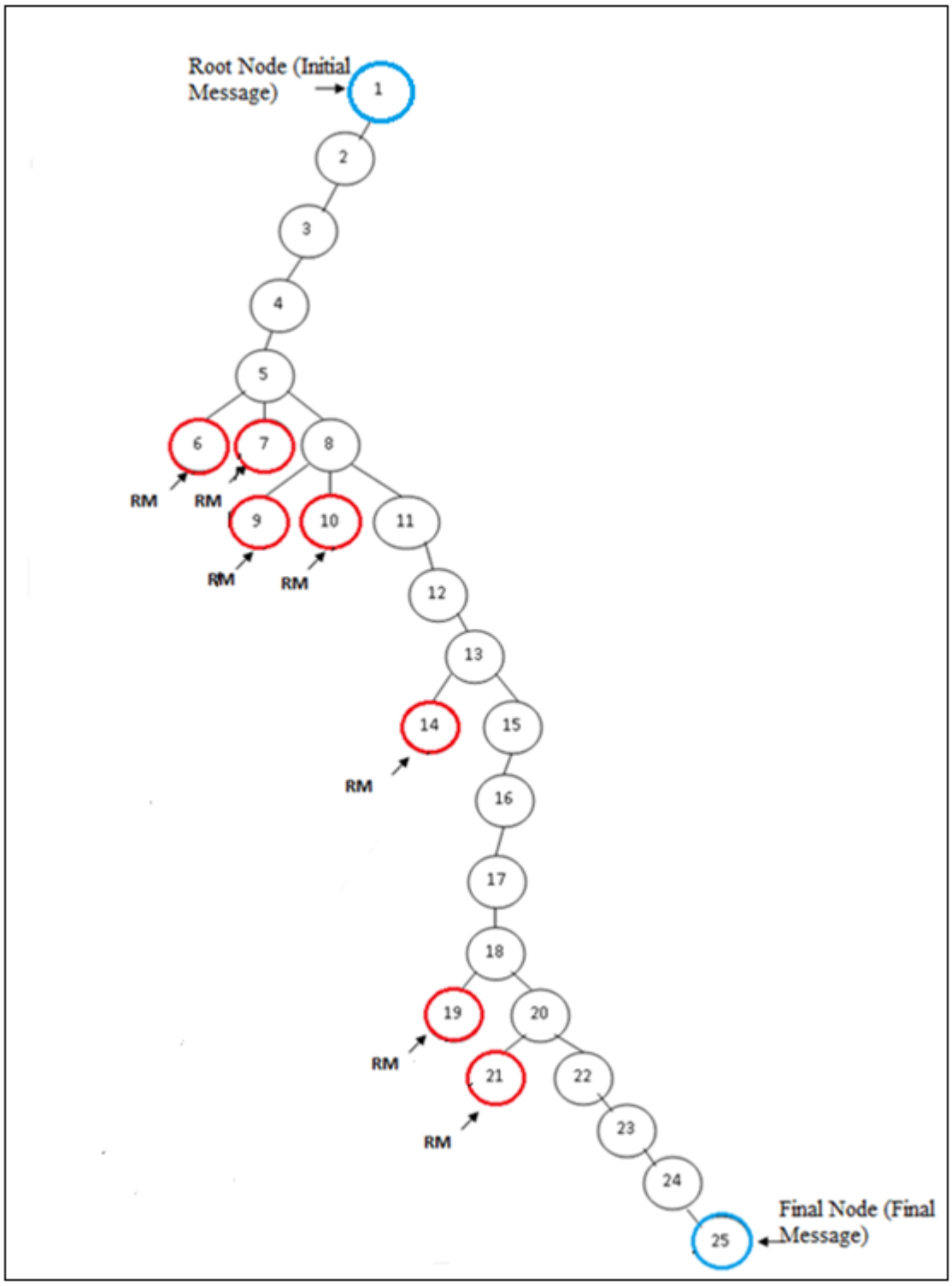


Figure 21: *The Single Message Sharing Tree (SMST)* in Figure 20

- ***Test Paths***

Path ID	Test Paths
1	1-2-3-4-5-6
2	1-2-3-4-5-7
3	1-2-3-4-5-8-9
4	1-2-3-4-5-8-10
5	1-2-3-4-5-8-11-12-13-14
6	1-2-3-4-5-8-11-12-13-15-16-17-18-19
7	1-2-3-4-5-8-11-12-13-15-16-17-18-20-21
8	1-2-3-4-5-8-11-12-13-15-16-17-18-20-22-23-24-25

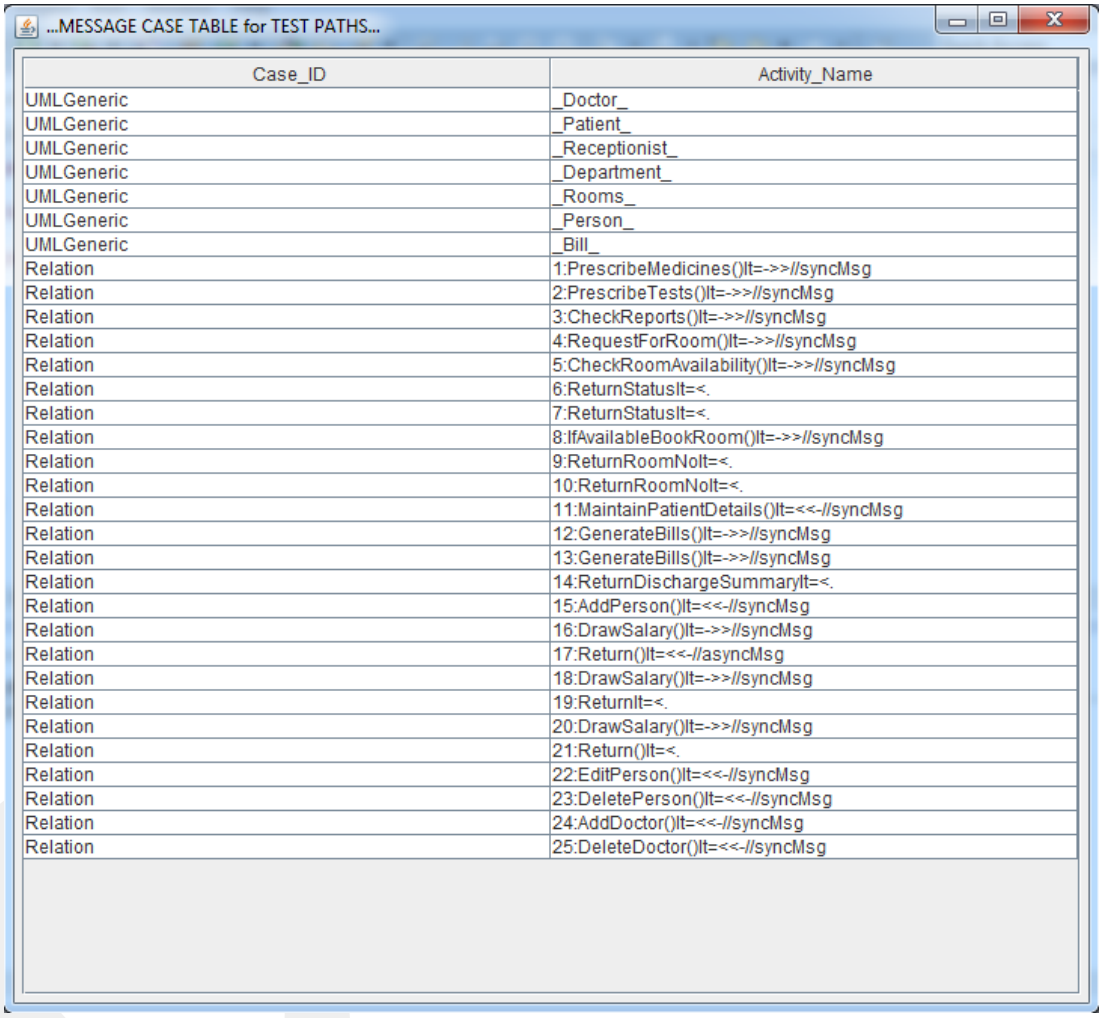
Table 3: Sequence of All Test Paths in Figure 21

Path ID	Test Paths
1	*Prescribe Medicines-Precribe Tests-CheckReports-RequestForRoom-Check Room Availability-ReturnStatus*
2	*Prescribe Medicines-Precribe Tests-CheckReports-RequestForRoom-Check Room Availability-ReturnStatus*
3	*Prescribe Medicines-Precribe Tests-CheckReports-RequestForRoom-Check Room Availability-If Available BookRoom-Return RoomNo*
4	*Prescribe Medicines-Precribe Tests-CheckReports-RequestForRoom-Check Room Availability-If Available BookRoom-Return RoomNo*
5	*Prescribe Medicines-Precribe Tests-CheckReports-RequestForRoom-Check Room Availability-If Available BookRoom-MaintainPatientDetails-Generate Bills-PayBills-Return DischargeSummary*
6	*Prescribe Medicines-Precribe Tests-CheckReports-RequestForRoom-Check Room Availability-If Available BookRoom-MaintainPatientDetails-GenerateBills-PayBills-AddPerson-DrawSalary-Return-DrawSalary-Return*
7	*Prescribe Medicines-Precribe Tests-CheckReports-RequestForRoom-Check Room Availability-If Available BookRoom-MaintainPatientDetails-Generate Bills-PayBills-AddPerson-DrawSalary-Return-DrawSalary-DrawSalary-Return*
8	*PrescribeMedicines-PrecribeTests-CheckReports-RequestForRoom-CheckRoom Availability-IfAvailableBookRoom-MaintainPatientDetails-GenerateBills-PayBills-AddPerson-DrawSalary-Return-DrawSalary-DrawSalary-EditPerson -Delete Person-Add Doctor-Delete Doctor*

Table 4: List of All Test Paths Name of Figure 21

3.4.2 JTable & Test Paths For Proposed Automated Testing

All necessary elements of XML file are loaded into jtable with *'Table'* code. Jtable includes all necessary information of sequence diagram for test paths in Figure 22. Also, the following part shows all test paths results of proposed junit automated testing screen results in *'JUnitCode'* code.



Case_ID	Activity_Name
UMLGeneric	_Doctor_
UMLGeneric	_Patient_
UMLGeneric	_Receptionist_
UMLGeneric	_Department_
UMLGeneric	_Rooms_
UMLGeneric	_Person_
UMLGeneric	_Bill_
Relation	1:PrescribeMedicines()lt=->>//syncMsg
Relation	2:PrescribeTests()lt=->>//syncMsg
Relation	3:CheckReports()lt=->>//syncMsg
Relation	4:RequestForRoom()lt=->>//syncMsg
Relation	5:CheckRoomAvailability()lt=->>//syncMsg
Relation	6:ReturnStatuslt=<.
Relation	7:ReturnStatuslt=<.
Relation	8:IfAvailableBookRoom()lt=->>//syncMsg
Relation	9:ReturnRoomNolt=<.
Relation	10:ReturnRoomNolt=<.
Relation	11:MaintainPatientDetails()lt=<<-//syncMsg
Relation	12:GenerateBills()lt=->>//syncMsg
Relation	13:GenerateBills()lt=->>//syncMsg
Relation	14:ReturnDischargeSummarylt=<.
Relation	15:AddPerson()lt=<<-//syncMsg
Relation	16:DrawSalary()lt=->>//syncMsg
Relation	17:Return()lt=<<-//asyncMsg
Relation	18:DrawSalary()lt=->>//syncMsg
Relation	19:Returnlt=<.
Relation	20:DrawSalary()lt=->>//syncMsg
Relation	21:Return()lt=<.
Relation	22:EditPerson()lt=<<-//syncMsg
Relation	23:DeletePerson()lt=<<-//syncMsg
Relation	24:AddDoctor()lt=<<-//syncMsg
Relation	25:DeleteDoctor()lt=<<-//syncMsg

Figure 22: JTable of XML File in Appendix K

- **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3 - 4 - 5 - 6]
2. [1 - 2 - 3 - 4 - 5 - 6 - 7]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19]
```

7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 25]

1. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus]
2. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus]
3. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus - 8:IfAvailableBookRoom() - 9:ReturnRoomNo]
4. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus - 8:IfAvailableBookRoom() - 9:ReturnRoomNo - 10:ReturnRoomNo]
5. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus - 8:IfAvailableBookRoom() - 9:ReturnRoomNo - 10:ReturnRoomNo - 11:MaintainPatientDetails() - 12:GenerateBills() - 13:GenerateBills() - 14:ReturnDischargeSummary]
6. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus - 8:IfAvailableBookRoom() - 9:ReturnRoomNo - 10:ReturnRoomNo - 11:MaintainPatientDetails() - 12:GenerateBills() - 13:GenerateBills() - 14:ReturnDischargeSummary - 15:AddPerson() - 16:DrawSalary() - 17:Return() - 18:DrawSalary() - 19:Return]
7. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus - 8:IfAvailableBookRoom() - 9:ReturnRoomNo - 10:ReturnRoomNo - 11:MaintainPatientDetails() - 12:GenerateBills() - 13:GenerateBills() - 14:ReturnDischargeSummary - 15:AddPerson() - 16:DrawSalary() - 17:Return() - 18:DrawSalary() - 19:Return - 20:DrawSalary() - 21:Return()]
8. [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus - 8:IfAvailableBookRoom() - 9:ReturnRoomNo - 10:ReturnRoomNo - 11:MaintainPatientDetails() - 12:GenerateBills() - 13:GenerateBills() - 14:ReturnDischargeSummary - 15:AddPerson() - 16:DrawSalary() - 17:Return() - 18:DrawSalary() - 19:Return - 20:DrawSalary() - 21:Return() - 22:EditPerson() - 23:DeletePerson() - 24:AddDoctor() - 25:DeleteDoctor()]

BEST PATH

The Best Path: 8. Path [1:PrescribeMedicines() - 2:PrescribeTests() - 3:CheckReports() - 4:RequestForRoom() - 5:CheckRoomAvailability() - 6:ReturnStatus - 7:ReturnStatus - 8:IfAvailableBookRoom() - 9:ReturnRoomNo - 10:ReturnRoomNo - 11:MaintainPatientDetails() - 12:GenerateBills() - 13:GenerateBills() - 14:ReturnDischargeSummary - 15:AddPerson() - 16:DrawSalary() - 17:Return() - 18:DrawSalary() - 19:Return - 20:DrawSalary() - 21:Return() - 22:EditPerson() - 23:DeletePerson() - 24:AddDoctor() - 25:DeleteDoctor()]

ALL MESSAGES

All messages of sequence diagram: [1:PrescribeMedicines() -> 2:PrescribeTests() -> 3:CheckReports() -> 4:RequestForRoom() -> 5:CheckRoomAvailability() -> 6:Return Status -> 7:ReturnStatus -> 8:IfAvailableBookRoom() -> 9:ReturnRoomNo -> 10:Return RoomNo -> 11:MaintainPatientDetails() -> 12:GenerateBills() -> 13:GenerateBills() -> 14:ReturnDischargeSummary -> 15:AddPerson() -> 16:DrawSalary () -> 17:Return() -> 18:DrawSalary() -> 19:Return -> 20:DrawSalary() -> 21:Return() -> 22:EditPerson() -> 23:DeletePerson() -> 24:AddDoctor() -> 25:DeleteDoctor()]

Total message number of sequence diagram: 25

3.4.3 Test Cases Table for Both Manual & Automated Testing

The following table is a test cases table for all requirements of both manual and automated testing. The proposed test cases table includes test case id, test case sequence, activity name id, test case inputs and test case expected and actual outputs for sequence diagram. Test case sequence defines order of all test paths until the last message for each test case. Test case input defines all messages of sequence diagram. Test case output defines both expected and actual results for sequence diagram. Expected output is proposed outputs. Also, actual output is return messages for each input message.

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3-4-5-6	1	Prescribe Medicines	Treatment Results	-
		2	Prescribe Tests	Test Results	-
		3	CheckReports	Check Report	-
		4	RequestForRoom	Return Status	Return Status
		5	CheckRoomAvaiability	Return Status	Return Status
		6	Return Status		
2	1-2-3-4-5-7	7	Return Status		
3	1-2-3-4-5-8-9	8	IfAvailableBookRoom	Return Room No	Return Room No
		9	Return Room No		
4	1-2-3-4-5-8-10	10	Return Room No		
5	1-2-3-4-5-8-11-12-13-14	11	MaintainPatientDetails	Receptionist Control	-
		12	GenerateBills	Bill	-
		13	PayBills	Return Discharge Summary	Return Discharge Summary
		14	Return Discharge Summary		
6	1-2-3-4-5-8-11-12-13-15-16-17-18-19	15	AddPerson	Record Patient Info	-
		16	DrawSalary	Return	Return
		17	Return		
		18	DrawSalary	Return	Return
		19	Return		
7	1-2-3-4-5-8-11-12-13-15-16-17-18-20-21	20	DrawSalary	Return	Return
		21	Return		

8	1-2-3-4-5-8-11-12-13-15-16-17-18-20-22-23-24-25	**22**	EditPerson	Record Patient Info	-
		23	DeletePerson	Delete Patient Info	-
		24	AddDoctor	Record Doctor Info	-
		25	DeleteDoctor	Delete Doctor Info	-

Table 5: Test Cases Table of Example for Manual & Automated Testing

3.5 Testing Results with Examples

In this thesis, the same sequence diagrams examples is used for both manual and automated testing. It is reached all test paths results in terms of proposed manual and automated methods with the following examples. In conclusion, it is achieved test cases tables from all manual and automated testing results. Also, the proposed approach is compared to approaches in research papers, and all results are shown in a different table.

3.5.1 Example1

The model is a banking system (ATM). Example 1 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix A** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

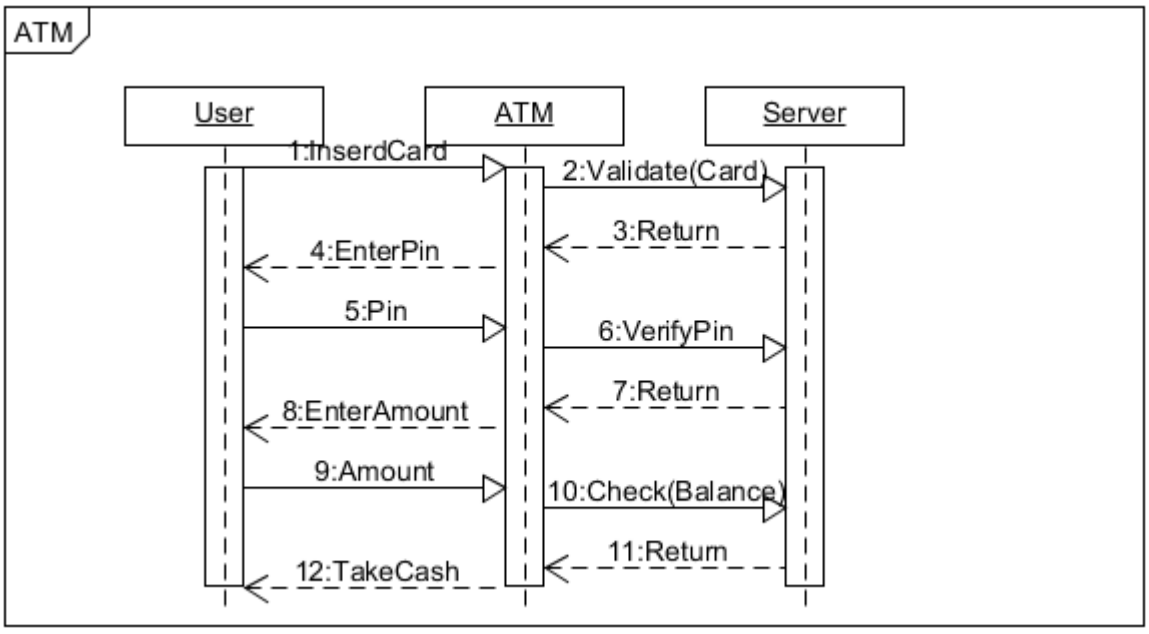


Figure 23: Sequence Diagram of Banking System [3]

3.5.1.1 Manual Testing of Example 1

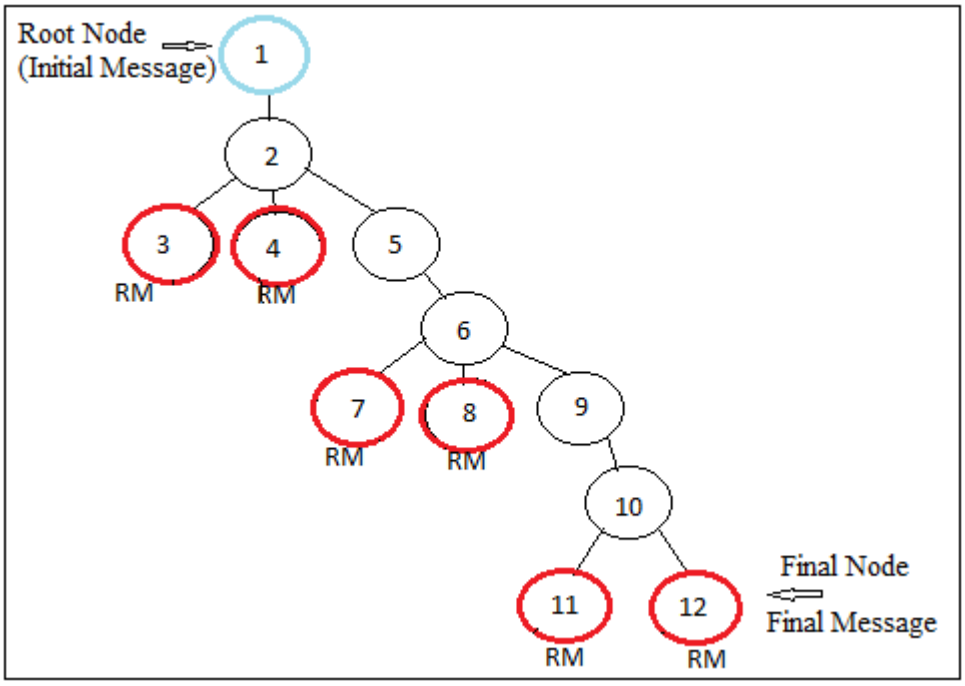


Figure 24: *The Single Message Sharing Tree (SMST)* of Figure23

- ***Test Paths***

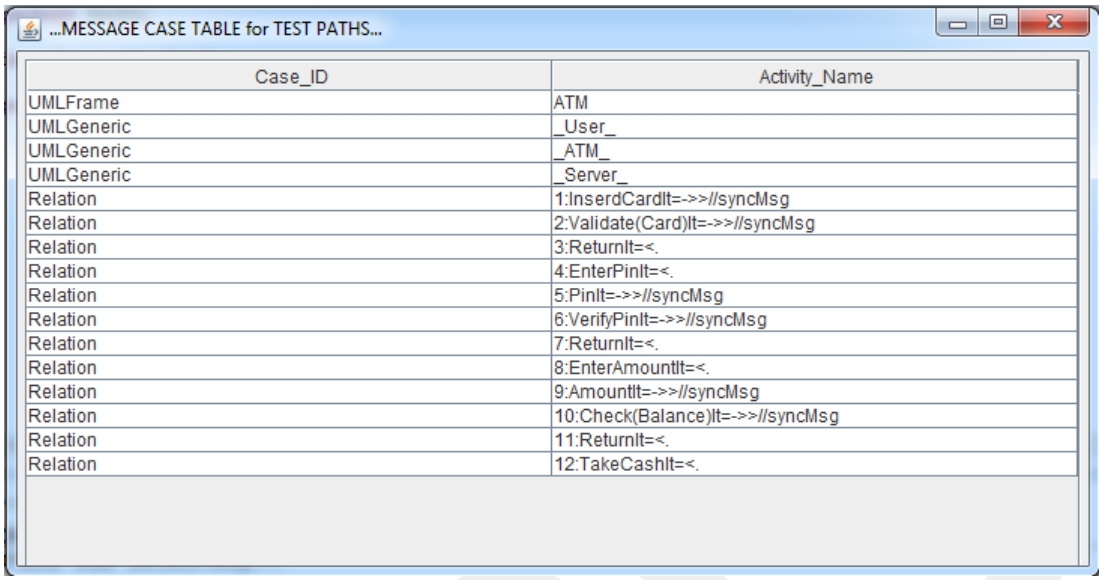
Path ID	Test Paths
1	1-2-3
2	1-2-4
3	1-2-5-6-7
4	1-2-5-6-8
5	1-2-5-6-9-10-11
6	1-2-5-6-9-10-12

Table 6: Sequence of All Test Paths in Figure 24

Path ID	Test Paths
1	*Insert card-Validate(card)-Return*
2	*Insert card-Validate(card)-EnterPin*
3	*Insert card-Validate(card)-Pin-VerifyPin-Return*
4	*Insert card-Validate(card)-Pin-VerifyPin-EnterAmount*
5	*Insert card-Validate(card)-Pin-VerifyPin-Amount-Check(Balance)-Return*
6	*Insert card-Validate(card)-Pin-VerifyPin-Amount-Check(Balance)-TakeCash*

Table 7: List of All Test Paths Name of Figure 24

3.5.1.2 Automated Testing of Example 1



Case_ID	Activity_Name
UMLFrame	ATM
UMLGeneric	_User_
UMLGeneric	_ATM_
UMLGeneric	_Server_
Relation	1:InserdCardIt=->>//syncMsg
Relation	2:Validate(Card)It=->>//syncMsg
Relation	3:ReturnIt=<.
Relation	4:EnterPinIt=<.
Relation	5:PinIt=->>//syncMsg
Relation	6:VerifyPinIt=->>//syncMsg
Relation	7:ReturnIt=<.
Relation	8:EnterAmountIt=<.
Relation	9:AmountIt=->>//syncMsg
Relation	10:Check(Balance)It=->>//syncMsg
Relation	11:ReturnIt=<.
Relation	12:TakeCashIt=<.

Figure 25: JTable of XML File in Appendix A

- **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12]

1. [1:InserdCard - 2:Validate(Card) - 3:Return]
2. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin]
3. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin -
7:Return]
4. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin -
7:Return - 8:EnterAmount]
5. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin -
7:Return - 8:EnterAmount - 9:Amount - 10:Check(Balance) - 11:Return]
6. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin -
7:Return - 8:EnterAmount - 9:Amount - 10:Check(Balance) - 11:Return - 12:TakeCash]


BEST PATH

The Best Path: 6.  Path [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin -
5:Pin - 6:VerifyPin - 7:Return - 8:EnterAmount - 9:Amount - 10:Check(Balance) -
11:Return - 12:TakeCash]

ALL MESSAGES

All messages of sequence diagram: [1:InserdCard - 2:Validate(Card) - 3:Return -
4:EnterPin - 5:Pin - 6:VerifyPin - 7:Return - 8:EnterAmount - 9:Amount -
10:Check(Balance) - 11:Return - 12:TakeCash]

Total message number of sequence diagram: 12
```

- **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3	**1**	Insert card	Loading Number	-
		2	Validate(card)	Valid Card	-
		3	Return		
2	1-2-4	**4**	EnterPin		
3	1-2-5-6-7	**5**	Pin	Valid Pin	-
		6	VerifyPin	Return Verify Pin	Return
		7	Return		
4	1-2-5-6-8	**8**	EnterAmount		
5	1-2-5-6-9-10-11	**9**	Amount	Sufficient Amount	-
		10	Check(Balance)	Balance Control	Return
		11	Return		
6	1-2-5-6-9-10-12	**12**	TakeCash		

Table 8: Test Cases Table of Example1

3.5.2 Example2

The model is a bank ATM system (Withdraw Money). Example 2 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix B** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

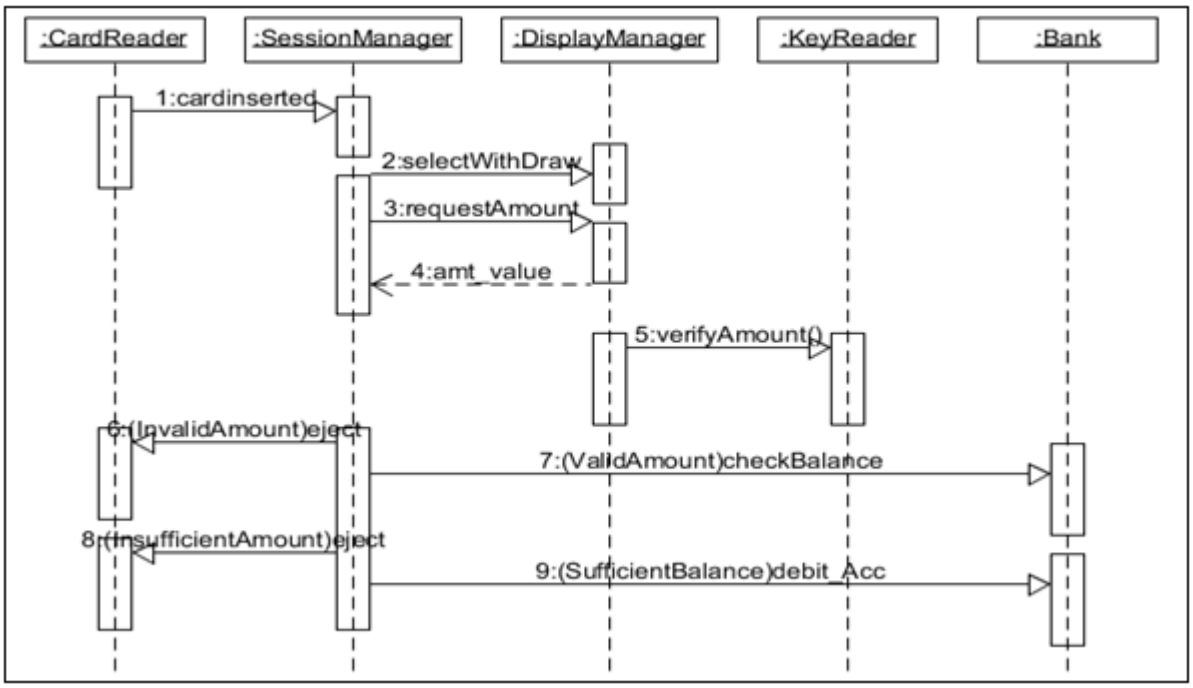


Figure 26: Sequence Diagram of Bank ATM System (Withdraw Money) [1]

3.5.2.1 Manual Testing of Example 2

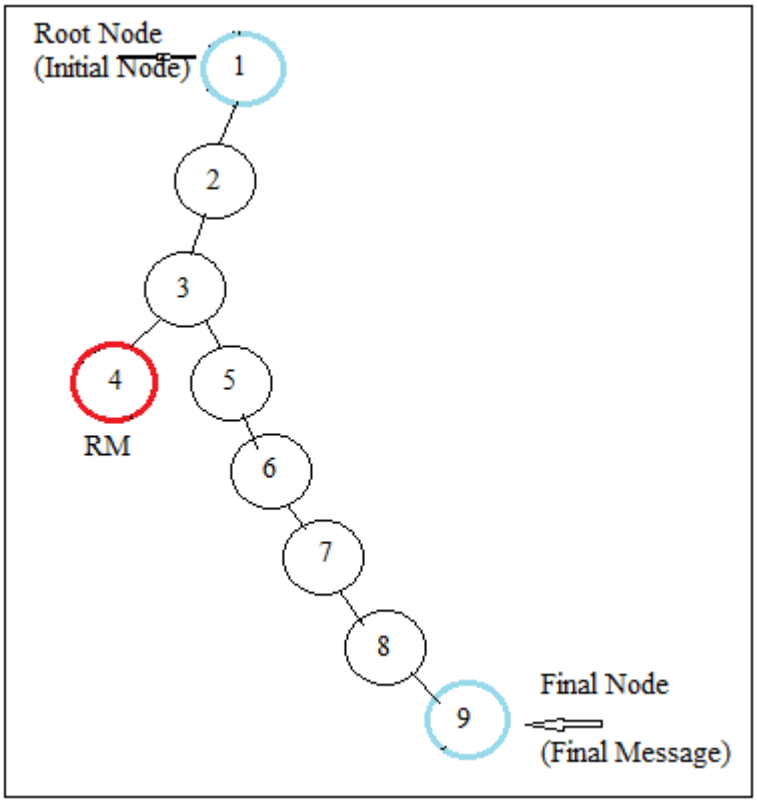


Figure 27: *The Single Message Sharing Tree (SMST)* of Figure 26

➢ ***Test Paths***

Path ID	Test Paths
1	1-2-3-4
2	1-2-3-5-6-7-8-9

Table 9: Sequence of All Test Paths in Figure 27

Path ID	Test Paths
1	*cardInserted-selectWithDraw-requestAmount-amt_value*
2	*cardInserted-selectWithDraw-requestAmount-verifyAmount()-(InvalidAmount)eject-(ValidAmount)checkBalance-(InsufficientAmount)eject-(SufficientBalance)debit_Acc*

Table 10: List of All Test Paths Name of Figure 27

3.5.2.2 Automated Testing of Example 2

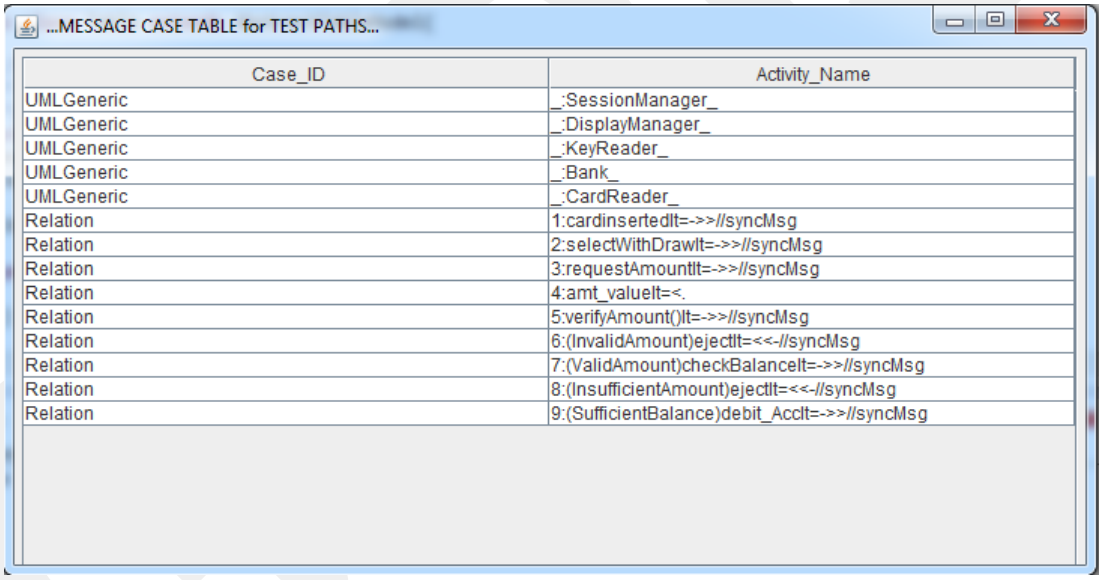


Figure 28: JTable of XML File in Appendix B

➢ **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3 - 4]
2. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9]

1. [1:cardinserted - 2:selectWithDraw - 3:requestAmount - 4:amt_value]
2. [1:cardinserted - 2:selectWithDraw - 3:requestAmount - 4:amt_value -
5:verifyAmount() - 6:(InvalidAmount)eject - 7:(ValidAmount)checkBalance -
8:(InsufficientAmount)eject - 9:(SufficientBalance)debit_Acc]
BEST PATH
```

```
The Best Path: 2.  Path [1:cardinserted - 2:selectWithDraw - 3:requestAmount -
4:amt_value - 5:verifyAmount() - 6:(InvalidAmount)eject -
7:(ValidAmount)checkBalance - 8:(InsufficientAmount)eject -
9:(SufficientBalance)debit_Acc]

ALL MESSAGES

All messages of sequence diagram: [1:cardinserted -> 2:selectWithDraw ->
3:requestAmount -> 4:amt_value -> 5:verifyAmount() -> 6:(InvalidAmount)eject ->
7:(ValidAmount)checkBalance -> 8:(InsufficientAmount)eject ->
9:(SufficientBalance)debit_Acc]

Total message number of sequence diagram: 9
```

- **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3-4	**1**	cardinserted	Loading Number	-
		2	selectWithDraw	Return Selected	-
		3	requestAmount	Amount Value	amt_value
		4	amt_value		
2	1-2-3-5-6-7-8-9	**5**	verifyAmount()	Check Verify Amount	-
		6	(InvalidAmount)eject	Check Valid Amount	-
		7	(ValidAmount)checkBalance	Valid Amount	-
		8	(InsufficientAmount) eject	Sufficient Amount	-
		9	(SufficientBalance)debit_Acc	Sufficient Amount	-

Table 11: Test Cases Table of Example2

3.5.3 Example3

The model is a deal cards scenario. Example 3 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix C** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

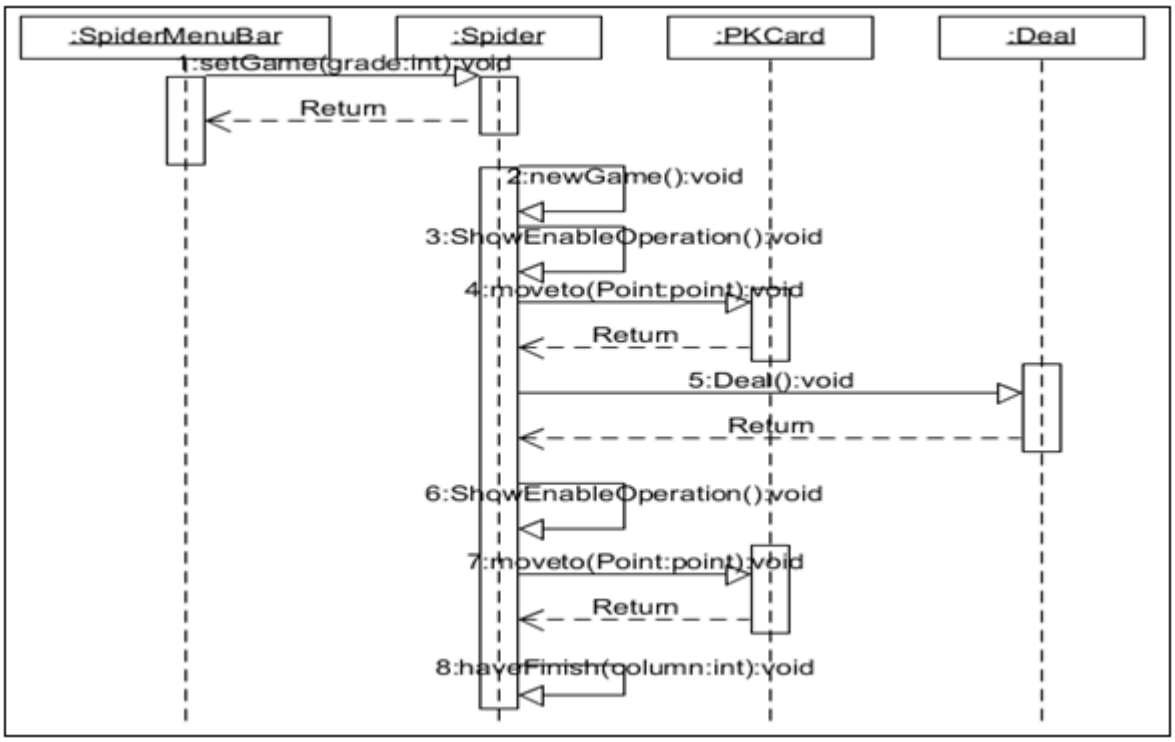


Figure 29: The Sequence Diagram for Deal Cards Scenario [10]

3.5.3.1 Manual Testing of Example 3

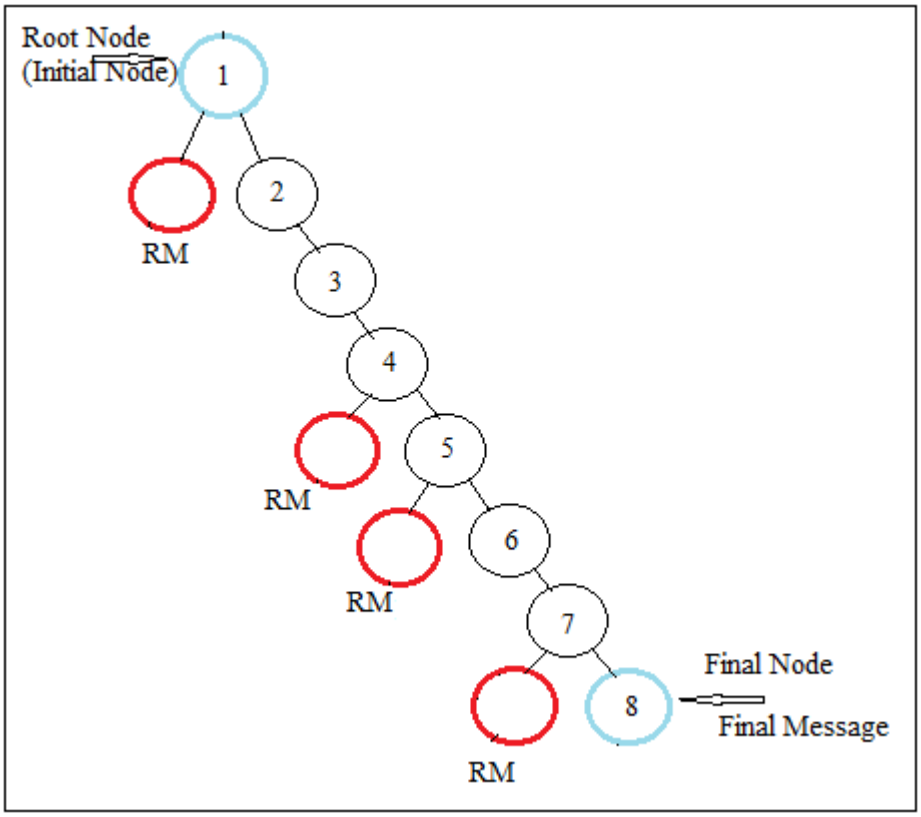


Figure 30: *The Single Message Sharing Tree (SMST)* of Figure 29

➢ ***Test Paths***

Path ID	Test Paths
1	1
2	1-2-3-4
3	1-2-3-4-5
4	1-2-3-4-5-6-7
5	1-2-3-4-5-6-7-8

Table 12: Sequence of All Test Paths in Figure 30

Path ID	Test Paths
1	*setGame*
2	*setGame-newGame-ShowEnableOperation-moveto*
3	*setGame-newGame-ShowEnableOperation-moveto-Deal*
4	*setGame-newGame-ShowEnableOperation-moveto-Deal-ShowEnableOperation-moveto*
5	*setGame-newGame-ShowEnableOperation-moveto-Deal- ShowEnableOperation-moveto-haveFinish*

Table 13: List of All Test Paths Name of Figure 30

3.5.3.2 Automated Testing of Example 3

...MESSAGE CASE TABLE for TEST PATHS...

Case_ID	Activity_Name
UMLGeneric	_:SpiderMenuBar_
UMLGeneric	_:Spider_
UMLGeneric	_:PKCard_
UMLGeneric	_:Deal_
Relation	1:setGame(grade:int):voidIt=->>//syncMsg
Relation	ReturnIt=<.
Relation	2:newGame():voidIt=<<-//syncMsg
Relation	3:ShowEnableOperation():voidIt=<<-//syncMsg
Relation	4:moveto(Point:point):voidIt=->>//syncMsg
Relation	ReturnIt=<.
Relation	5:Deal():voidIt=->>//syncMsg
Relation	ReturnIt=<.
Relation	6:ShowEnableOperation():voidIt=<<-//syncMsg
Relation	7:moveto(Point:point):voidIt=->>//syncMsg
Relation	ReturnIt=<.
Relation	8:haveFinish(column:int):voidIt=<<-//syncMsg

Figure 31: JTable of XML File in Appendix C

➢ All Results for Test Paths

```
TEST PATHS
All Possible Test Paths...

1. [1]
2. [1 - 2 - 3 - 4]
3. [1 - 2 - 3 - 4 - 5]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]

1. [1:setGame(grade:int):void - Return]
2. [1:setGame(grade:int):void - Return - 2:newGame():void -
3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return]
3. [1:setGame(grade:int):void - Return - 2:newGame():void -
3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return - 5:Deal():void
- Return]
4. [1:setGame(grade:int):void - Return - 2:newGame():void -
3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return - 5:Deal():void
- Return - 6:ShowEnableOperation():void - 7:moveto(Point:point):void - Return]
5. [1:setGame(grade:int):void - Return - 2:newGame():void -
3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return - 5:Deal():void
- Return - 6:ShowEnableOperation():void - 7:moveto(Point:point):void - Return -
8:haveFinish(column:int):void]
BEST PATH

The Best Path: 5.  Path [1:setGame(grade:int):void - Return - 2:newGame():void -
3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return - 5:Deal():void
- Return - 6:ShowEnableOperation():void - 7:moveto(Point:point):void - Return -
8:haveFinish(column:int):void]

ALL MESSAGES

All messages of sequence diagram: [1:setGame(grade:int):void -> Return ->
2:newGame():void -> 3:ShowEnableOperation():void -> 4:moveto(Point:point):void ->
Return -> 5:Deal():void -> Return -> 6:ShowEnableOperation():void ->
7:moveto(Point:point):void -> Return -> 8:haveFinish(column:int):void]

Total message number of sequence diagram: 12
```

➢ Test Cases Table

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1	**1**	setGame	Return	Return
2	1-2-3-4	**2**	newGame	Select Game	-
		3	ShowEnableOperation	Return Operation	-
		4	moveto	Return	Return
3	1-2-3-4-5	**5**	Deal	Return	Return
4	1-2-3-4-5-6-7	**6**	ShowEnableOperation	Return Operation	-
		7	moveto	Return	Return
5	1-2-3-4-5-6-7-8	**8**	haveFinish	Return	Finish

Table 14: Test Cases Table of Example 3

3.5.4 Example4

The model is a medical consultation system. Example 4 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix D** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

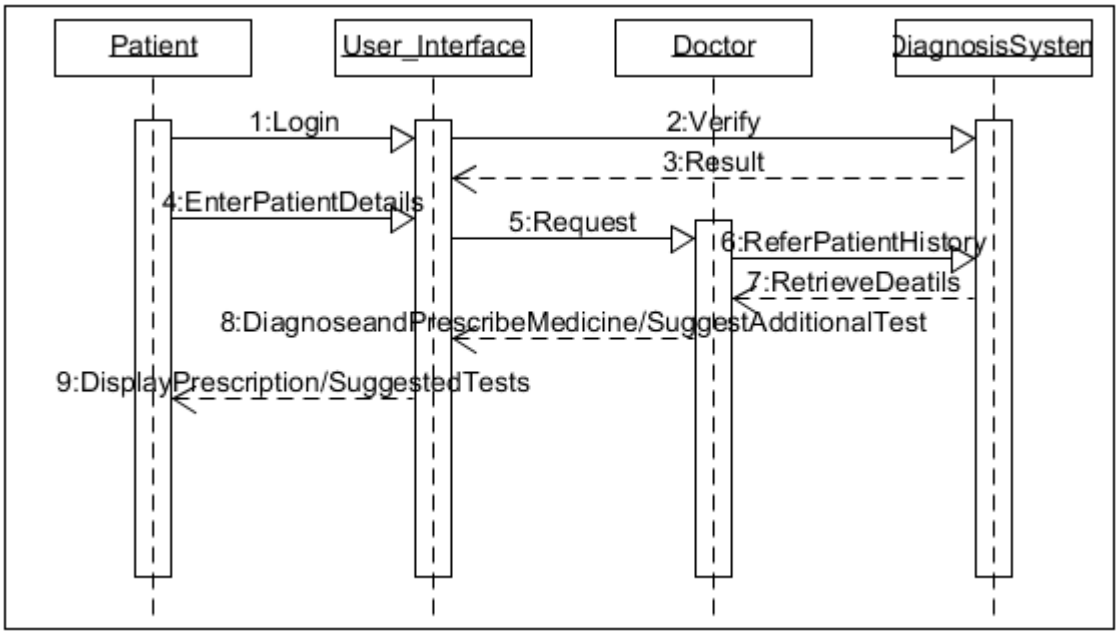


Figure 32: Sequence Diagram for Medical Consultation System [9]

3.5.4.1 Manual Testing of Example 4

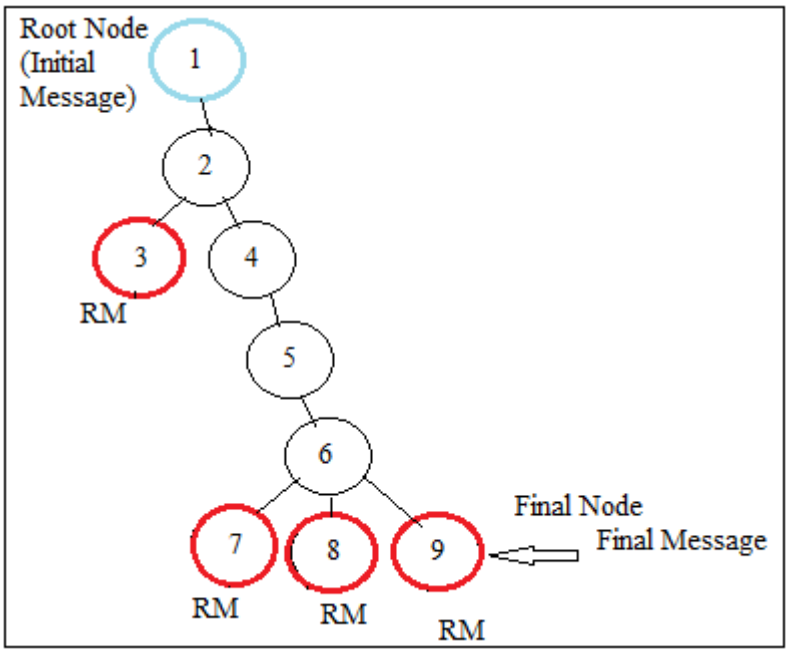


Figure 33: *The Single Message Sharing Tree (SMST)* of Figure 32

➢ ***Test Paths***

Path ID	Test Paths
1	1-2-3
2	1-2-4-5-6-7
3	1-2-4-5-6-8
4	1-2-4-5-6-9

Table 15: Sequence of All Test Paths in Figure 33

Path ID	Test Paths
1	*Login-Verify-Result*
2	*Login-Verify-EnterPatientDetails-Request-ReferPatientHistory-RetrieveDetails*
3	*Login-Verify-EnterPatientDetails-Request-ReferPatientHistory Diagnose_ and_ Prescribe Medicine/SuggestAdditionalTests*
4	*Login-Verify-EnterPatientDetails-Request-ReferPatientHistory-Display Prescription / SuggestedTests*

Table 16: List of All Test Paths Name of Figure 33

3.5.4.2 Automated Testing of Example 4

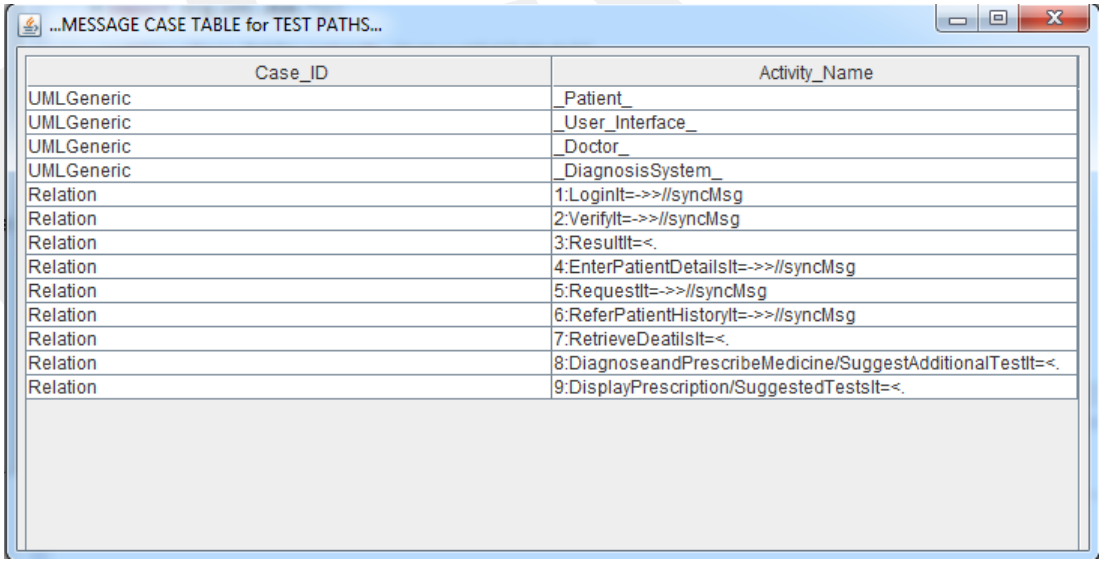


Figure 34: JTable of XML File in Appendix D

➢ All Results For Test Paths

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4 - 5 - 6 - 7]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9]

1. [1:Login - 2:Verify - 3:Result]
2. [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails - 5:Request -
6:ReferPatientHistory - 7:RetrieveDeatils]
3. [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails - 5:Request -
6:ReferPatientHistory - 7:RetrieveDeatils -
8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest]
4. [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails - 5:Request -
6:ReferPatientHistory - 7:RetrieveDeatils -
8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest -
9:DisplayPrescription/SuggestedTests]

BEST PATH

The Best Path: 4.  Path [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails -
5:Request - 6:ReferPatientHistory - 7:RetrieveDeatils -
8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest -
9:DisplayPrescription/SuggestedTests]

ALL MESSAGES

All messages of sequence diagram: [1:Login - 2:Verify - 3:Result -
4:EnterPatientDetails - 5:Request - 6:ReferPatientHistory - 7:RetrieveDeatils -
8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest -
9:DisplayPrescription/SuggestedTests]

Total message number of sequence diagram: 9
```

➢ Test Cases Table

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3	**1**	Login	Loading Info	-
		2	Verify	Verify Information	Result
		3	Result		
2	1-2-4-5-6-7	**4**	EnterPatientDetails	Loading Info	-
		5	Request	Return Result	-
		6	ReferPatientHistory	RetrieveDetails	RetrieveDetails
		7	RetrieveDetails		
3	1-2-4-5-6-8	**8**	Diagnose_and_Prescribe Medicine /Suggest AdditionalTests		
4	1-2-4-5-6-9	**9**	DisplayPrescription/ SuggestedTests		

Table 17: Test Cases Table of Example 4

3.5.5 Example5

The model is a facebook authorization. Example 5 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix E** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

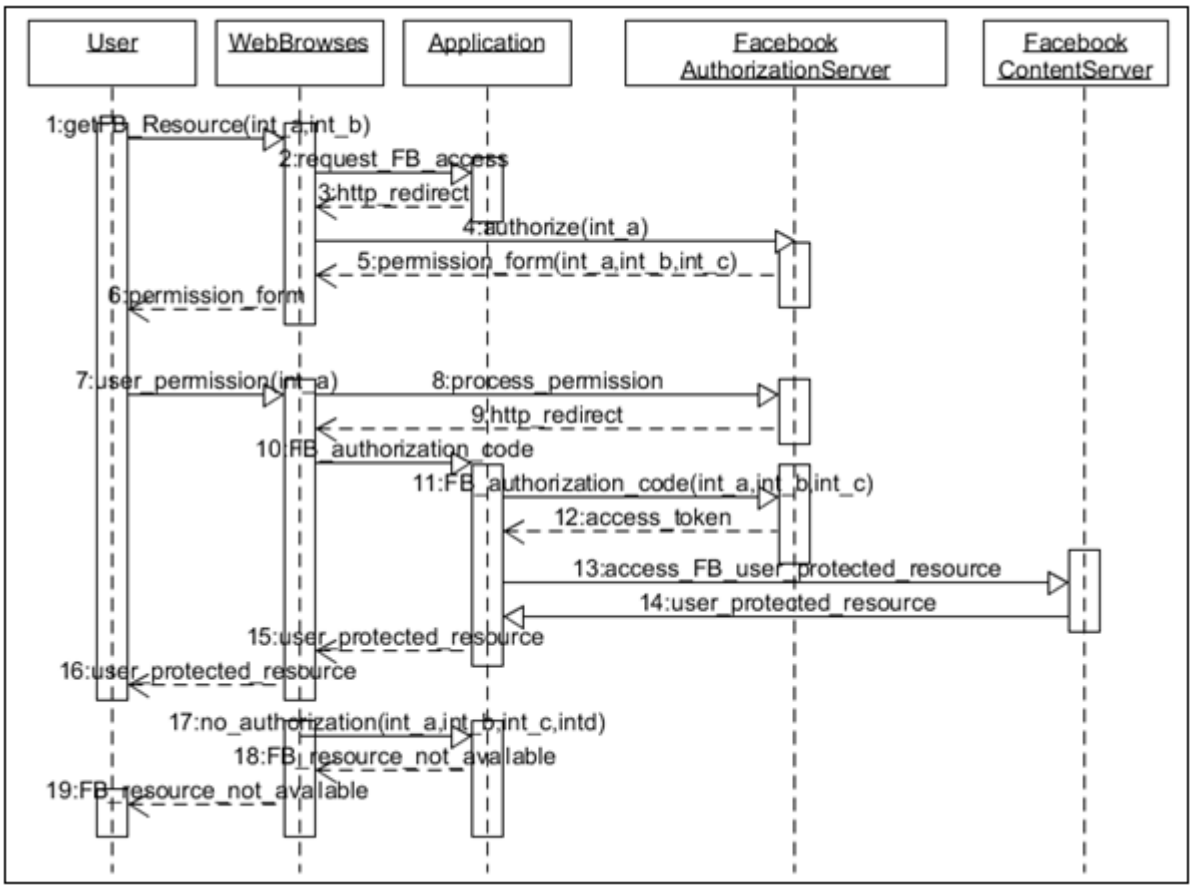


Figure 35: Sequence Diagram of FB Authorization [2]

3.5.5.1 Manual Testing of Example 5

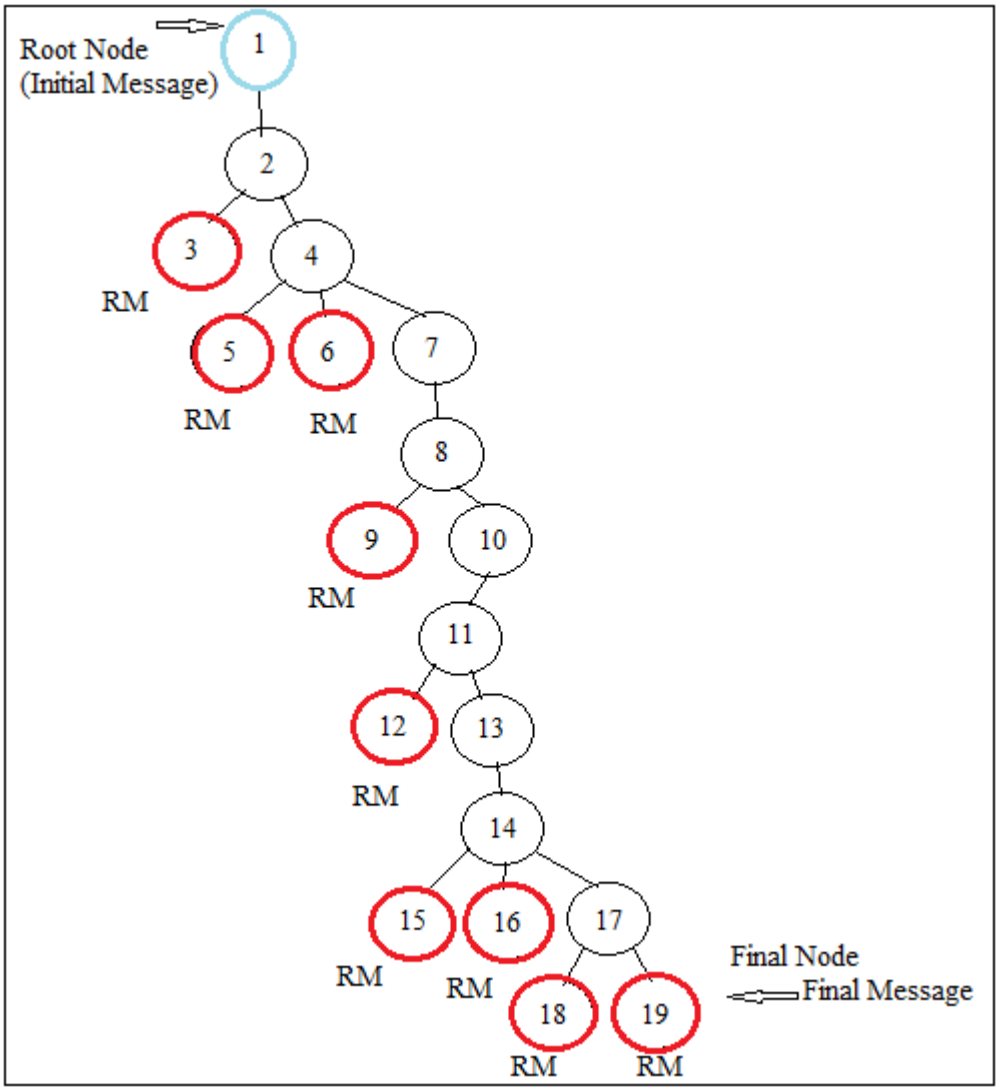


Figure 36: *The Single Message Sharing Tree (SMST)* of Figure 35

- ***Test Paths***

Path ID	Test Paths
1	1-2-3
2	1-2-4-5
3	1-2-4-6
4	1-2-4-7-8-9
5	1-2-4-7-8-10-11-12
6	1-2-4-7-8-10-11-13-14-15
7	1-2-4-7-8-10-11-13-14-16
8	1-2-4-7-8-10-11-13-14-17-18

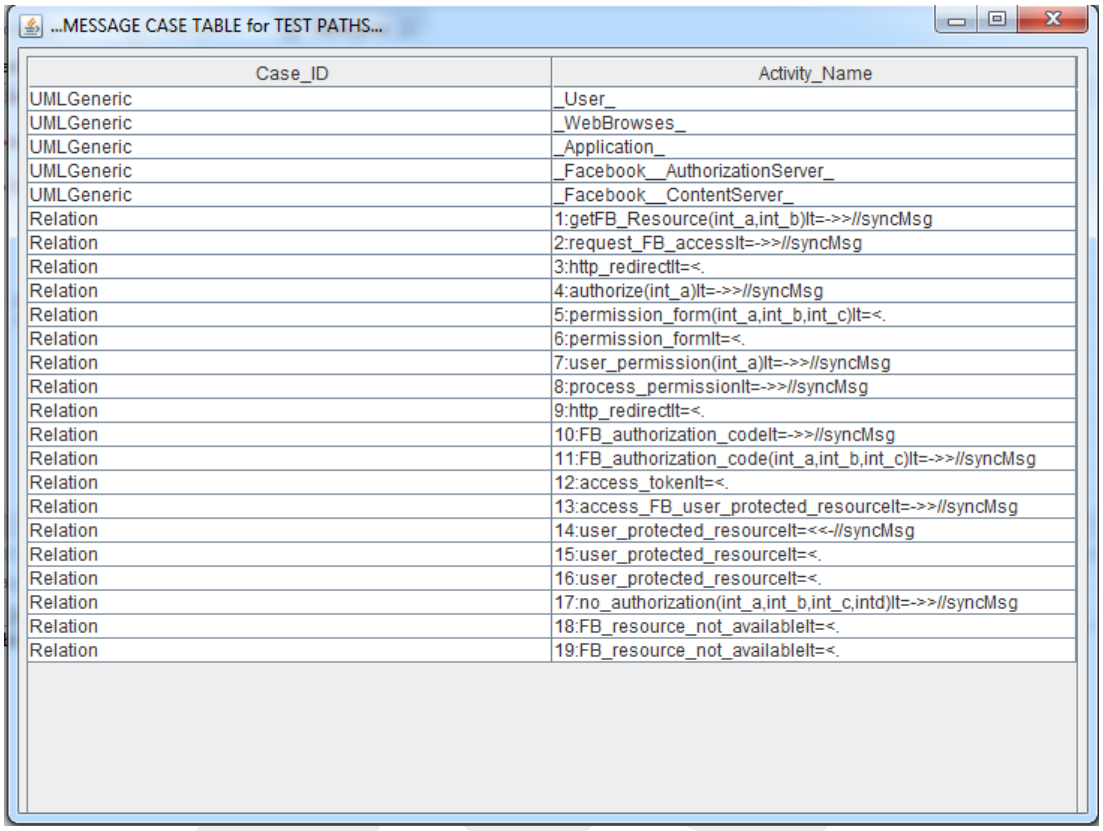
9	1-2-4-7-8-10-11-13-14-17-19

Table 18: Sequence of All Test Paths in Figure 36

Path ID	Test Paths
1	*getFB_Resource-request_FB_access-http_redirect*
2	*getFB_Resource-request_FB_access-http-authorize-permission_form*
3	*getFB_Resource-request_FB_access-http-authorize-permission_form*
4	*getFB_Resource-request_FB_access-http-authorize-user_permission-process_permission-http_redirect*
5	*getFB_Resource-request_FB_access-http-authorize-user_permission-process_permission-FB _ authorization_code- FB_authorization_code-access_token*
6	*getFB_Resource-request_FB_access-http-authorize-user_permission-process_permission-FB _ authorization_code- FB_authorization_code-access_FB_user_protected_resource-user _ protected _ resource-user_protected_resource*
7	*getFB_Resource-request_FB_access-http-authorize-user_permission-process_permission-FB _ authorization_code- FB_authorization_code-access_FB_user_protected_resource-user_ protected _resource-user_protected_resource*
8	*getFB_Resource-request_FB_access-http-authorize-user_permission-process_permission-FB_ authorization_code- FB_authorization_code-access_FB_user_protected_resource-user_protected _resource-no_authorization-FB_resource_not_available*
9	*getFB_Resource-request_FB_access-http-authorize-user_permission-process_permission-FB_ authorization_code- FB_authorization_code-access_FB_user_protected_resource-user_protected_ resource-no_authorization-FB_resource_not_available*

Table 19: List of All Test Paths Name of Figure 36

3.5.5.2 Automated Testing of Example 5



Case_ID	Activity_Name
UMLGeneric	_User_
UMLGeneric	_WebBrowses_
UMLGeneric	_Application_
UMLGeneric	_Facebook__AuthorizationServer_
UMLGeneric	_Facebook__ContentServer_
Relation	1:getFB_Resource(int_a,int_b)lt=->>//syncMsg
Relation	2:request_FB_accesslt=->>//syncMsg
Relation	3:http_redirectlt=<.
Relation	4:authorize(int_a)lt=->>//syncMsg
Relation	5:permission_form(int_a,int_b,int_c)lt=<.
Relation	6:permission_formlt=<.
Relation	7:user_permission(int_a)lt=->>//syncMsg
Relation	8:process_permissionlt=->>//syncMsg
Relation	9:http_redirectlt=<.
Relation	10:FB_authorization_codelt=->>//syncMsg
Relation	11:FB_authorization_code(int_a,int_b,int_c)lt=->>//syncMsg
Relation	12:access_tokenlt=<.
Relation	13:access_FB_user_protected_resourcelt=->>//syncMsg
Relation	14:user_protected_resourcelt=<<-//syncMsg
Relation	15:user_protected_resourcelt=<.
Relation	16:user_protected_resourcelt=<.
Relation	17:no_authorization(int_a,int_b,int_c,intd)lt=->>//syncMsg
Relation	18:FB_resource_not_availablelt=<.
Relation	19:FB_resource_not_availablelt=<.

Figure 37: JTable of XML File in Appendix E

- **All Results for Tests Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4 - 5]
3. [1 - 2 - 3 - 4 - 5 - 6]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18]
9. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19]


1. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect]
2. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c)]
3. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form]
```

4. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect]
5. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect - 10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token]
6. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect - 10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource - 15:user_protected_resource]
7. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect - 10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource - 15:user_protected_resource - 16:user_protected_resource]
8. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect - 10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource - 15:user_protected_resource - 16:user_protected_resource - 17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available]
9. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect - 10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource - 15:user_protected_resource - 16:user_protected_resource - 17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available - 19:18:FB_resource_not_available]

BEST PATH

The Best Path: 9. Path [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect - 10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource - 15:user_protected_resource - 16:user_protected_resource - 17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available - 19:18:FB_resource_not_available]

ALL MESSAGES

All messages of sequence diagram: [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a) - 8:process_permission - 9:http_redirect - 10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource - 15:user_protected_resource - 16:user_protected_resource - 17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available - 19:18:FB_resource_not_available]

Total message number of sequence diagram: 19

- **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3	**1**	get_FB_Resource	Return	-
		2	request_FB_access	http_redirect	http_redirect
		3	http_redirect		
2	1-2-4-5	**4**	authorize	Permission	permission_form
		5	permission_form		
3	1-2-4-6	**6**	permission_form		
4	1-2-4-7-8-9	**7**	user_permission	Permission	-
		8	process_permission	http_redirect	http_redirect
		9	http_redirect		
5	1-2-4-7-8-10-11-12	**10**	FB_authorization_code	Access	-
		11	FB_authorization_code	Access	access_token
		12	access_token		
6	1-2-4-7-8-10-11-13-14-15	**13**	access_FB_user_ protected_resource	ProtectedforAccess	-
		14	user_protected_resource	user_protected_ resource	user_protected_ resource
		15	user_protected_resource		
7	1-2-4-7-8-10-11-13-14-16	**16**	user_protected_resource		
8	1-2-4-7-8-10-11-13-14-17-18	**17**	no_authorization	FB_resource_not_ available	FB_resource_ not_available
		18	FB_resource_not_ available		
9	1-2-4-7-8-10-11-13-14-17-19	**19**	FB_resource_not_ available		

Table 20: Test Cases Table of Example 5

3.5.6 Example6

The model is a ATM banking system. Example 6 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix F** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

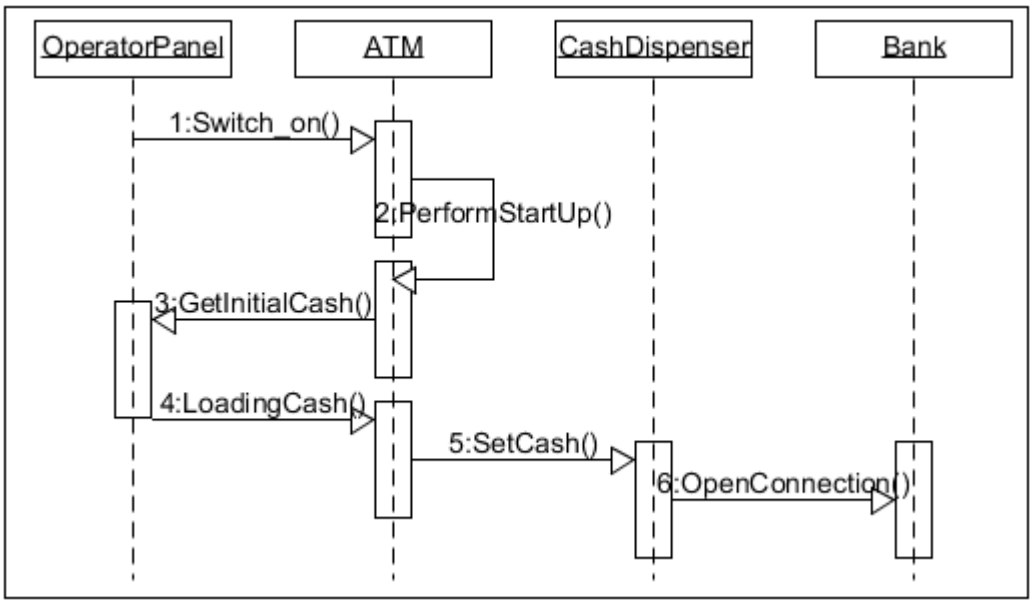


Figure 38: Sequence diagram for ATM Banking System [5]

3.5.6.1 Manual Testing of Example 6

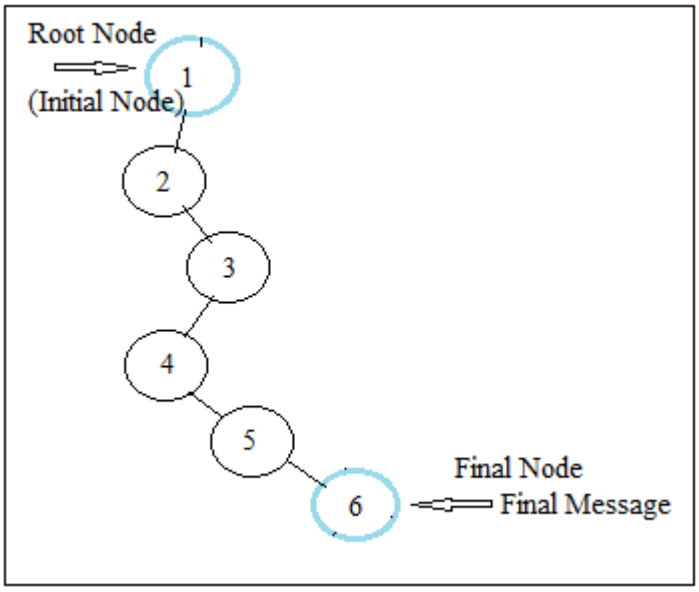


Figure 39: *The Single Message Sharing Tree (SMST)* of Figure 38

- ***Test Paths***

Path ID	Test Paths
1	1-2-3-4-5-6

Table 21: Sequence of All Test Paths in Figure 39

Path ID	Test Paths
1	*SwitchOn()-PerformStartUp()-GetInitialCash()-LoadingCash()-SetCash()-OpenConnection()*

Table 22: List of All Test Paths Name of Figure 39

3.5.6.2 Automated Testing of Example 6

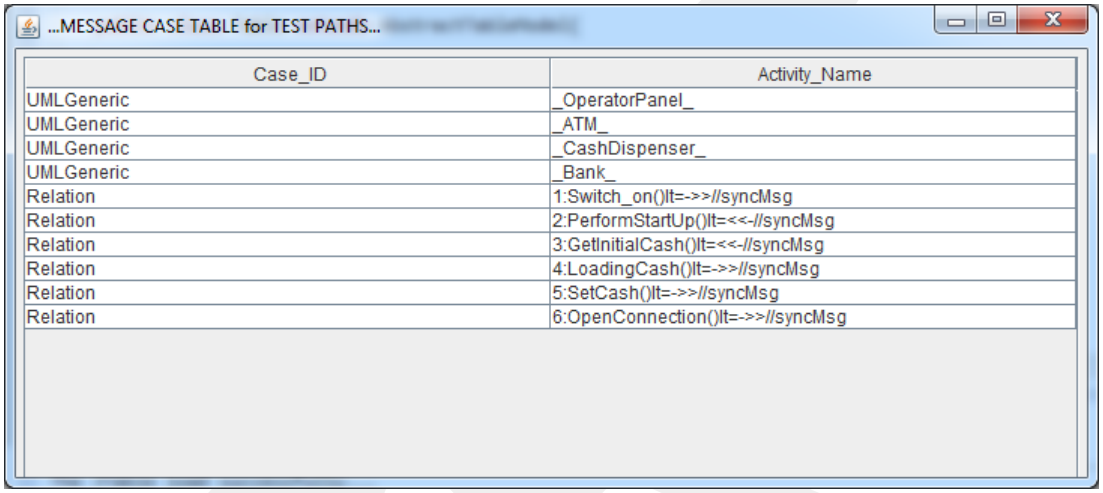


Figure 40: JTable of XML File in Appendix F

- **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3 - 4 - 5 - 6]

ALL MESSAGES

All messages of sequence diagram: [1:Switch_on() -> 2:PerformStartUp() ->
3:GetInitialCash() -> 4:LoadingCash() -> 5:SetCash() -> 6:OpenConnection()]

Total message number of sequence diagram: 6
```

➢ **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3-4-5-6	**1**	Switch_on	Load System	-
		2	PerformStartUp	Start Performing	-
		3	GetInitialCash	Loading Cash	-
		4	LoadingCash	Get Cash	-
		5	SetCash	Return	-
		6	OpenConnection	System Connection	-

Table 23: Test Cases Table of Example 6

3.5.7 Example7

The model is a bookstore composite web service. Example 7 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix G** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

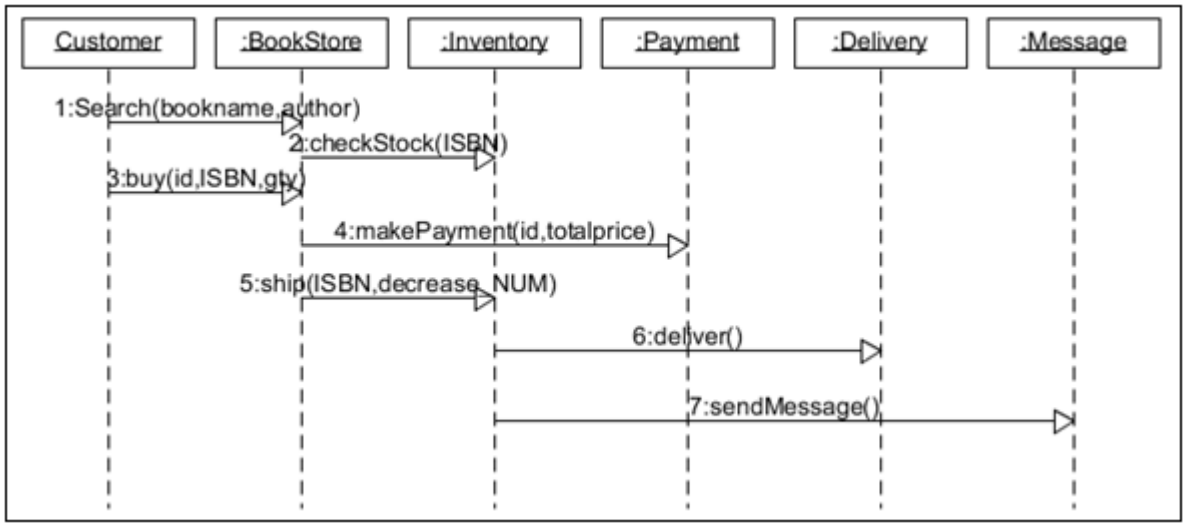


Figure 41: Sequence Diagram for Bookstore Composite Web Service [6]

3.5.7.1 Manual Testing of Example 7

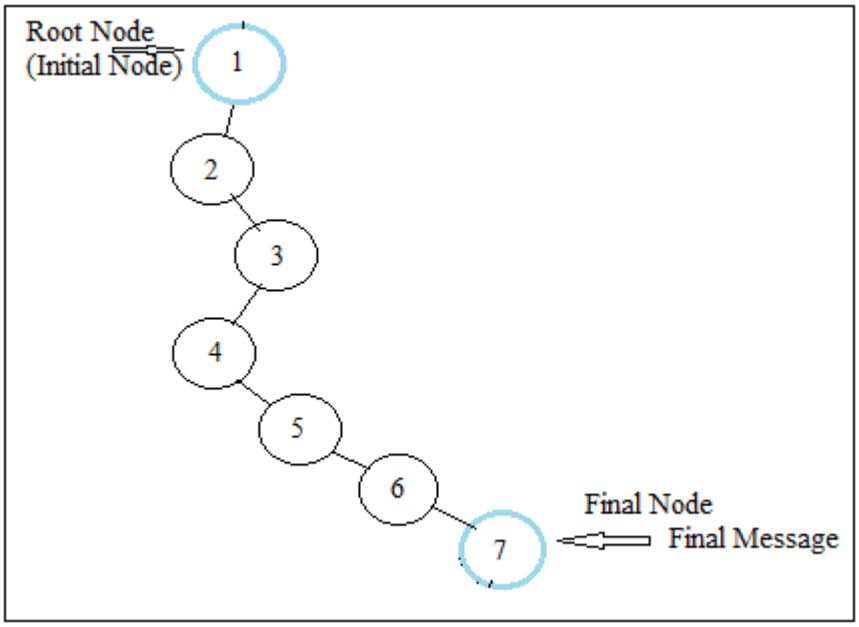


Figure 42: *The Single Message Sharing Tree (SMST)* of Figure 41

- ***Test Paths***

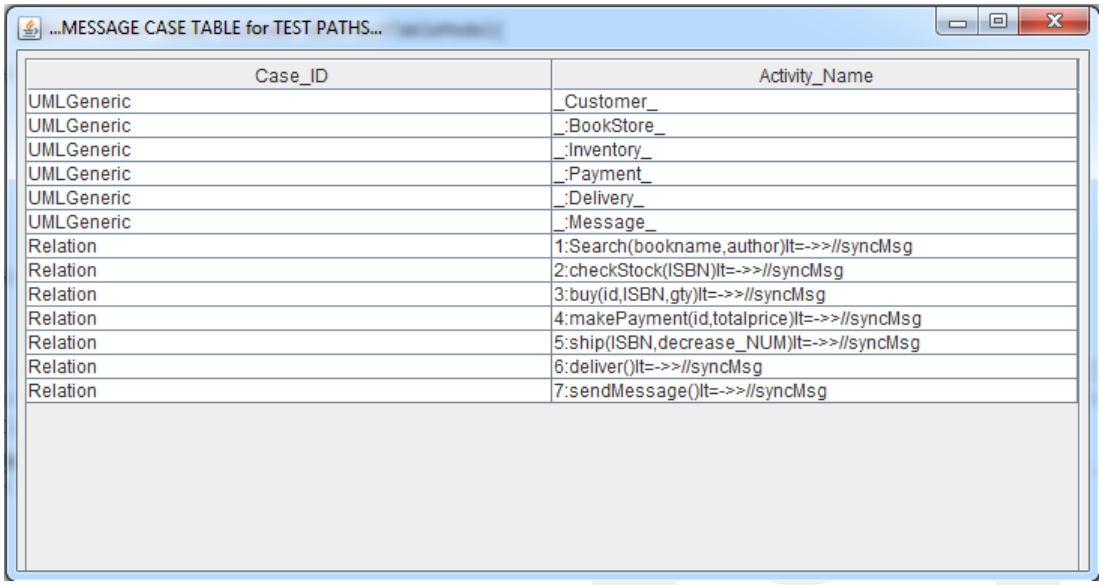
Path ID	Test Paths
1	1-2-3-4-5-6-7

Table 24: Sequence of All Test Paths in Figure 42

Path ID	Test Paths
1	*Search()-CheckStock()-Buy()-MakePayment()-Ship()-Deliver()-SendMessage()*

Table 25: List of All Test Paths Name of Figure 42

3.5.7.2 Automated Testing of Example 7



Case_ID	Activity_Name
UMLGeneric	_Customer_
UMLGeneric	_:BookStore_
UMLGeneric	_:Inventory_
UMLGeneric	_:Payment_
UMLGeneric	_:Delivery_
UMLGeneric	_:Message_
Relation	1:Search(bookname,author)lt=->>//syncMsg
Relation	2:checkStock(ISBN)lt=->>//syncMsg
Relation	3:buy(id,ISBN,gty)lt=->>//syncMsg
Relation	4:makePayment(id,totalprice)lt=->>//syncMsg
Relation	5:ship(ISBN,decrease_NUM)lt=->>//syncMsg
Relation	6:deliver()lt=->>//syncMsg
Relation	7:sendMessage()lt=->>//syncMsg

Figure 43: JTable of XML File in Appendix G

➢ **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3 - 4 - 5 - 6 - 7]

ALL MESSAGES

All messages of sequence diagram: [1:Search(bookname,author) -> 2:checkStock(ISBN)
-> 3:buy(id,ISBN,gty) -> 4:makePayment(id,totalprice) -> 5:ship(ISBN,decrease_NUM)
-> 6:deliver() -> 7:sendMessage()]

Total message number of sequence diagram: 7
```

➢ **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3-4-5-6-7	1	Search	Return Result	-
		2	CheckStock	Return Result	-
		3	Buy	Return	-
		4	MakePayment	Payment	-
		5	Ship	Choose Shipping Options	-
		6	Deliver	Return Info	-
		7	SendMessage	Return Message	-

Table 26: Test Cases Table of Example 7

3.5.8 Example8

The model is a library management system. Example 8 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix H** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

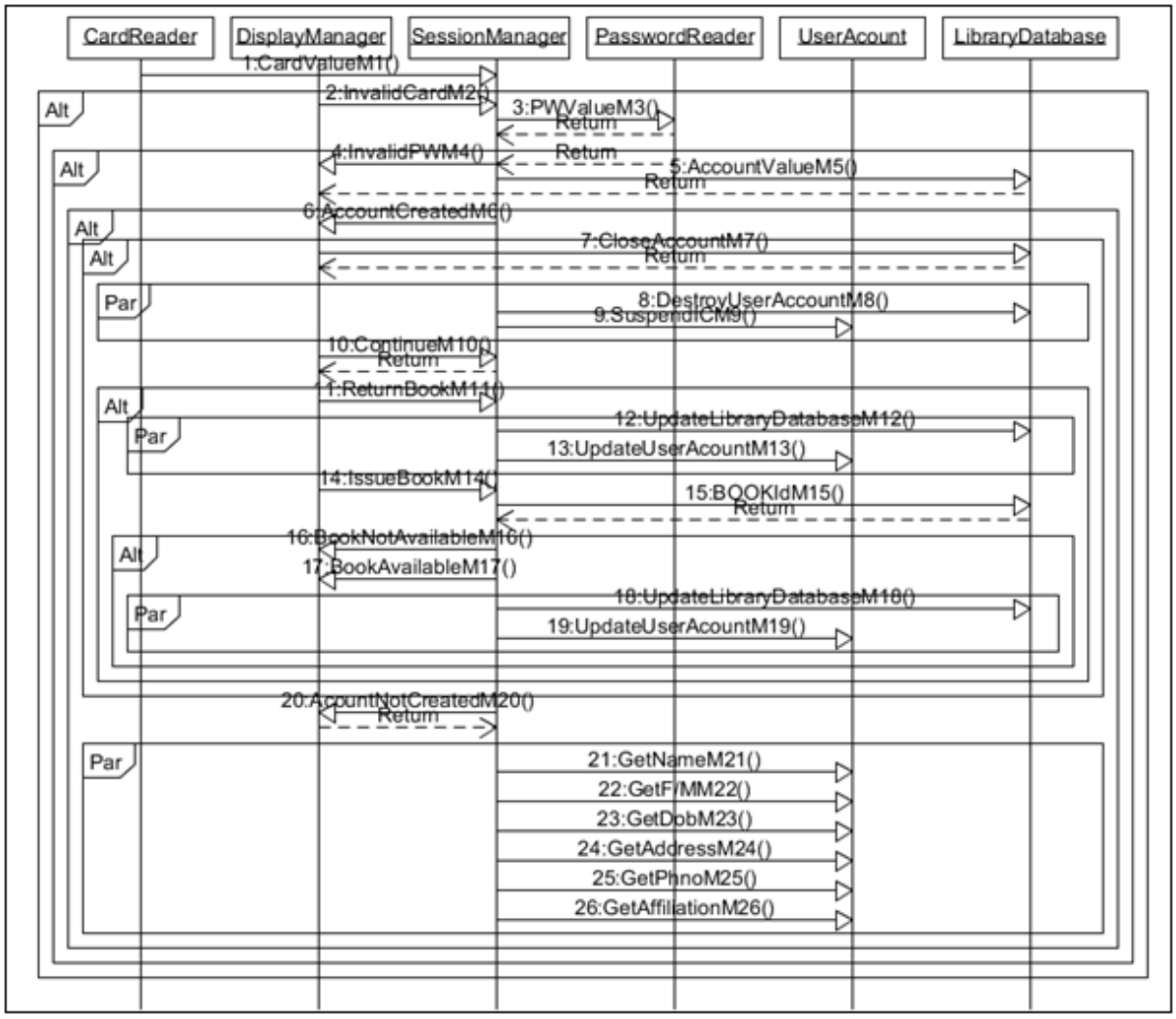


Figure 44: Sequence Diagram of Library Management System [4]

3.5.8.1 Manual Testing of Example 8

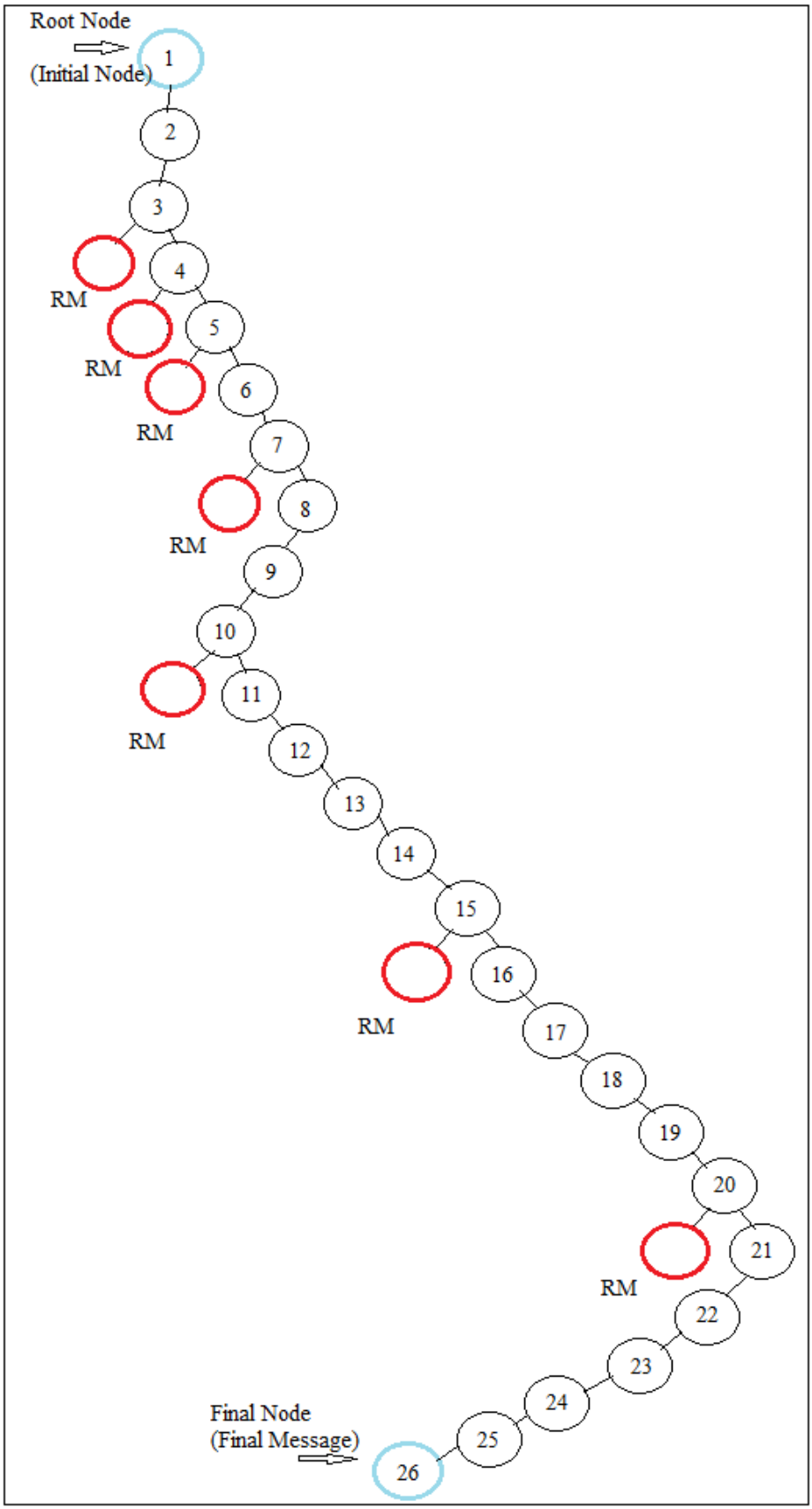


Figure 45: *The Single Message Sharing Tree (SMST)* of Figure 44

- ***Test Paths***

Path ID	Test Paths
1	1-2-3
2	1-2-3-4
3	1-2-3-4-5
4	1-2-3-4-5-6-7
5	1-2-3-4-5-6-7-8-9-10
6	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15
7	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-8-19-20
8	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-8-19-20-21-22-23-24-25-26

Table 27: Sequence of All Test Paths in Figure 45

Path ID	Test Paths
1	*CardValue-InvalidCard-PWValue*
2	*CardValue-InvalidCard-PWValue-InvalidPW*
3	*CardValue-InvalidCard-PWValue-InvalidPW-AccountValue*
4	*CardValue-InvalidCard-PWValue-InvalidPW-AccountValue-AccountCreated-CloseAccount*
5	*CardValue-InvalidCard-PWValue-InvalidPW-AccountValue-AccountCreated-CloseAccount-Destroy UserAccount-SuspendIC-Continue-*
6	*CardValue-InvalidCard-PWValue-InvalidPW-AccountValue-AccountCreated-CloseAccount-Destroy UserAccount-SuspendIC-Continue-ReturnBook-UpdateLibraryDatabase-UpdateUserAccount-IssueBook-BookID*
7	*CardValue-InvalidCard-PWValue-InvalidPW-AccountValue-AccountCreated-CloseAccount-Destroy UserAccount-SuspendIC-Continue-ReturnBook-UpdateLibraryDatabase-UpdateUserAccount-IssueBook-BookID-BookNotAvailable-BookAvailable-UpdateLibraryDatabase-UpdateUserAccount-AccountNotCreated*
8	*CardValue-InvalidCard-PWValue-InvalidPW-AccountValue-AccountCreated-CloseAccount-Destroy UserAccount-SuspendIC-Continue-ReturnBook-UpdateLibraryDatabase-UpdateUserAccount-Issue Book-BookID-BookNotAvailable-BookAvailable-UpdateLibraryDatabase-UpdateUserAccount-AccountNotCreated-GetName-GetF/M-GetDob-GetAddress-GetPhno-GetAffiliation*

Table 28: List of All Test Paths Name of Figure 45

3.5.8.2 Automated Testing of Example 8

...MESSAGE CASE TABLE for TEST PATHS...

Case_ID	Activity_Name
UMLGeneric	_CardReader_
UMLGeneric	_DisplayManager_
UMLGeneric	_SessionManager_
UMLGeneric	_PasswordReader_
UMLGeneric	_UserAcount_
UMLGeneric	_LibraryDatabase_
UMLFrame	Alt
Relation	1:CardValueM1()lt=->>//syncMsg
Relation	2:InvalidCardM2()lt=->>//syncMsg
Relation	3:PWValueM3()lt=->>//syncMsg
Relation	ReturnIt=<.
UMLFrame	Alt
Relation	4:InvalidPWM4()lt=<<-//syncMsg
Relation	ReturnIt=<.
Relation	5:AccountValueM5()lt=->>//syncMsg
Relation	ReturnIt=<.
UMLFrame	Alt
Relation	6:AccountCreatedM6()lt=<<-//syncMsg
UMLFrame	Alt
Relation	7:CloseAccountM7()lt=->>//syncMsg
Relation	ReturnIt=<.
UMLFrame	Par
Relation	8:DestroyUserAccountM8()lt=->>//syncMsg
Relation	9:SuspendICM9()lt=->>//syncMsg
Relation	10:ContinueM10()lt=->>//syncMsg
Relation	ReturnIt=<.
UMLFrame	Alt
Relation	11:ReturnBookM11()lt=->>//syncMsg
UMLFrame	Par
Relation	12:UpdateLibraryDatabaseM12()lt=->>//syncMsg
Relation	13:UpdateUserAcountM13()lt=->>//syncMsg
Relation	14:IssueBookM14()lt=->>//syncMsg
Relation	15:BOOKIdM15()lt=->>//syncMsg
Relation	ReturnIt=<.
UMLFrame	Alt
Relation	16:BookNotAvailableM16()lt=<<-//syncMsg
Relation	17:BookAvailableM17()lt=<<-//syncMsg
UMLFrame	Par
Relation	18:UpdateLibraryDatabaseM18()lt=->>//syncMsg
Relation	19:UpdateUserAcountM19()lt=->>//syncMsg
Relation	20:AcountNotCreatedM20()lt=<<-//syncMsg
Relation	ReturnIt=.>
UMLFrame	Par
Relation	21:GetNameM21()lt=->>//syncMsg
Relation	22:GetF/MM22()lt=->>//syncMsg
Relation	23:GetDobM23()lt=->>//syncMsg
Relation	24:GetAddressM24()lt=->>//syncMsg
Relation	25:GetPhnoM25()lt=->>//syncMsg
Relation	26:GetAffiliationM26()lt=->>//syncMsg

Figure 46: JTable of XML File in Appendix H

- **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4]
3. [1 - 2 - 3 - 4 - 5]
```

```
4. [1 - 2 - 3 - 4 - 5 - 6 - 7]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19 - 20]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26]

1. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return]
2. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return]
3. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return]
4. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return]
5. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return]
6. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return]
7. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return - 16:BookNotAvailableM16() -
17:BookAvailableM17() - 18:UpdateLibraryDatabaseM18() - 19:UpdateUserAcountM19() -
20:AcountNotCreatedM20() - Return]
8. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return - 16:BookNotAvailableM16() -
17:BookAvailableM17() - 18:UpdateLibraryDatabaseM18() - 19:UpdateUserAcountM19() -
20:AcountNotCreatedM20() - Return - 21:GetNameM21() - 22:GetF/MM22() -
23:GetDobM23() - 24:GetAddressM24() - 25:GetPhnoM25() - 26:GetAffiliationM26()]
BEST PATH

The Best Path: 8.  Path [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() -
Return - 4:InvalidPWM4() - Return - 5:AccountValueM5() - Return -
6:AccountCreatedM6() - 7:CloseAccountM7() - Return - 8:DestroyUserAccountM8() -
9:SuspendICM9() - 10:ContinueM10() - Return - 11:ReturnBookM11() -
12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() - 14:IssueBookM14() -
15:BOOKIdM15() - Return - 16:BookNotAvailableM16() - 17:BookAvailableM17() -
18:UpdateLibraryDatabaseM18() - 19:UpdateUserAcountM19() - 20:AcountNotCreatedM20()
- Return - 21:GetNameM21() - 22:GetF/MM22() - 23:GetDobM23() - 24:GetAddressM24() -
25:GetPhnoM25() - 26:GetAffiliationM26()]

ALL MESSAGES

All messages of sequence diagram: [1:CardValueM1() -> 2:InvalidCardM2() ->
3:PWValueM3() -> Return -> 4:InvalidPWM4() -> Return -> 5:AccountValueM5() ->
Return -> 6:AccountCreatedM6() -> 7:CloseAccountM7() -> Return ->
8:DestroyUserAccountM8() -> 9:SuspendICM9() -> 10:ContinueM10() -> Return ->
11:ReturnBookM11() -> 12:UpdateLibraryDatabaseM12() -> 13:UpdateUserAcountM13() ->
14:IssueBookM14() -> 15:BOOKIdM15() -> Return -> 16:BookNotAvailableM16() ->
17:BookAvailableM17() -> 18:UpdateLibraryDatabaseM18() -> 19:UpdateUserAcountM19()
-> 20:AcountNotCreatedM20() -> Return -> 21:GetNameM21() -> 22:GetF/MM22() ->
23:GetDobM23() -> 24:GetAddressM24() -> 25:GetPhnoM25() -> 26:GetAffiliationM26()]

Total message number of sequence diagram: 33
```

- **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3	**1**	CardValue	Loading Value	-
		2	InvalidCard	Valid Value	-
		3	PW_Value	Return	Return
2	1-2-3-4	**4**	InvalidPW	ValidPW	Return
3	1-2-3-4-5	**5**	AccountValue	Return Value	Return
4	1-2-3-4-5-6-7	**6**	AccountCreated	Loading Info	-
		7	CloseAccount	Out	Return
5	1-2-3-4-5-6-7-8-9-10	**8**	DestroyUserAccount	Return	-
		9	SuspendIC	Return	-
		10	Continue	Return	Return
6	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15	**11**	ReturnBook	Loading	-
		12	UpdateLibrary Database	Return Updated	-
		13	UpdateUserAccount	Return Updated	-
		14	IssueBook	Return Book Issue	-
		15	BOOKId	Return Book ID	Return
7	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-8-19-20	**16**	BookNotAvailable	Return Not Available	-
		17	BookAvailable	Return Available	-
		18	UpdateLibrary Database	Return Updated	-
		19	UpdateUserAccount	Return Updated	-
		20	AccountNotCreated	Check Info	Return
8	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-8-19-20-21-22-23-24-25-26	**21**	GetName	Loading Name	-
		22	GetF/M	Loading F/M	-
		23	GetDob	Loading Dob	-
		24	GetAddress	Loading Address	-
		25	GetPhno	Loading Phno	-
		26	GetAffiliation	Loading Affiliation	-

Table 29: Test Cases Table of Example 8

3.5.9 Example9

The model is a withdrawal usecase. Example 9 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix I** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

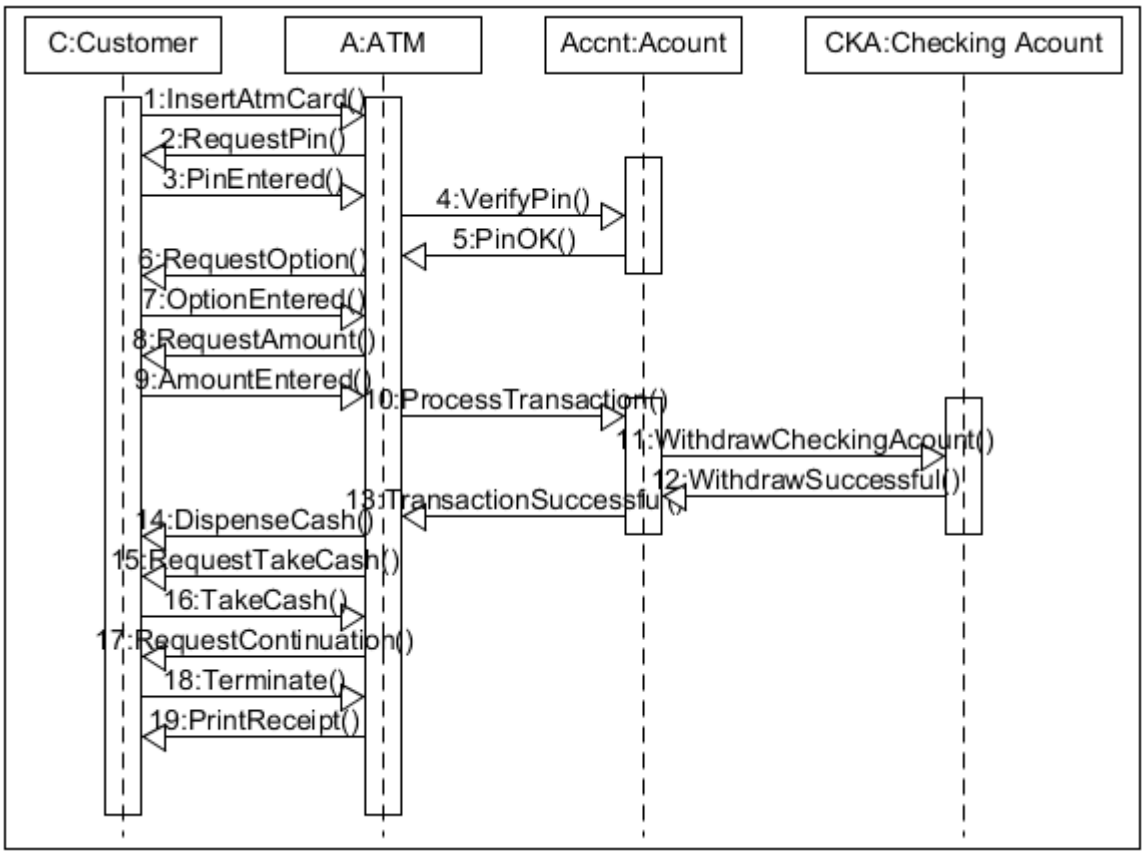


Figure 47: Sequence diagram for Withdrawal Usecase [7]

3.5.9.1 Manual Testing of Example 9

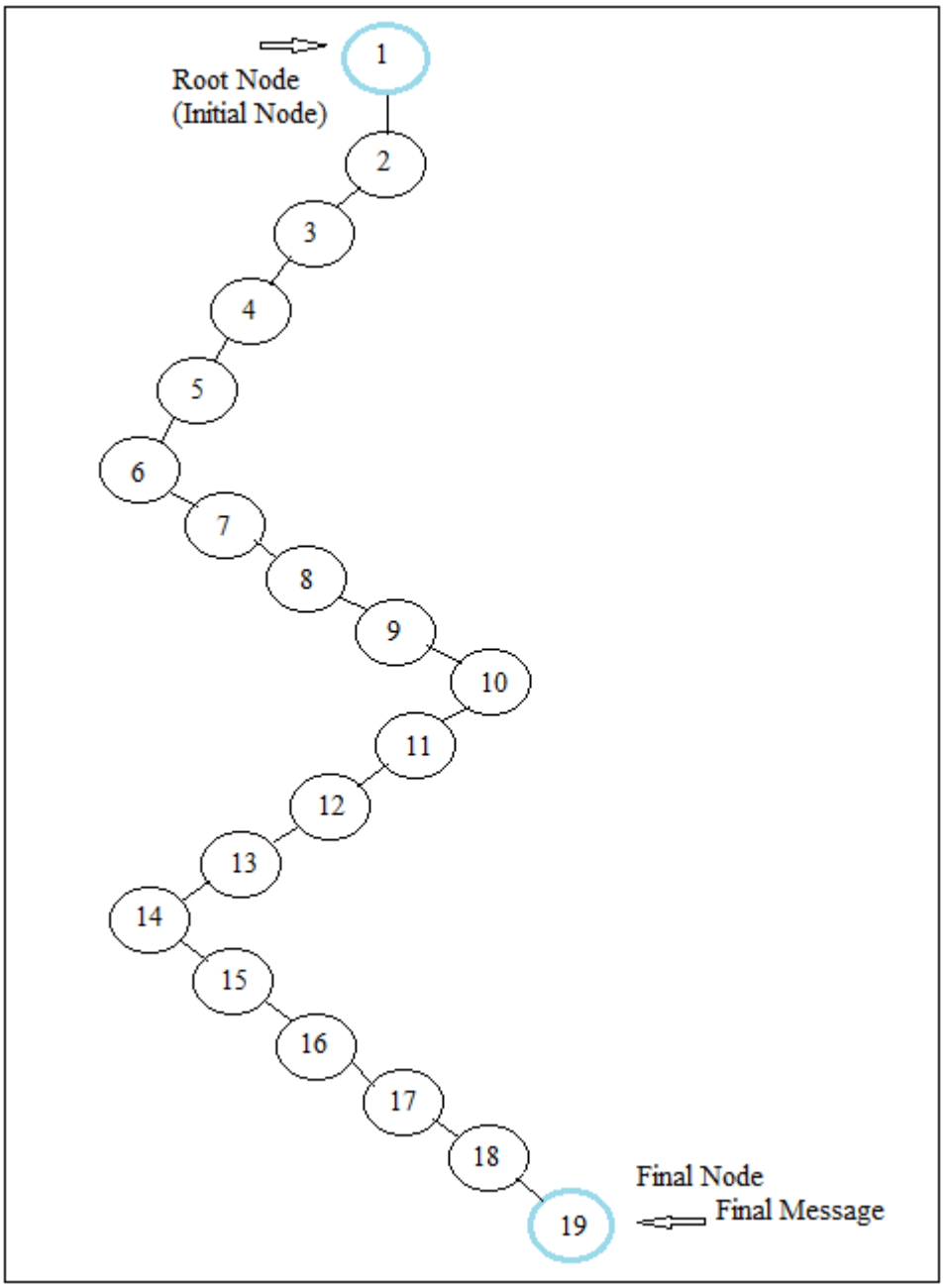


Figure 48: *The Single Message Sharing Tree (SMST)* of Figure 47

- ***Test Paths***

Path ID	Test Paths
1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19

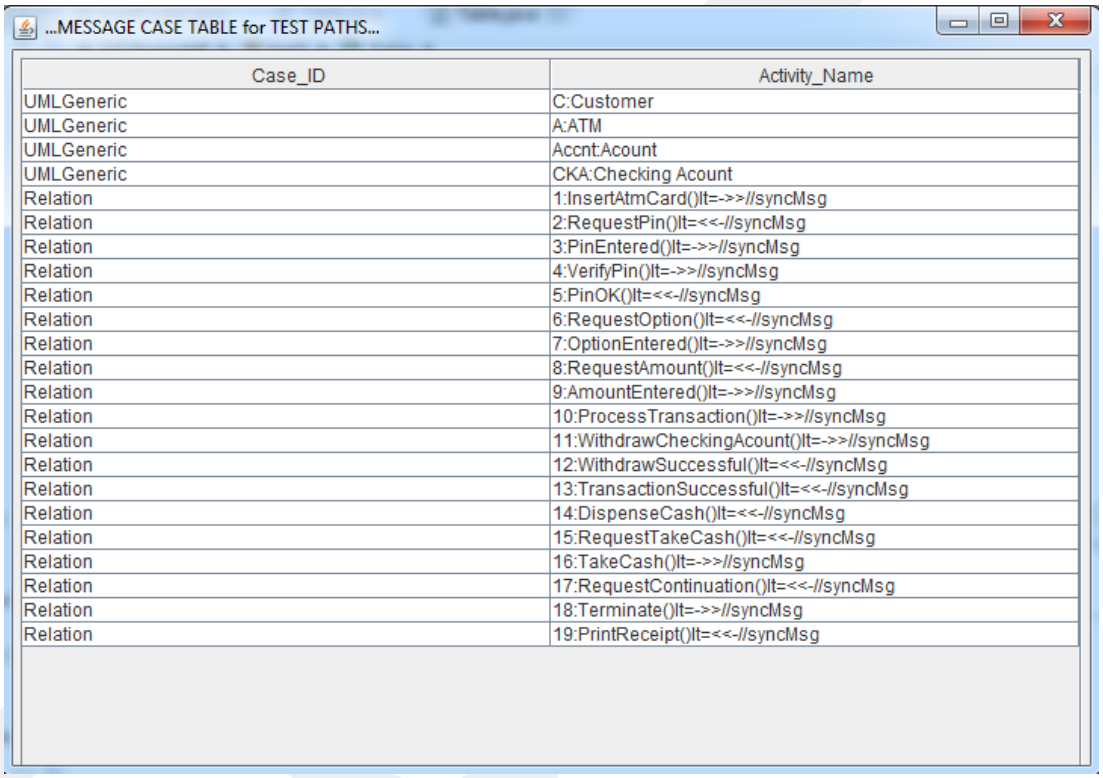
Table 30: Sequence of All Test Paths in Figure 48

Path ID	Test Paths

1	*Insert_ATM_Card()-RequestPin()-PinEntered()-VerifyPin()-PinOK()-RequestOption()-Option Entered ()-RequestAmount()-AmountEntered()-ProcessTransaction()-Withdraw CheckingAcount()-Withdraw Successful()-TransactionSuccessful()-Dispense Cash()-RequestTakeCash()-TakeCash()-RequestContinuation()-Terminate()-Print Receipt()*

Table 31: List of All Test Paths Name of Figure 48

3.5.9.2 Automated Testing of Example 9



Case_ID	Activity_Name
UMLGeneric	C:Customer
UMLGeneric	A:ATM
UMLGeneric	Accnt:Acount
UMLGeneric	CKA:Checking Acount
Relation	1:InsertAtmCard()lt=->>//syncMsg
Relation	2:RequestPin()lt=<<-//syncMsg
Relation	3:PinEntered()lt=->>//syncMsg
Relation	4:VerifyPin()lt=->>//syncMsg
Relation	5:PinOK()lt=<<-//syncMsg
Relation	6:RequestOption()lt=<<-//syncMsg
Relation	7:OptionEntered()lt=->>//syncMsg
Relation	8:RequestAmount()lt=<<-//syncMsg
Relation	9:AmountEntered()lt=->>//syncMsg
Relation	10:ProcessTransaction()lt=->>//syncMsg
Relation	11:WithdrawCheckingAcount()lt=->>//syncMsg
Relation	12:WithdrawSuccessful()lt=<<-//syncMsg
Relation	13:TransactionSuccessful()lt=<<-//syncMsg
Relation	14:DispenseCash()lt=<<-//syncMsg
Relation	15:RequestTakeCash()lt=<<-//syncMsg
Relation	16:TakeCash()lt=->>//syncMsg
Relation	17:RequestContinuation()lt=<<-//syncMsg
Relation	18:Terminate()lt=->>//syncMsg
Relation	19:PrintReceipt()lt=<<-//syncMsg

Figure 49: JTable of XML File in Appendix I

- **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19]

ALL MESSAGES

All messages of sequence diagram: [1:InsertAtmCard() -> 2:RequestPin() ->
3:PinEntered() -> 4:VerifyPin() -> 5:PinOK() -> 6:RequestOption() ->
7:OptionEntered() -> 8:RequestAmount() -> 9:AmountEntered() ->
10:ProcessTransaction() -> 11:WithdrawCheckingAcount() -> 12:WithdrawSuccessful() -
> 13:TransactionSuccessful() -> 14:DispenseCash() -> 15:RequestTakeCash() ->
16:TakeCash() -> 17:RequestContinuation() -> 18:Terminate() -> 19:PrintReceipt()]

Total message number of sequence diagram: 19
```

- **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19	**1**	InsertATMCard	Enter System	-
		2	RequestPin	Load Pin	-
		3	PinEntered	Return Pin	-
		4	VerifyPin	Valid Pin	-
		5	PinOK	Return	-
		6	RequestOption	Return Option	-
		7	OptionEntered	Select Option	-
		8	RequestAmount	Load Amount	-
		9	AmountEntered	Return Amount	-
		10	ProcessTransaction	Return Transaction	-
		11	WithdrawCheckingAcount	Control Account	-
		12	WithdrawSuccessful	Withdraw	-
		13	TransactionSuccessful	Return	-
		14	DispenseCash	Return	-
		15	RequestTakeCash	Return	-
		16	TakeCash	Return Payment	-
		17	RequestContinuation	Return	-
		18	Terminate	System Load	-
		19	PrintReceipt	Return Receipt	-

Table 32: Test Cases Table of Example 9

3.5.10 Example10

The model is a operation getRegistered(). Example 10 shows tree method and test paths results of proposed manual testing using our algorithms. Also, this example shows jtable and test paths results of proposed automated testing method using '*Table'* and '*JUnitCode'* programming codes. **Appendix J** includes XML file of automated testing. Also, a general test cases table is achieved with these methods.

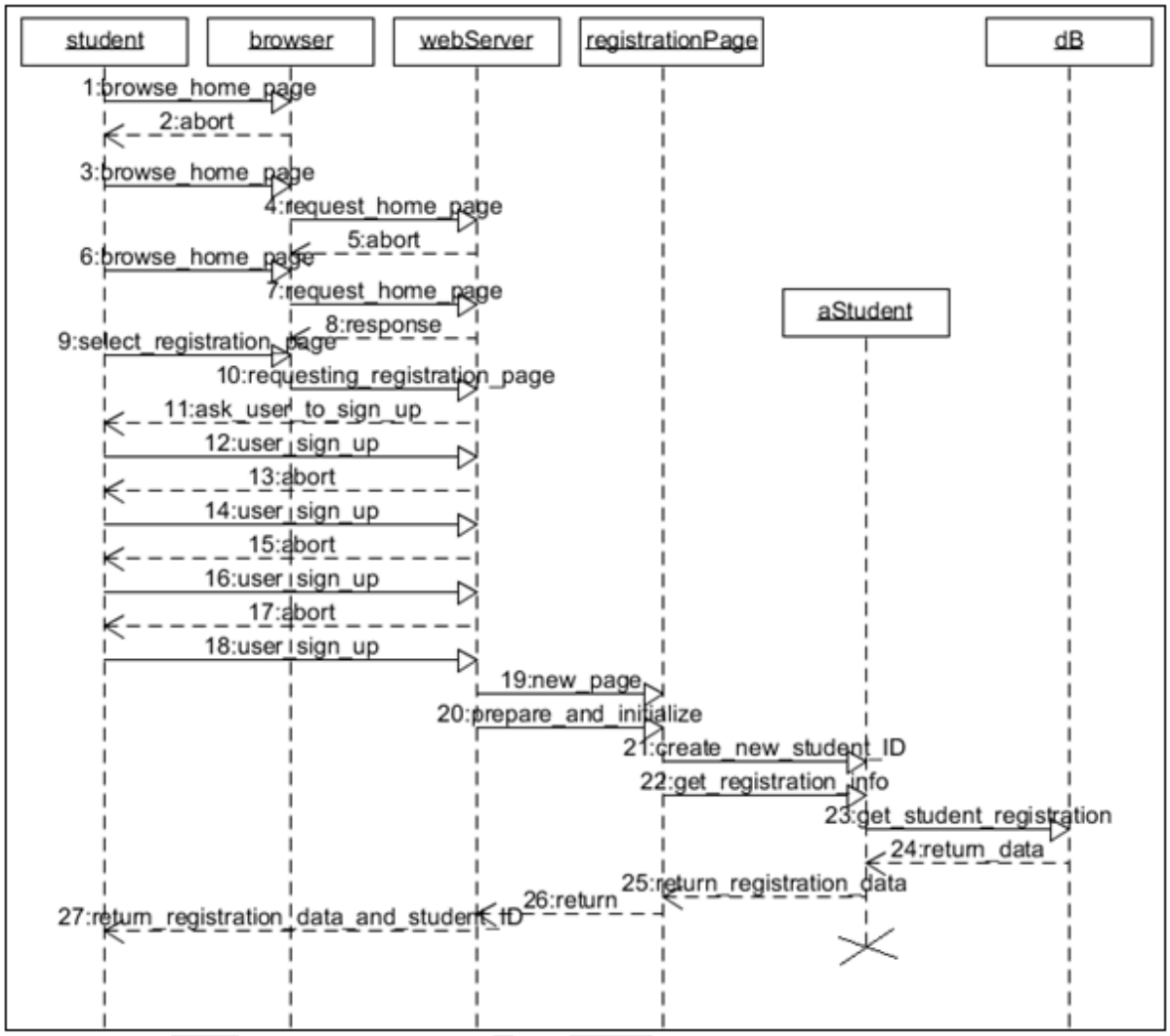


Figure 50: Sequence Diagram for Operation getRegistered() [8]

3.5.10.1 Manual Testing of Example 10

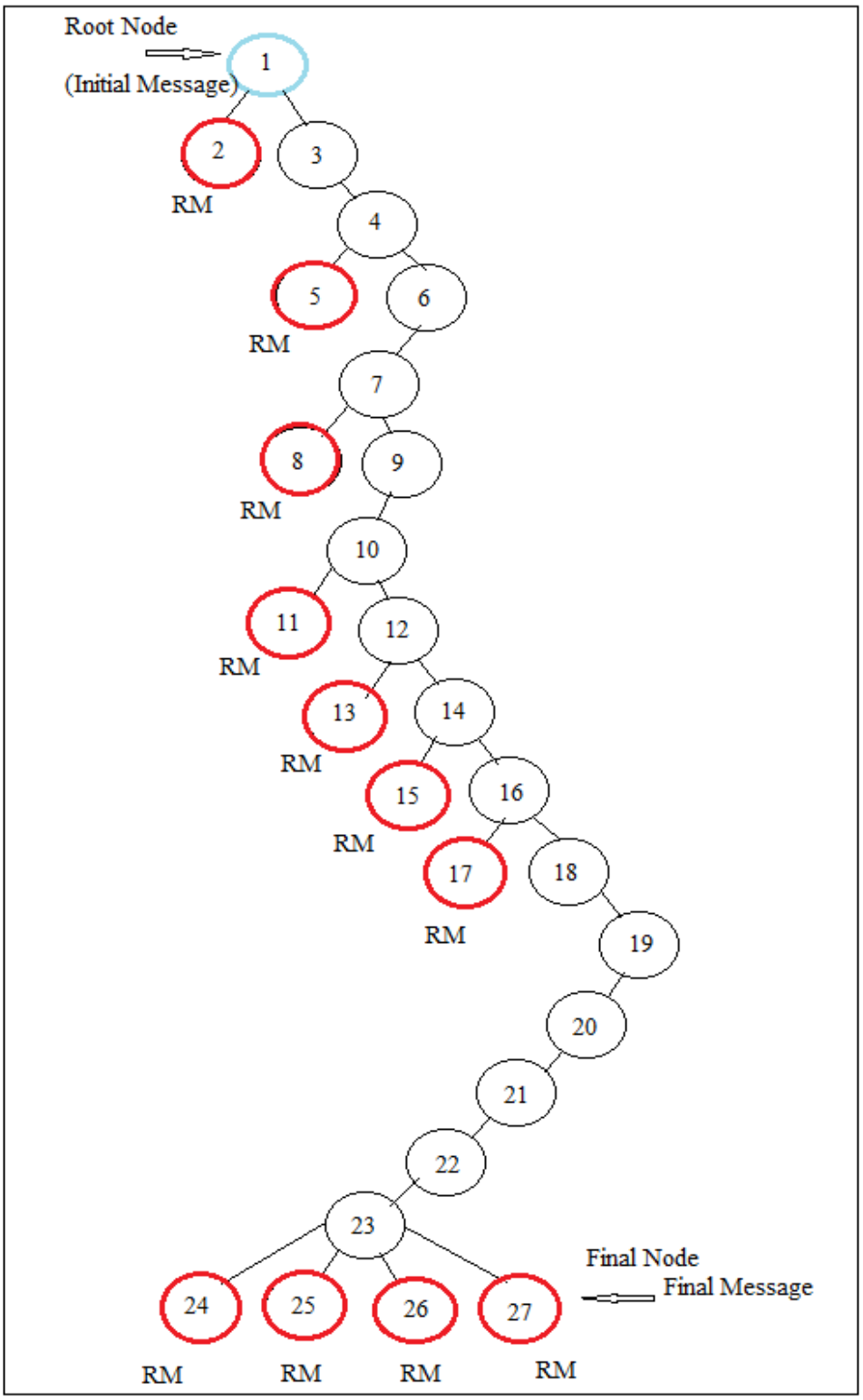


Figure 51: *The Single Message Sharing Tree (SMST)* of Figure 50

- ***Test Paths***

Path ID	Test Paths
1	1-2
2	1-3-4-5
3	1-3-4-6-7-8
4	1-3-4-6-7-9-10-11
5	1-3-4-6-7-9-10-12-13
6	1-3-4-6-7-9-10-12-14-15
7	1-3-4-6-7-9-10-12-14-16-17
8	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-24
9	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-25
10	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-26
11	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-27

Table 33: Sequence of All Test Paths in Figure 51

Path ID	Test Paths
1	*browse_home_page-abort*
2	*browse_home_page-browse_home_page-request_home_page-abort*
3	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home_ page -response*
4	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home_ page -select_registration_page-requesting_registration_page-ask_user_to_sign_up*
5	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home _page -select_registration_page-requesting_registration_page-user_sign_up-abort*
6	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home_ page -select_registration_page-requesting_registration_page-user_sign_up-user_sign_up-abort*
7	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home_ page -select_registration_page-requesting_registration_page-user_sign_up-user_sign_up-user_sign _ up-abort*
8	*browse_home_page-browse_home_page-request_home_page- browse_home_page-*

	request_home _ page-select_registration_page-requesting_registration_page-user_sign_up-user_sign_up-user_sign _ up-user_sign_up-new_page-prepare_and_initialize-create_new_student_ID-get_registration_info-get _student_registration-return_data
9	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home_ page-select_registration_page-requesting_registration_page-user_sign_up-user_sign_up-user_sign_ up-user_sign_up-new_page-prepare_and_initialize-create_new_student_ID-get_registration_info-get _student_registration-return_registration_data*
10	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home_ page-select_registration_page-requesting_registration_page-user_sign_up-user_sign_up-user_sign_ up-user_sign_up-new_page-prepare_and_initialize-create_new_student_ID-get_registration_info-get_ student_registration-return*
11	*browse_home_page-browse_home_page-request_home_page- browse_home_page-request_home _ page-select_registration_page-requesting_registration_page-user_sign_up-user_sign_up-user_sign _ up-user_sign_up-new_page-prepare_and_initialize-create_new_student_ID-get_registration_info-get _student_registration-return_registration_data_and_student_ID*

Table 34: List of All Test Paths Name of Figure 51

3.5.10.2 Automated Testing of Example 10

...MESSAGE CASE TABLE for TEST PATHS...

Case_ID	Activity_Name
UMLGeneric	_student_
UMLGeneric	_browser_
UMLGeneric	_webServer_
UMLGeneric	_registrationPage_
UMLGeneric	_dB_
Relation	1:browse_home_pageIt=->>//syncMsg
Relation	2:abortIt=<.
Relation	3:browse_home_pageIt=->>//syncMsg
Relation	4:request_home_pageIt=->>//syncMsg
Relation	5:abortIt=<.
Relation	6:browse_home_pageIt=->>//syncMsg
Relation	7:request_home_pageIt=->>//syncMsg
Relation	8:responseIt=<.
Relation	9:select_registration_pageIt=->>//syncMsg
Relation	10:requesting_registration_pageIt=->>//syncMsg
Relation	11:ask_user_to_sign_upIt=<.
Relation	12:user_sign_upIt=->>//syncMsg
UMLGeneric	_aStudent_
Relation	13:abortIt=<.
Relation	14:user_sign_upIt=->>//syncMsg
Relation	15:abortIt=<.
Relation	16:user_sign_upIt=->>//syncMsg
Relation	17:abortIt=<.
Relation	18:user_sign_upIt=->>//syncMsg
Relation	19:new_pageIt=->>//syncMsg
Relation	20:prepare_and_initializeIt=->>//syncMsg
Relation	21:create_new_student_IDIt=->>//syncMsg
Relation	22:get_registration_infoIt=->>//syncMsg
Relation	23:get_student_registrationIt=->>//syncMsg
Relation	24:return_dataIt=<.
Relation	25:return_registration_dataIt=<.
Relation	26:returnIt=<.
Relation	27:return_registration_data_and_student_IDIt=<.

Figure 52: JTable of XML File in Appendix J

- **All Results for Test Paths**

```
TEST PATHS
All Possible Test Paths...

1. [1 - 2]
2. [1 - 2 - 3 - 4 - 5]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19 - 20 - 21 - 22 - 23 - 24]
9. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19 - 20 - 21 - 22 - 23 - 24 - 25]
10. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
- 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26]
```

11. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27]

1. [1:browse_home_page - 2:abort]
2. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort]
3. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response]
4. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up]
5. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort]
6. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort]
7. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort]
8. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page - 20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info - 23:get_student_registration - 24:return_data]
9. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page - 20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info - 23:get_student_registration - 24:return_data - 25:return_registration_data]
10. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page - 20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info - 23:get_student_registration - 24:return_data - 25:return_registration_data - 26:return]
11. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page - 20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info - 23:get_student_registration - 24:return_data - 25:return_registration_data - 26:return - 27:return_registration_data_and_student_ID]

BEST PATH

The Best Path: 11. Path [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page - 8:response - 9:select_registration_page - 10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page - 20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info -

```
23:get_student_registration - 24:return_data - 25:return_registration_data -
26:return - 27:return_registration_data_and_student_ID]

ALL MESSAGES

All messages of sequence diagram: [1:browse_home_page - 2:abort -
3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page -
7:request_home_page - 8:response - 9:select_registration_page -
10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up -
13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort -
18:user_sign_up - 19:new_page - 20:prepare_and_initialize -
21:create_new_student_ID - 22:get_registration_info - 23:get_student_registration -
24:return_data - 25:return_registration_data - 26:return -
27:return_registration_data_and_student_ID]

Total message number of sequence diagram: 27
```

- **Test Cases Table**

Test Case ID	Test Case Sequence	Test Case Activity Name ID	Test Case Activity Name (INPUT Messages)	Test Case Expected OUTPUT	Test Case Actual OUTPUT
1	1-2	**1**	browse_home_page	Redirect	abort
		2	abort		
2	1-3-4-5	**3**	browse_home_page	Redirect	-
		4	request_home_page	Response	abort
		5	abort		
3	1-3-4-6-7-8	**6**	browse_home_page	Redirect	-
		7	request_home_page	Response	Response
		8	response		
4	1-3-4-6-7-9-10-11	**9**	select_registration_page	Return Page	-
		10	requesting_registration_page	Response	ask_user_to_sign_up
		11	ask_user_to_sign_up		
5	1-3-4-6-7-9-10-12-13	**12**	user_sign_up	Return sign up	abort
		13	abort		
6	1-3-4-6-7-9-10-12-14-15	**14**	user_sign_up	Return sign up	abort
		15	abort		
7	1-3-4-6-7-9-10-12-14-16-17	**16**	user_sign_up	Return sign up	abort
		17	abort		
8	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-24	**18**	user_sign_up	Return sign up	return_registration_data_and_student_ID

		19	new_page	Return Page	-
		20	prepare_and_initialize	Return	Return
		21	create_new_student_ID	Record	-
		22	get_registration_info	Load System	Return_Registration_Data
		23	get_student_registration	Load System	Return_Data
		24	return_data		
9	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-25	**25**	return_registration_data		
10	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-26	**26**	return		
11	1-3-4-6-7-9-10-12-14-16-18-19-20-21-22-23-27	**27**	return_registration_data_and_student_ID		

Table 35: Test Cases Table of Example 10

3.6 Comparison Table of Manual & Automated Testing

Name_of Examples	MANUAL	AUTOMATED	RESULTS
1	**6** Paths	**6** Paths	Same
2	**2** Paths	**2** Paths	Same
3	**5** Paths	**5** Paths	Same
4	**4** Paths	**4** Paths	Same
5	**9** Paths	**9** Paths	Same
6	**1** Path	(ERROR)	Manual Better
7	**1** Path	(ERROR)	Manual Better
8	**8** Paths	**8** Paths	Same
9	**1** Path	(ERROR)	Manual Better
10	**11** Paths	**11** Paths	Same

Table 36: Comparison of Results of Manual & Automatic Testing

In this chapter, it is achieved test paths with sequence diagram modelling using both manual and automated testing in the MBT. UML sequence diagram is used as a

model, and it is studied on the same UML sequence diagram model examples that contains different areas. Same operation is used for both manual and automated testing. The proposed approach runs similarly for manual and automated testing.

The results of all examples are compared with each other for both manual and automated testing. Table 36 shows all testing results. Most of the time, test path results are the same both manual and automated testing. According to seven cases, test path results are the same. However, manual testing is better than the automated testing for three cases. In table 36, results of three examples are different from each other (examples 6. 7. 9.), because automated testing results give errors. Although, the system operation is the same for both manual and automated testing, automated testing code methods sometimes can run different. The junit testing code gives error because there is no any return message. Therefore, results give error according to code rules. These methods are in the *testPosibbleTestPaths(), testAllPaths()* and *testBestPath()* functions. *TestPosibbleTestPaths()* shows errors in sequence of test paths and number of test paths. *TestAllPaths()* do not show all possible test path relations and their names. Also, *testBestPath()* can not find the best path in result screen.

Therefore, this approach is not enough because of errors so, the approach should be improved. This thesis is evaluated in terms of manual and automated approaches. Changing of manual testing is more difficult than changing automated testing. Also, manual tests are not reliable compared to automated testing. Errors can increase in case of any changing in manual testing. On the other hand, automated testing are more reliable and more effective than manual testing. Automated tests can change easily. Moreover, errors can be found easily in automated testing compared to manual testing. Therefore, in this thesis automated testing method should be improved.

CHAPTER IV

COMPARISON OF MANUAL MODEL BASED TESTING TECHNIQUE WITH IMPROVED AUTOMATED MODEL BASED TESTING

4.1 Comparison of Manual & Automated Testing

In this thesis, both manual and automated testing as software testing in MBT are compared to each other. UML sequence diagram is preferred as a model for MBT. In this study, two testing methods are executed using the same operation on the same UML sequence diagram examples. The same test results are achieved from these methods, and it is decided to improve automated testing method. In this study, manual testing method was not changed because changing of manual testing is more difficult than changing of automated testing. Manual tests are not reliable, stimulating and reuse. Also, manual tests is time consuming, and this tests can give error in case of any changing. Automated testing has more advantage compared to manual testing. Automated tests are more reliable, useful, fast, effective and easy than manual tests. Automated testing method can give less error in case of any changing. Errors can be correctted easier than manual tests. Tools are very useful in terms of automated testing but, automated codes that are generated from automated testing are more useful. Every tool can not give desired results for testing, but automated codes are more flexible compared to existing tools. The proposed automated testing method is preferred to change, and improved junit testing code is established using different testing methods. In conclusion, improved automated

testing method has more successful test results than the manual testing method. **Appendix M** includes the proposed improved junit code.

4.2 Improved Automated Model Based Testing

In chapter 4, the proposed automated testing method is improved. It is studied on again the same UML sequence diagram examples in chapter 3. Some different methods are used in this improved automated testing method. First of all, method that is used in chapter 3 is changed. Contents of *testPossibleTestPaths()* method is changed measuring dependency of sending and receiving message to find better sequence of test paths. Secondly, *testReadMessages()* method is improved to read all message relations of the sequence diagram. Thirdly, some sequence diagrams have interaction frames, and because of this properties of the diagrams a new method is added to improved junit code. This method name is *testNameof Diagram(),* and this method shows names of all interaction frames. Finally, *testObjects()* method is added into the improved code to read names of all object, and to count number of objects of sequence diagram. Different implementations are done to find better test paths. Improved automated junit code is proven with the following example.

The following examples show results of the improved junit testing. In chapter 3, examples of ten sequence diagrams are used as a model for MBT. In chapter 4 has more better results than chapter 3. Contents of test cases tables do not change in chapter 3, but just, sequence of test cases changes because of changing of test paths in chapter 4.

4.2.1 Example1

```
OBJECTS

All objects of sequence diagram: [_User_, _ATM_, _Server_]

Total objects number of sequence diagram: 3

TEST PATHS

All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12]
```

```
1. [1:InserdCard - 2:Validate(Card) - 3:Return]
2. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin]
3. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin
   - 7:Return]
4. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin
   - 7:Return - 8:EnterAmount]
5. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin
   - 7:Return - 8:EnterAmount - 9:Amount - 10:Check(Balance) - 11:Return]
6. [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin - 5:Pin - 6:VerifyPin
   - 7:Return - 8:EnterAmount - 9:Amount - 10:Check(Balance) - 11:Return –
   12:TakeCash]

BEST PATH

The Best Path: 6.  Path [1:InserdCard - 2:Validate(Card) - 3:Return - 4:EnterPin -
5:Pin - 6:VerifyPin - 7:Return - 8:EnterAmount - 9:Amount - 10:Check(Balance) -
11:Return - 12:TakeCash]

ALL MESSAGES

All messages of sequence diagram: [1:InserdCard - 2:Validate(Card) - 3:Return -
4:EnterPin - 5:Pin - 6:VerifyPin - 7:Return - 8:EnterAmount - 9:Amount -
10:Check(Balance) - 11:Return - 12:TakeCash]

Total message number of sequence diagram: 12

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:InserdCard->syncMsg,
2:Validate(Card)->syncMsg, 3:Return, 4:EnterPin, 5:Pin->syncMsg, 6:VerifyPin-
>syncMsg, 7:Return, 8:EnterAmount, 9:Amount->syncMsg, 10:Check(Balance)->syncMsg,
11:Return, 12:TakeCash]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: [ATM]
```

4.2.2 Example2

```
OBJECTS

All objects of sequence diagram: [_:SessionManager_, _:DisplayManager_,
_:KeyReader_, _:Bank_, _:CardReader_]

Total objects number of sequence diagram: 5

TEST PATHS

All Possible Test Paths...

1. [1 - 2 - 3 - 4]
2. [1 - 2 - 3 - 4 - 5 - 6]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9]

1. [1:cardinserted - 2:selectWithDraw - 3:requestAmount - 4:amt_value]
2. [1:cardinserted - 2:selectWithDraw - 3:requestAmount - 4:amt_value –
   5:verifyAmount() - 6:(InvalidAmount)eject]

3. [1:cardinserted - 2:selectWithDraw - 3:requestAmount - 4:amt_value –
   5:verifyAmount() - 6:(InvalidAmount)eject - 7:(ValidAmount)checkBalance –
   8:(InsufficientAmount)eject]

4. [1:cardinserted - 2:selectWithDraw - 3:requestAmount - 4:amt_value -
```

```
   5:verifyAmount() - 6:(InvalidAmount)eject - 7:(ValidAmount)checkBalance –
   8:(InsufficientAmount)eject - 9:(SufficientBalance)debit_Acc]
BEST PATH

The Best Path: 4.  Path [1:cardinserted - 2:selectWithDraw - 3:requestAmount -
4:amt_value - 5:verifyAmount() - 6:(InvalidAmount)eject -
7:(ValidAmount)checkBalance - 8:(InsufficientAmount)eject -
9:(SufficientBalance)debit_Acc]

ALL MESSAGES

All messages of sequence diagram: [1:cardinserted -> 2:selectWithDraw ->
3:requestAmount -> 4:amt_value -> 5:verifyAmount() -> 6:(InvalidAmount)eject ->
7:(ValidAmount)checkBalance -> 8:(InsufficientAmount)eject ->
9:(SufficientBalance)debit_Acc]

Total message number of sequence diagram: 9

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:cardinserted->syncMsg,
2:selectWithDraw->syncMsg, 3:requestAmount->syncMsg, 4:amt_value, 5:verifyAmount()-
>syncMsg, 6:(InvalidAmount)eject->syncMsg, 7:(ValidAmount)checkBalance->syncMsg,
8:(InsufficientAmount)eject->syncMsg, 9:(SufficientBalance)debit_Acc->syncMsg]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []
```

4.2.3 Example3

```
OBJECTS

All objects of sequence diagram: [_:SpiderMenuBar_, _:Spider_, _:PKCard_, _:Deal_]

Total objects number of sequence diagram: 4

TEST PATHS

All Possible Test Paths...

1. [1]
2. [1 - 2]
3. [1 - 2 - 3]
4. [1 - 2 - 3 - 4]
5. [1 - 2 - 3 - 4 - 5]
6. [1 - 2 - 3 - 4 - 5 - 6]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]


1. [1:setGame(grade:int):void - Return]
2. [1:setGame(grade:int):void - Return - 2:newGame():void]
3. [1:setGame(grade:int):void - Return - 2:newGame():void –
   3:ShowEnableOperation():void]
4. [1:setGame(grade:int):void - Return - 2:newGame():void –
   3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return]
5. [1:setGame(grade:int):void - Return - 2:newGame():void –
   3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return –
   5:Deal():void - Return]
6. [1:setGame(grade:int):void - Return - 2:newGame():void –
   3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return –
   5:Deal():void - Return - 6:ShowEnableOperation():void]
7. [1:setGame(grade:int):void - Return - 2:newGame():void –
   3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return –
```

```
   5:Deal():void - Return - 6:ShowEnableOperation():void –
   7:moveto(Point:point):void - Return]
8. [1:setGame(grade:int):void - Return - 2:newGame():void –
   3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return –
   5:Deal():void - Return - 6:ShowEnableOperation():void –
   7:moveto(Point:point):void - Return - 8:haveFinish(column:int):void]

BEST PATH

The Best Path: 8.  Path [1:setGame(grade:int):void - Return - 2:newGame():void -
3:ShowEnableOperation():void - 4:moveto(Point:point):void - Return - 5:Deal():void
- Return - 6:ShowEnableOperation():void - 7:moveto(Point:point):void - Return -
8:haveFinish(column:int):void]

ALL MESSAGES

All messages of sequence diagram: [1:setGame(grade:int):void -> Return ->
2:newGame():void -> 3:ShowEnableOperation():void -> 4:moveto(Point:point):void ->
Return -> 5:Deal():void -> Return -> 6:ShowEnableOperation():void ->
7:moveto(Point:point):void -> Return -> 8:haveFinish(column:int):void]

Total message number of sequence diagram: 12

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:setGame(grade:int):void->syncMsg,
Return, 2:newGame():void->syncMsg, 3:ShowEnableOperation():void->syncMsg,
4:moveto(Point:point):void->syncMsg, Return, 5:Deal():void->syncMsg, Return,
6:ShowEnableOperation():void->syncMsg, 7:moveto(Point:point):void->syncMsg, Return,
8:haveFinish(column:int):void->syncMsg]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []
```

4.2.4 Example4

```
OBJECTS

All objects of sequence diagram: [_Patient_, _User_Interface_, _Doctor_,
_DiagnosisSystem_]

Total objects number of sequence diagram: 4

TEST PATHS

All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4 - 5 - 6 - 7]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9]


1. [1:Login - 2:Verify - 3:Result]
2. [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails - 5:Request –
   6:ReferPatientHistory - 7:RetrieveDeatils]
3. [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails - 5:Request –
   6:ReferPatientHistory - 7:RetrieveDeatils –
   8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest]
4. [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails - 5:Request –
   6:ReferPatientHistory - 7:RetrieveDeatils –
   8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest –
   9:DisplayPrescription/SuggestedTests]

BEST PATH
```

```
The Best Path: 4.  Path [1:Login - 2:Verify - 3:Result - 4:EnterPatientDetails -
5:Request - 6:ReferPatientHistory - 7:RetrieveDeatils -
8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest -
9:DisplayPrescription/SuggestedTests]

ALL MESSAGES

All messages of sequence diagram: [1:Login - 2:Verify - 3:Result -
4:EnterPatientDetails - 5:Request - 6:ReferPatientHistory - 7:RetrieveDeatils -
8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest -
9:DisplayPrescription/SuggestedTests]

Total message number of sequence diagram: 9

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:Login->syncMsg, 2:Verify->syncMsg,
3:Result, 4:EnterPatientDetails->syncMsg, 5:Request->syncMsg,
6:ReferPatientHistory->syncMsg, 7:RetrieveDeatils,
8:DiagnoseandPrescribeMedicine/SuggestAdditionalTest,
9:DisplayPrescription/SuggestedTests]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []
```

4.2.5 Example5

```
OBJECTS

All objects of sequence diagram: [_User_, _WebBrowses_, _Application_,
_Facebook__AuthorizationServer_, _Facebook__ContentServer_]

Total objects number of sequence diagram: 5

TEST PATHS

All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4 - 5]
3. [1 - 2 - 3 - 4 - 5 - 6]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16]
9. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 –
   18]
10. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 –
    18 - 19]


1. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect]
2. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c)]
3. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form]
4. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form -
7:user_permission(int_a) - 8:process_permission - 9:http_redirect]
5. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form -
7:user_permission(int_a) - 8:process_permission - 9:http_redirect -
```

10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) -
12:access_token]
6. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form -
7:user_permission(int_a) - 8:process_permission - 9:http_redirect -
10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) -
12:access_token - 13:access_FB_user_protected_resource -
14:user_protected_resource]
7. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form -
7:user_permission(int_a) - 8:process_permission - 9:http_redirect -
10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) -
12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource
- 15:user_protected_resource]
8. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form -
7:user_permission(int_a) - 8:process_permission - 9:http_redirect -
10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) -
12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource
- 15:user_protected_resource - 16:user_protected_resource]
9. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form -
7:user_permission(int_a) - 8:process_permission - 9:http_redirect -
10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) -
12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource
- 15:user_protected_resource - 16:user_protected_resource -
17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available]
10. [1:getFB_Resource(int_a,int_b) - 2:request_FB_access - 3:http_redirect -
4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) - 6:permission_form -
7:user_permission(int_a) - 8:process_permission - 9:http_redirect -
10:FB_authorization_code - 11:FB_authorization_code(int_a,int_b,int_c) -
12:access_token - 13:access_FB_user_protected_resource - 14:user_protected_resource
- 15:user_protected_resource - 16:user_protected_resource -
17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available -
19:18:FB_resource_not_available]

BEST PATH

The Best Path: 10. Path [1:getFB_Resource(int_a,int_b) - 2:request_FB_access -
3:http_redirect - 4:authorize(int_a) - 5:permission_form(int_a,int_b,int_c) -
6:permission_form - 7:user_permission(int_a) - 8:process_permission -
9:http_redirect - 10:FB_authorization_code -
11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token -
13:access_FB_user_protected_resource - 14:user_protected_resource -
15:user_protected_resource - 16:user_protected_resource -
17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available -
19:18:FB_resource_not_available]

ALL MESSAGES

All messages of sequence diagram: [1:getFB_Resource(int_a,int_b) -
2:request_FB_access - 3:http_redirect - 4:authorize(int_a) -
5:permission_form(int_a,int_b,int_c) - 6:permission_form - 7:user_permission(int_a)
- 8:process_permission - 9:http_redirect - 10:FB_authorization_code -
11:FB_authorization_code(int_a,int_b,int_c) - 12:access_token -
13:access_FB_user_protected_resource - 14:user_protected_resource -
15:user_protected_resource - 16:user_protected_resource -
17:no_authorization(int_a,int_b,int_c,intd) - 18:FB_resource_not_available -
19:18:FB_resource_not_available]

Total message number of sequence diagram: 19

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:getFB_Resource(int_a,int_b)-
>syncMsg, 2:request_FB_access->syncMsg, 3:http_redirect->returnMsg,

```
4:authorize(int_a)->syncMsg, 5:permission_form(int_a,int_b,int_c)->returnMsg,
6:permission_form->returnMsg, 7:user_permission(int_a)->syncMsg,
8:process_permission->syncMsg, 9:http_redirect->returnMsg,
10:FB_authorization_code->syncMsg, 11:FB_authorization_code(int_a,int_b,int_c)-
>syncMsg, 12:access_token->returnMsg, 13:access_FB_user_protected_resource-
>syncMsg, 14:user_protected_resource->syncMsg, 15:user_protected_resource-
>returnMsg, 16:user_protected_resource->returnMsg,
17:no_authorization(int_a,int_b,int_c,intd)->syncMsg, 18:FB_resource_not_available-
>returnMsg, 19:18:FB_resource_not_available->returnMsg]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []
```

4.2.6 Example6

```
OBJECTS

All objects of sequence diagram: [_OperatorPanel_, _ATM_, _CashDispenser_, _Bank_]

Total objects number of sequence diagram: 4

TEST PATHS

All Possible Test Paths...

1. [1 - 2]
2. [1 - 2 - 3]
3. [1 - 2 - 3 - 4 - 5 - 6]

1. [1:Switch_on() - 2:PerformStartUp()]
2. [1:Switch_on() - 2:PerformStartUp() - 3:GetInitialCash()]
3. [1:Switch_on() - 2:PerformStartUp() - 3:GetInitialCash() - 4:LoadingCash() –
   5:SetCash() - 6:OpenConnection()]
BEST PATH

The Best Path: 3.  Path [1:Switch_on() - 2:PerformStartUp() - 3:GetInitialCash() -
4:LoadingCash() - 5:SetCash() - 6:OpenConnection()]

ALL MESSAGES

All messages of sequence diagram: [1:Switch_on() -> 2:PerformStartUp() ->
3:GetInitialCash() -> 4:LoadingCash() -> 5:SetCash() -> 6:OpenConnection()]

Total message number of sequence diagram: 6

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:Switch_on()->syncMsg,
2:PerformStartUp()->syncMsg, 3:GetInitialCash()->syncMsg, 4:LoadingCash()->syncMsg,
5:SetCash()->syncMsg, 6:OpenConnection()->syncMsg]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []
```

4.2.7 Example7

```
OBJECTS

All objects of sequence diagram: [_Customer_, _:BookStore_, _:Inventory_,
_:Payment_, _:Delivery_, _:Message_]

Total objects number of sequence diagram: 6
```

```
TEST PATHS

All Possible Test Paths...

1. [1 - 2 - 3 - 4 - 5 - 6 - 7]

1. [1:Search(bookname,author) - 2:checkStock(ISBN) - 3:buy(id,ISBN,gty) –
   4:makePayment(id,totalprice) - 5:ship(ISBN,decrease_NUM) - 6:deliver() –
   7:sendMessage()]
BEST PATH

The Best Path: 1.  Path [1:Search(bookname,author) - 2:checkStock(ISBN) -
3:buy(id,ISBN,gty) - 4:makePayment(id,totalprice) - 5:ship(ISBN,decrease_NUM) -
6:deliver() - 7:sendMessage()]

ALL MESSAGES

All messages of sequence diagram: [1:Search(bookname,author) -> 2:checkStock(ISBN)
-> 3:buy(id,ISBN,gty) -> 4:makePayment(id,totalprice) -> 5:ship(ISBN,decrease_NUM)
-> 6:deliver() -> 7:sendMessage()]

Total message number of sequence diagram: 7

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:Search(bookname,author)->syncMsg,
2:checkStock(ISBN)->syncMsg, 3:buy(id,ISBN,gty)->syncMsg,
4:makePayment(id,totalprice)->syncMsg, 5:ship(ISBN,decrease_NUM)->syncMsg,
6:deliver()->syncMsg, 7:sendMessage()->syncMsg]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []
```

4.2.8 Example8

```
OBJECTS

All objects of sequence diagram: [_CardReader_, _DisplayManager_, _SessionManager_,
_PasswordReader_, _UserAcount_, _LibraryDatabase_]

Total objects number of sequence diagram: 6

TEST PATHS

All Possible Test Paths...

1. [1 - 2 - 3]
2. [1 - 2 - 3 - 4]
3. [1 - 2 - 3 - 4]
4. [1 - 2 - 3 - 4 - 5]
5. [1 - 2 - 3 - 4 - 5 - 6]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15]
9. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16]
10. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17]
11. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 -
    18 - 19 - 20]
12. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 –
    18 - 19 - 20]
13. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 –
    18 - 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26]

1. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return]
2. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()]
```

```
3. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return]
4. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return]
5. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6()]
6. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return]
7. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return]
8. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return]
9. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return - 16:BookNotAvailableM16()]
10. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return - 16:BookNotAvailableM16() -
17:BookAvailableM17()]
11. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return - 16:BookNotAvailableM16() -
17:BookAvailableM17() - 18:UpdateLibraryDatabaseM18() - 19:UpdateUserAcountM19() -
20:AcountNotCreatedM20()]
12. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return - 16:BookNotAvailableM16() -
17:BookAvailableM17() - 18:UpdateLibraryDatabaseM18() - 19:UpdateUserAcountM19() -
20:AcountNotCreatedM20() - Return]
13. [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() - Return - 4:InvalidPWM4()
- Return - 5:AccountValueM5() - Return - 6:AccountCreatedM6() - 7:CloseAccountM7()
- Return - 8:DestroyUserAccountM8() - 9:SuspendICM9() - 10:ContinueM10() - Return -
11:ReturnBookM11() - 12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() -
14:IssueBookM14() - 15:BOOKIdM15() - Return - 16:BookNotAvailableM16() -
17:BookAvailableM17() - 18:UpdateLibraryDatabaseM18() - 19:UpdateUserAcountM19() -
20:AcountNotCreatedM20() - Return - 21:GetNameM21() - 22:GetF/MM22() -
23:GetDobM23() - 24:GetAddressM24() - 25:GetPhnoM25() - 26:GetAffiliationM26()]
BEST PATH

The Best Path: 13.  Path [1:CardValueM1() - 2:InvalidCardM2() - 3:PWValueM3() -
Return - 4:InvalidPWM4() - Return - 5:AccountValueM5() - Return -
6:AccountCreatedM6() - 7:CloseAccountM7() - Return - 8:DestroyUserAccountM8() -
9:SuspendICM9() - 10:ContinueM10() - Return - 11:ReturnBookM11() -
12:UpdateLibraryDatabaseM12() - 13:UpdateUserAcountM13() - 14:IssueBookM14() -
15:BOOKIdM15() - Return - 16:BookNotAvailableM16() - 17:BookAvailableM17() -
18:UpdateLibraryDatabaseM18() - 19:UpdateUserAcountM19() - 20:AcountNotCreatedM20()
- Return - 21:GetNameM21() - 22:GetF/MM22() - 23:GetDobM23() - 24:GetAddressM24() -
25:GetPhnoM25() - 26:GetAffiliationM26()]

ALL MESSAGES

All messages of sequence diagram: [1:CardValueM1() -> 2:InvalidCardM2() ->
3:PWValueM3() -> Return -> 4:InvalidPWM4() -> Return -> 5:AccountValueM5() ->
```

```
Return -> 6:AccountCreatedM6() -> 7:CloseAccountM7() -> Return ->
8:DestroyUserAccountM8() -> 9:SuspendICM9() -> 10:ContinueM10() -> Return ->
11:ReturnBookM11() -> 12:UpdateLibraryDatabaseM12() -> 13:UpdateUserAcountM13() ->
14:IssueBookM14() -> 15:BOOKIdM15() -> Return -> 16:BookNotAvailableM16() ->
17:BookAvailableM17() -> 18:UpdateLibraryDatabaseM18() -> 19:UpdateUserAcountM19()
-> 20:AcountNotCreatedM20() -> Return -> 21:GetNameM21() -> 22:GetF/MM22() ->
23:GetDobM23() -> 24:GetAddressM24() -> 25:GetPhnoM25() -> 26:GetAffiliationM26()]

Total message number of sequence diagram: 33

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:CardValueM1()->syncMsg,
2:InvalidCardM2()->syncMsg, 3:PWValueM3()->syncMsg, Return->returnMsg,
4:InvalidPWM4()->syncMsg, Return->returnMsg, 5:AccountValueM5()->syncMsg, Return-
>returnMsg, 6:AccountCreatedM6()->syncMsg, 7:CloseAccountM7()->syncMsg, Return-
>returnMsg, 8:DestroyUserAccountM8()->syncMsg, 9:SuspendICM9()->syncMsg,
10:ContinueM10()->syncMsg, Return->returnMsg, 11:ReturnBookM11()->syncMsg,
12:UpdateLibraryDatabaseM12()->syncMsg, 13:UpdateUserAcountM13()->syncMsg,
14:IssueBookM14()->syncMsg, 15:BOOKIdM15()->syncMsg, Return->returnMsg,
16:BookNotAvailableM16()->syncMsg, 17:BookAvailableM17()->syncMsg,
18:UpdateLibraryDatabaseM18()->syncMsg, 19:UpdateUserAcountM19()->syncMsg,
20:AcountNotCreatedM20()->syncMsg, Return->returnMsg, 21:GetNameM21()->syncMsg,
22:GetF/MM22()->syncMsg, 23:GetDobM23()->syncMsg, 24:GetAddressM24()->syncMsg,
25:GetPhnoM25()->syncMsg, 26:GetAffiliationM26()->syncMsg]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: [Alt, Alt, Alt, Alt, Par, Alt, Par, Alt, Par,
Par]
```

4.2.9 Example9

```
OBJECTS

All objects of sequence diagram: [C:Customer, A:ATM, Accnt:Acount, CKA:Checking
Acount]

Total objects number of sequence diagram: 4

TEST PATHS

All Possible Test Paths...

1. [1 - 2]
2. [1 - 2 - 3 - 4 - 5]
3. [1 - 2 - 3 - 4 - 5 - 6]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15]
9. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17]
10. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 –
    18 - 19]


1. [1:InsertAtmCard() - 2:RequestPin()]
2. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK()]
3. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK() - 6:RequestOption()]
4. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK() - 6:RequestOption() - 7:OptionEntered() - 8:RequestAmount()]
5. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK() - 6:RequestOption() - 7:OptionEntered() - 8:RequestAmount() –
```

```
   9:AmountEntered() - 10:ProcessTransaction() - 11:WithdrawCheckingAcount() –
   12:WithdrawSuccessful()]
6. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK() - 6:RequestOption() - 7:OptionEntered() - 8:RequestAmount() –
   9:AmountEntered() - 10:ProcessTransaction() - 11:WithdrawCheckingAcount() –
   12:WithdrawSuccessful() - 13:TransactionSuccessful()]
7. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK() - 6:RequestOption() - 7:OptionEntered() - 8:RequestAmount() –
   9:AmountEntered() - 10:ProcessTransaction() - 11:WithdrawCheckingAcount() –
   12:WithdrawSuccessful() - 13:TransactionSuccessful() - 14:DispenseCash()]
8. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK() - 6:RequestOption() - 7:OptionEntered() - 8:RequestAmount() –
   9:AmountEntered() - 10:ProcessTransaction() - 11:WithdrawCheckingAcount() –
   12:WithdrawSuccessful() - 13:TransactionSuccessful() - 14:DispenseCash() –
   15:RequestTakeCash()]
9. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
   5:PinOK() - 6:RequestOption() - 7:OptionEntered() - 8:RequestAmount() –
   9:AmountEntered() - 10:ProcessTransaction() - 11:WithdrawCheckingAcount() –
   12:WithdrawSuccessful() - 13:TransactionSuccessful() - 14:DispenseCash() –
   15:RequestTakeCash() - 16:TakeCash() - 17:RequestContinuation()]
10. [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() - 4:VerifyPin() –
    5:PinOK() - 6:RequestOption() - 7:OptionEntered() - 8:RequestAmount() –
    9:AmountEntered() - 10:ProcessTransaction() - 11:WithdrawCheckingAcount() –
    12:WithdrawSuccessful() - 13:TransactionSuccessful() - 14:DispenseCash() –
    15:RequestTakeCash() - 16:TakeCash() - 17:RequestContinuation() –
    18:Terminate() - 19:PrintReceipt()]

BEST PATH

The Best Path: 10.  Path [1:InsertAtmCard() - 2:RequestPin() - 3:PinEntered() -
4:VerifyPin() - 5:PinOK() - 6:RequestOption() - 7:OptionEntered() -
8:RequestAmount() - 9:AmountEntered() - 10:ProcessTransaction() -
11:WithdrawCheckingAcount() - 12:WithdrawSuccessful() - 13:TransactionSuccessful()
- 14:DispenseCash() - 15:RequestTakeCash() - 16:TakeCash() -
17:RequestContinuation() - 18:Terminate() - 19:PrintReceipt()]

ALL MESSAGES

All messages of sequence diagram: [1:InsertAtmCard() -> 2:RequestPin() ->
3:PinEntered() -> 4:VerifyPin() -> 5:PinOK() -> 6:RequestOption() ->
7:OptionEntered() -> 8:RequestAmount() -> 9:AmountEntered() ->
10:ProcessTransaction() -> 11:WithdrawCheckingAcount() -> 12:WithdrawSuccessful() -
> 13:TransactionSuccessful() -> 14:DispenseCash() -> 15:RequestTakeCash() ->
16:TakeCash() -> 17:RequestContinuation() -> 18:Terminate() -> 19:PrintReceipt()]

Total message number of sequence diagram: 19

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:InsertAtmCard()->syncMsg,
2:RequestPin()->syncMsg, 3:PinEntered()->syncMsg, 4:VerifyPin()->syncMsg,
5:PinOK()->syncMsg, 6:RequestOption()->syncMsg, 7:OptionEntered()->syncMsg,
8:RequestAmount()->syncMsg, 9:AmountEntered()->syncMsg, 10:ProcessTransaction()-
>syncMsg, 11:WithdrawCheckingAcount()->syncMsg, 12:WithdrawSuccessful()->syncMsg,
13:TransactionSuccessful()->syncMsg, 14:DispenseCash()->syncMsg,
15:RequestTakeCash()->syncMsg, 16:TakeCash()->syncMsg, 17:RequestContinuation()-
>syncMsg, 18:Terminate()->syncMsg, 19:PrintReceipt()->syncMsg]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []
```

4.2.10 Example10

```
OBJECTS
```

```
All objects of sequence diagram: [_student_, _browser_, _webServer_,
_registrationPage_, _dB_, _aStudent_]

Total objects number of sequence diagram: 6

TEST PATHS

All Possible Test Paths...

1. [1 - 2]
2. [1 - 2 - 3 - 4 - 5]
3. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8]
4. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11]
5. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13]
6. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15]
7. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17]
8. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
   - 19 - 20 - 21 - 22 - 23 - 24]
9. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18
   - 19 - 20 - 21 - 22 - 23 - 24 - 25]
10. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 -
    18 - 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26]
11. [1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 -
    18 - 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27]


1. [1:browse_home_page - 2:abort]
2. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort]
3. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort - 6:browse_home_page - 7:request_home_page - 8:response]
4. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort - 6:browse_home_page - 7:request_home_page - 8:response -
   9:select_registration_page - 10:requesting_registration_page -
   11:ask_user_to_sign_up]
5. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort - 6:browse_home_page - 7:request_home_page - 8:response -
   9:select_registration_page - 10:requesting_registration_page -
   11:ask_user_to_sign_up - 12:user_sign_up - 13:abort]
6. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort - 6:browse_home_page - 7:request_home_page - 8:response -
   9:select_registration_page - 10:requesting_registration_page -
   11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up -
   15:abort]
7. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort - 6:browse_home_page - 7:request_home_page - 8:response -
   9:select_registration_page - 10:requesting_registration_page -
  11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort
  - 16:user_sign_up - 17:abort]
8. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort - 6:browse_home_page - 7:request_home_page - 8:response -
   9:select_registration_page - 10:requesting_registration_page -
   11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up -
   15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page -
   20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info
   - 23:get_student_registration - 24:return_data]
9. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
   5:abort - 6:browse_home_page - 7:request_home_page - 8:response -
   9:select_registration_page - 10:requesting_registration_page -
   11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up -
   15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page -
   20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info
   - 23:get_student_registration - 24:return_data - 25:return_registration_data]
10. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
    5:abort - 6:browse_home_page - 7:request_home_page - 8:response -
    9:select_registration_page - 10:requesting_registration_page -
```

11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up –
15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page –
20:prepare_and_initialize - 21:create_new_student_ID –
22:get_registration_info - 23:get_student_registration - 24:return_data –
25:return_registration_data - 26:return]

11. [1:browse_home_page - 2:abort - 3:browse_home_page - 4:request_home_page -
5:abort - 6:browse_home_page - 7:request_home_page - 8:response –
9:select_registration_page - 10:requesting_registration_page –
11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up –
15:abort - 16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page –
20:prepare_and_initialize - 21:create_new_student_ID –
22:get_registration_info - 23:get_student_registration - 24:return_data –
25:return_registration_data - 26:return –
27:return_registration_data_and_student_ID]

BEST PATH

The Best Path: 11. Path [1:browse_home_page - 2:abort - 3:browse_home_page -
4:request_home_page - 5:abort - 6:browse_home_page - 7:request_home_page -
8:response - 9:select_registration_page - 10:requesting_registration_page -
11:ask_user_to_sign_up - 12:user_sign_up - 13:abort - 14:user_sign_up - 15:abort -
16:user_sign_up - 17:abort - 18:user_sign_up - 19:new_page -
20:prepare_and_initialize - 21:create_new_student_ID - 22:get_registration_info -
23:get_student_registration - 24:return_data - 25:return_registration_data -
26:return - 27:return_registration_data_and_student_ID]

ALL MESSAGES

All messages of sequence diagram: [1:browse_home_page - 2:abort -
3:browse_home_page - 4:request_home_page - 5:abort - 6:browse_home_page -
7:request_home_page - 8:response - 9:select_registration_page -
10:requesting_registration_page - 11:ask_user_to_sign_up - 12:user_sign_up -
13:abort - 14:user_sign_up - 15:abort - 16:user_sign_up - 17:abort -
18:user_sign_up - 19:new_page - 20:prepare_and_initialize -
21:create_new_student_ID - 22:get_registration_info - 23:get_student_registration -
24:return_data - 25:return_registration_data - 26:return -
27:return_registration_data_and_student_ID]

Total message number of sequence diagram: 27

MESSAGES RELATIONS

All messages relations of sequence diagram: [1:browse_home_page->syncMsg, 2:abort,
3:browse_home_page->syncMsg, 4:request_home_page->syncMsg, 5:abort,
6:browse_home_page->syncMsg, 7:request_home_page->syncMsg, 8:response,
9:select_registration_page->syncMsg, 10:requesting_registration_page->syncMsg,
11:ask_user_to_sign_up, 12:user_sign_up->syncMsg, 13:abort, 14:user_sign_up-
>syncMsg, 15:abort, 16:user_sign_up->syncMsg, 17:abort, 18:user_sign_up->syncMsg,
19:new_page->syncMsg, 20:prepare_and_initialize->syncMsg, 21:create_new_student_ID-
>syncMsg, 22:get_registration_info->syncMsg, 23:get_student_registration->syncMsg,
24:return_data, 25:return_registration_data, 26:return,
27:return_registration_data_and_student_ID]

INTERACTION FRAMES OF DIAGRAM

Interaction Frames names of diagram: []

4.3 Comparison of Proposed Manual vs Automated & Improved Automated Testing

Number of Examples	Manual Testing	Automated Testing (JUNIT)	Improved Automated Testing (JUNIT)	COMPARISON of RESULTS
1	**6** Paths	**6** Paths	**6** Paths	Same
2	**2** Paths	**2** Paths	**4** Paths	**Improved Junit Better**
3	**5** Paths	**5** Paths	**8** Paths	**Improved Junit Better**
4	**4** Paths	**4** Paths	**4** Paths	Same
5	**9** Paths	**9** Paths	**10** Paths	**Improved Junit Better**
6	**1** Path	(ERROR)	**3** Paths	**Improved Junit Better**
7	**1** Path	(ERROR)	**1** Path	Manual & Improved Junit Better
8	**8** Paths	**8** Paths	**13** Paths	**Improved Junit Better**
9	**1** Path	(ERROR)	**10** Paths	**Improved Junit Better**
10	**11** Paths	**11** Paths	**11** Paths	Same

Table 37: Comparison Table of All Proposed Testing

In Table 37, all results of test paths of the proposed approaches are shown using the same UML sequence diagram examples for model based software testing. According to this table, results of manual and improved automated testing methods are better than result of automated testing method for one case. The proposed automated junit method gives errors for test paths, so results of manual and improved junit method results are available. For three cases, test results are the same in terms of all methods. In six cases, the improved junit automatic testing results are more better than the other methods.

4.4 Comparison of Proposed Manual & Improved Automated Testing with Existing Literature

Number of Examples	Manual Testing	Improved Automated Testing (JUNIT)	Research Papers with Same Examples	Technique Used by Research Papers	Results of Research Papers	COMPARISON of RESULTS
1	**6** Paths	**6** Paths	A.V.K. Shanthi &G. Mohan Kumar [3]	A Novel Approach (Using GA)	**4** Paths	Manual & Improved Junit Better
2	**2** Paths	**4** Paths	Vinaya Sawant & Ketan Shah [1]	A Novel Tecnique to Develop New Tool(Using Java & Tree)	**2** Paths	**Improved Junit Better**
3	**5** Paths	**8** Paths	V.MarySumalatha &G.S.V.P. Raju [10]	GA	**4** Paths	**Improved Junit Better**
4	**4** Paths	**4** Paths	S. Shanmuga Priya& P. D. Sheba Kezia Malarchelvi [9]	SDT & SDG	**4** Paths	Same
5	**9** Paths	**10** Paths	Rishi Seth & Sanyam Anand [2]	Develop New Tool (Using Visual Studio 2010, C#)	**4** Paths	**Improved Junit Better**
6	**1** Path	**3** Paths	M.Dhineshkumar& PG Scholar& Professor.Mr. Galeebathullah[5]	A Novel Approach, SDG	**6** Paths	Research Paper Better
7	**1** Path	**1** Path	Ching-Seh Wu & Chi-Hsin Huang [6]	EFSM	**2** Paths	Research Paper Better
8	**8** Paths	**13** Paths	Monalisha Khandai & Arup Abhinna Acharya & Durga Prasad Mohapatra [4]	SG, DFS & BFS	**7** Paths	**Improved Junit Better**
9	**1** Path	**10** Paths	V.MarySumalatha &Dr G.S.V.P. Raju [7]	SDG & BFS	**5** Paths	**Improved Junit Better**
10	**11** Paths	**11** Paths	Rohin Verma & Rajesh Bhatia [8]	An Innovative Approach (Using Petal File Reader & SQL *Loader)	**7** Paths	Manual & Improved Junit Better

Table 38: Comparison Table of All Results

In this chapter, results of all manual, improved and research papers methods are compared in terms of number of test paths. Also, authors and techniques of the research papers are specified. All results in the Table 38 are shown. Most of the time, the improved junit automated testing results are better than the other results (examples 2. 3. 5. 8. and 9.). In two cases, testing results are different from each other. In examples 1. and 10. , the manual and improved junit methods are better than the research papers method. In one case, testing results are the same with each other (example 4.) for all methods. In two cases, results of research papers method are better than the proposed method results (examples 6. and 7.). As a result, the improved junit testing method results are better than the other results according to Table 38.

CHAPTER V

CONCLUSION

5.1 Conclusion

Model Based Testing (MBT) is a black-box testing approach of the software testing. Testing is a very important phase for quality, reliability and usability of software. Use of modeling techniques in software testing is effective way.

In software testing, testing consists of three phases that are test case generation, test execution, and test evaluation. It is mentioned only test case generation phases of the software testing in this thesis. Test case generation phase is the basic process of testing. Also, this phase is the most important phases in model based testing. Test data and some steps are effective to obtain test cases in this phase.

Testing is separated into automated and manual tesing. In this thesis, it is used both automated and manual testing. Software testing is the basic activities in the software development life cycle. This phase depends on many states such as time, effort, money.

Manual testing has some good properties such as flexibility, easiness, low cost etc.. Tool is not used in manual testing, and desired operations are difficut to do manually. Therefore, manual testing has some bad properties such as reliability, stimulating and time consuming etc.. Errors can be shown very much in manual tests, because manual tests can not debug errors. Manual testing is more difficult than the automated testing. Test cases can not be obtained directly in manual testing. MBT always needs an intermediate process such as a technique, an intermediate model or an algorithm for manual testing. However, manual testing method is preferred by users in some studies of MBT.

In automated testing, tools are very practical to find all test cases, and tools can reduce errors. However, this state is not valid for every tool, because tools run different from each other, and they can not give the desired results in terms of test cases. Also, tools can be run different from each other. Some tools can find test paths automatically. Some tools can be used to write desired program code. Therefore, tools are very effective in terms of users, and especially free tools are preferred by many users for automated testing method in MBT.

In some studies for MBT, users use some free tools, but these tools are not enough for desired operations for the studies. Commercial tools can find good test paths and test cases, but they are expensive. Therefore, many researchers prefered manual testing for MBT.

In this thesis, both manual and automated testing approach are used for MBT. A UML sequence diagram as a model is chosen. Sequence diagrams are based on scenario-based testing. Scenario-based testing is also named as black-box testing. Sequence diagrams are preferred commonly for scenario-based testing in the MBT. Scenarios are benefiticial to use test requirements of model. In this paper, the same operations with the same examples are applied to both manual and automated testing. Sequence diagrams in the research papers are used as examples, and methods in the research papers are evaluated. First of all, it is produced a tree technique with a new algorithm from a sequence diagram for manual testing. According to this tree, a new algorithm is written by using DFS path algorithm to reach the test paths. Secondly, a new program code is written by using java programming language in eclipse for automated testing. Furthermore, junit testing is preferred since junit is the most popular unit testing framework. It is not preferred any other automated tools to find test paths because automated tools fail to satisfy the requirements of the system. In chapter 3, the same test paths are reached both manual and automated testing. Therefore, automated junit testing code is improved again in chapter 4 because errors that appear automated testing in chapter 3. Different test paths results are reached with improved automated junit code. Manual method, improved automated method and methods in the research papers are compared to each other. According to results of this comparison, improved automated testing method is better than the methods of the research papers and manual testing methods. In this thesis, it is proposed a new

technique for a different view in MBT. As a result of this work, it has been seen that automated testing is better than the manual testing.

5.2 Contributions & Limitations

In this thesis, there are some contributions. First of all, a new technique for MBT have been constituted taking into account existing literature on the model based software testing area. It has been seen that different test paths results are taken from the different techniques and approaches. Thus, in this work it is created a new technique that has better results than existing techniques and approaches. The new technique is constituted by comparing it with the existing techniques and approaches.

Moreover, the proposed technique has some limitations. It is used only sequence diagram not the other diagram for MBT. In this work, it is only considered 10 examples from literature survey about MBT although there are many examples about the MBT.

5.3 Directions for Further Research

In the future, the proposed automated testing should be improved for MBT instead of the manual testing because the manual testing is more expensive than automated testing.

Also, the proposed automated testing should be improved as a free tool in order to contribute to model based software testing. In this way, users can have more practical system for MBT. Besides, the proposed technique can be examined with more examples in the future work.

REFERENCES

[1] **Sawant, V. & Shah, K.,** (2011), "*Automatic Generation of Test Cases from UML Models*", International Conference on Technology Systems and Management (ICTSM), Proceedings published by International Journal of Computer Applications (IJCA), 2011.

[2] **Seth, R. & Anand, S.,** (2012), "*Prioritization of Test Cases scenarios derived from UML Diagrams*", International Journal of Computer Applications (0975 – 8887), Vol: 46, No:12, May 2012.

[3] **Shanthi, A.V.K. & Mohan, Kumar, G.,** (2012), "*Automated Test Cases Generation from UML Sequence Diagram*", International Conference on Software and Computer Applications (ICSCA 2012), Vol: 41, Singapore,2012.

[4] **Khandai, M., Acharya, A. A., Mohapatra, D. P.,** "*Test Case Generation for Concurrent System using UML Combinational Diagram*", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol: 2, Issue:3 , 2011.

[5] **M.Dhineshkumar, PG Scholar, Professor.Mr.Galeebathullah,.** (2014), "*An Approach to Generate Test Cases from Sequence Diagram*", International Conference on Intelligent Computing Applications, IEEE, 2014.

[6] **Ching-Seh Wu, & Chi-Hsin Huang.,** (2013), "*The Web Services Composition Testing Based on Extended Finite State Machine and UML Model*", Fifth International Conference on Service Science and Innovation, IEEE, 2013.

[7] **V.Mary Sumalatha. & G.S.V.P.Raju.,** (2012), "*UML based Automated Test Case Generation technique using Activity-Sequence diagram*", The International Journal of Computer Science & Applications (TIJCSA), Vol: 1, No: 9, November 2012.

[8] **Verma, R. & Bhatia, R.,** (2012), "*Behavior based Automated Test Case Generation for Object Oriented Systems*", International Journal of Computer Applications (0975 – 8887), Vol: 54, No:13, September 2012.

[9] **S. Shanmuga Priya, & P. D. Sheba Kezia Malarchelvi.,** (2013), "*Test Path Generation Using Uml Sequence Diagram*", International Journal of Advanced Research in Computer Science and Software Engineering, Vol: 3, Issue: 4, April 2013.

[10] **V.Mary Sumalatha. & G.S.V.P.Raju, PhD.,** (2013), "*Object Oriented Test Case Generation Technique using Genetic Algorithms*" International Journal of Computer Applications (0975 – 8887) Volume: 61, No:20, January 2013.

[11] **Biswal, B. N., Nanda, P., Mohapatra, D. P.,** "*A Novel Approach for Scenario-Based Test Case Generation*", International Conference on Information Technology, IEEE, 2008

[12] Mark Utting and Bruno Legeard. Practical Model-Based Testing. Morgan Kaufmann Publishers, 2006.

[13] Mark Utting, Alexander Pretschner, and Bruno Legeard. A Taxonomy of Model -based Testing. Technical report, 2006. ISSN 1170-487X, The University of Waikato http://www.cs.waikato.ac.nz/pubs/wp/2006/uow-cs-wp-2006-04.pdf Accessed August 2010.

[14] **Cormen, T. H., Leiserson, C. L., Rivest, R. L., Stein, C.,** (2009). "*Introduction to Algorithms*". Third Edition, The MIT Press Cambridge, Massachusetts London, England

[15] http://www.base36.com/2013/03/automated-vs-manual-testing-the-pros-and-cons-of-each/ Accessed December 2014.

[16] http://www.softwaretestingmentor.com/automation/manual-vs-automation/ Accessed December 2014.

[17] http://www.softwaretestingclass.com/automation-testing-vs-manual-testing/ Accessed December 2014.

[18] **Shirole, M., Kumar, R.,** (2013) "UML Behavioral Model Based Test Case Generation: A Survey" Vol: 38 No: 4. http://doi.acm.org/10.1145/2492248.2492274

[19] **WikiPedia,** http://en.wikipedia.org/wiki/Unified_Modeling_Language

[20] http://aviadezra.blogspot.com.tr/2009/06/uml-fragment-alt-opt-par-loop-region.html Accessed May 2014.

[21] **Dias Neto, A. C.,, Subramanyan, R., Vieira, M., Travassos,G. H.,** (2007) "*A Survey on Model-based Testing Approaches: A Systematic Review*"

[22] **Malik, Q. A.,** "*Combining Model Based Testing and Stepwise Formal Development*", TUCS Dissertations, Åbo Akademi University, Finland, No: 130, September 2010

[23] **Jama, O. M.,** (2009) "*A Case Study on Evaluating State-Based UML Modeling in Software Testing*", Master Thesis, University of Oslo

[24]http://umldiagramtutorial.blogspot.com.tr/2012/11/hospital-management-sequence-diagram.html Accessed May 2014.

APPENDICES

Appendix A – XML Format for Sequence Diagram in Example1

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
  <help_text/>
  <zoom_level>10</zoom_level>
  <element>
    <id>UMLFrame</id>
    <coordinates>
      <x>30</x>
      <y>10</y>
      <w>570</w>
      <h>310</h>
    </coordinates>
    <panel_attributes>ATM</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>90</x>
      <y>50</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_User_</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>240</x>
      <y>50</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_ATM_</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>380</x>
      <y>50</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_Server_</panel_attributes>
    <additional_attributes/>
```

```
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>130</x>
  <y>70</y>
  <w>30</w>
  <h>260</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;240.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>280</x>
  <y>70</y>
  <w>30</w>
  <h>260</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;240.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>420</x>
  <y>70</y>
  <w>30</w>
  <h>260</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;240.0</additional_attributes>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>130</x>
  <y>90</y>
  <w>20</w>
  <h>210</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>280</x>
  <y>90</y>
  <w>20</w>
  <h>210</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>420</x>
  <y>90</y>
  <w>20</w>
  <h>210</h>
```

```
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>130</x>
  <y>70</y>
  <w>170</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>1:InserdCardIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;150.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>290</x>
  <y>80</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>2:Validate(Card)It=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>290</x>
  <y>110</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>3:ReturnIt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>140</x>
  <y>120</y>
  <w>160</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>4:EnterPinIt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>140</x>
  <y>150</y>
  <w>160</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>5:PinIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>280</x>
```

```
    <y>160</y>
    <w>170</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>6:VerifyPinIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>20.0;20.0;140.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>290</x>
    <y>190</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>7:ReturnIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>140</x>
    <y>200</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>8:EnterAmountIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;140.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>140</x>
    <y>230</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>9:AmountIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;140.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>290</x>
    <y>240</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>10:Check(Balance)It=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>290</x>
    <y>270</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>11:ReturnIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
```

```
  <id>Relation</id>
  <coordinates>
    <x>140</x>
    <y>280</y>
    <w>160</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>12:TakeCashIt=&lt;.</panel_attributes>
  <additional_attributes>10.0;20.0;140.0;20.0</additional_attributes>
 </element>
</diagram>
```

Appendix B – XML Format for Sequence Diagram in Example2

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <zoom_level>10</zoom_level>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>140</x>
    <y>20</y>
    <w>120</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_:SessionManager_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>280</x>
    <y>20</y>
    <w>120</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_:DisplayManager_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>420</x>
    <y>20</y>
    <w>100</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_:KeyReader_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>540</x>
    <y>20</y>
    <w>100</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_:Bank_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
```

```
  <coordinates>
    <x>20</x>
    <y>20</y>
    <w>100</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_:CardReader_</panel_attributes>
  <additional_attributes/>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>60</x>
    <y>40</y>
    <w>30</w>
    <h>420</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;400.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>190</x>
    <y>40</y>
    <w>30</w>
    <h>420</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;400.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>330</x>
    <y>40</y>
    <w>30</w>
    <h>420</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;400.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>460</x>
    <y>40</y>
    <w>30</w>
    <h>420</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;400.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>580</x>
    <y>40</y>
    <w>30</w>
    <h>420</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;400.0</additional_attributes>
```

```
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>60</x>
  <y>70</y>
  <w>20</w>
  <h>60</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>190</x>
  <y>70</y>
  <w>20</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>190</x>
  <y>120</y>
  <w>20</w>
  <h>90</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>330</x>
  <y>150</y>
  <w>20</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>330</x>
  <y>220</y>
  <w>20</w>
  <h>60</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>460</x>
  <y>220</y>
  <w>20</w>
  <h>60</h>
```

```
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>60</x>
   <y>280</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>60</x>
   <y>350</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>190</x>
   <y>280</y>
   <w>20</w>
   <h>130</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>580</x>
   <y>290</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>580</x>
   <y>360</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>70</x>
```

```
    <y>60</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>1:cardinsertedlt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>330</x>
    <y>100</y>
    <w>20</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>200</x>
    <y>100</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>2:selectWithDrawlt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>200</x>
    <y>130</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>3:requestAmountlt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>200</x>
    <y>170</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>4:amt_valuelt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>340</x>
    <y>210</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>5:verifyAmount()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
```

```
  <id>Relation</id>
  <coordinates>
    <x>60</x>
    <y>270</y>
    <w>160</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>6:(InvalidAmount)ejectIt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>200</x>
    <y>290</y>
    <w>400</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>7:(ValidAmount)checkBalanceIt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;380.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>50</x>
    <y>340</y>
    <w>180</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>8:(InsufficientAmount)ejectIt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>30.0;20.0;140.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>200</x>
    <y>360</y>
    <w>400</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>9:(SufficientBalance)debit_AccIt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;380.0;20.0</additional_attributes>
 </element>
</diagram>
```

Appendix C – XML Format for Sequence Diagram in Example3

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <zoom_level>10</zoom_level>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>100</x>
    <y>0</y>
    <w>150</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_:SpiderMenuBar_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
```

```
    <x>280</x>
    <y>0</y>
    <w>100</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_:Spider_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>410</x>
    <y>0</y>
    <w>100</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_:PKCard_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>540</x>
    <y>0</y>
    <w>100</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_:Deal_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>170</x>
    <y>20</y>
    <w>30</w>
    <h>470</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;450.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>320</x>
    <y>20</y>
    <w>30</w>
    <h>470</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;450.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>450</x>
    <y>20</y>
    <w>30</w>
    <h>470</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;450.0</additional_attributes>
  </element>
```

```
<element>
 <id>Relation</id>
 <coordinates>
  <x>580</x>
  <y>20</y>
  <w>30</w>
  <h>470</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;450.0</additional_attributes>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>170</x>
  <y>40</y>
  <w>20</w>
  <h>60</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>320</x>
  <y>40</y>
  <w>20</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>320</x>
  <y>100</y>
  <w>20</w>
  <h>360</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>170</x>
  <y>20</y>
  <w>180</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>1:setGame(grade:int):voidlt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>20.0;20.0;150.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>180</x>
  <y>50</y>
  <w>160</w>
  <h>40</h>
 </coordinates>
```

```
 <panel_attributes>Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>330</x>
  <y>90</y>
  <w>130</w>
  <h>60</h>
 </coordinates>
 <panel_attributes>2:newGame():voidlt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>10.0;40.0;60.0;40.0;60.0;10.0;10.0;10.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>290</x>
  <y>130</y>
  <w>210</w>
  <h>60</h>
 </coordinates>
 <panel_attributes>3:ShowEnableOperation():voidlt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>50.0;40.0;100.0;40.0;100.0;10.0;50.0;10.0</additional_attributes>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>450</x>
  <y>180</y>
  <w>20</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>170</y>
  <w>180</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>4:moveto(Point:point):voidlt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>30.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>330</x>
  <y>200</y>
  <w>140</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>580</x>
  <y>230</y>
```

```
    <w>20</w>
    <h>60</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>330</x>
    <y>230</y>
    <w>270</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>5:Deal():voidlt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>330</x>
    <y>260</y>
    <w>270</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>Returnlt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>290</x>
    <y>300</y>
    <w>210</w>
    <h>60</h>
   </coordinates>
   <panel_attributes>6:ShowEnableOperation():voidlt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>50.0;40.0;100.0;40.0;100.0;10.0;50.0;10.0</additional_attributes>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>450</x>
    <y>350</y>
    <w>20</w>
    <h>60</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>310</x>
    <y>350</y>
    <w>180</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>7:moveto(Point:point):voidlt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;140.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
```

```
  <coordinates>
    <x>330</x>
    <y>380</y>
    <w>140</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>Returnlt=&lt;.</panel_attributes>
  <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>300</x>
    <y>420</y>
    <w>200</w>
    <h>50</h>
  </coordinates>
  <panel_attributes>8:haveFinish(column:int):voidlt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>40.0;30.0;90.0;30.0;90.0;10.0;40.0;10.0</additional_attributes>
 </element>
</diagram>
```

Appendix D – XML Format for Sequence Diagram in Example4

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <zoom_level>10</zoom_level>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>150</x>
    <y>20</y>
    <w>100</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_Patient_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>290</x>
    <y>20</y>
    <w>100</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_User_Interface_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
    <x>430</x>
    <y>20</y>
    <w>100</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_Doctor_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
```

```
    <x>570</x>
    <y>20</y>
    <w>100</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_DiagnosisSystem_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>40</y>
    <w>30</w>
    <h>300</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;280.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>330</x>
    <y>40</y>
    <w>30</w>
    <h>300</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;280.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>470</x>
    <y>40</y>
    <w>30</w>
    <h>300</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;280.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>610</x>
    <y>40</y>
    <w>30</w>
    <h>300</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;280.0</additional_attributes>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>190</x>
    <y>70</y>
    <w>20</w>
    <h>230</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
```

```
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>330</x>
  <y>70</y>
  <w>20</w>
  <h>230</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>470</x>
  <y>120</y>
  <w>20</w>
  <h>180</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>610</x>
  <y>70</y>
  <w>20</w>
  <h>230</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>200</x>
  <y>60</y>
  <w>150</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>1:LoginIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>340</x>
  <y>60</y>
  <w>290</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>2:VerifyIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;270.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>340</x>
  <y>80</y>
  <w>290</w>
  <h>40</h>
 </coordinates>
```

```
 <panel_attributes>3:Resultlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;270.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>200</x>
  <y>100</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>4:EnterPatientDetailslt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>340</x>
  <y>110</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>5:Requestlt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>480</x>
  <y>120</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>6:ReferPatientHistorylt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>480</x>
  <y>140</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>7:RetrieveDeatilslt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>230</x>
  <y>160</y>
  <w>370</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>8:DiagnoseandPrescribeMedicine/SuggestAdditionalTestlt=&lt;.</panel_attributes>
 <additional_attributes>120.0;20.0;240.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>150</x>
  <y>190</y>
```

```
    <w>250</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>9:DisplayPrescription/SuggestedTestsIt=&lt;.</panel_attributes>
   <additional_attributes>60.0;20.0;180.0;20.0</additional_attributes>
  </element>
</diagram>
```

Appendix E – XML Format for Sequence Diagram in Example5

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <zoom_level>10</zoom_level>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>70</x>
   <y>0</y>
   <w>100</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>_User_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>180</x>
   <y>0</y>
   <w>100</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>_WebBrowses_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>290</x>
   <y>0</y>
   <w>100</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>_Application_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>420</x>
   <y>0</y>
   <w>190</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>_Facebook__AuthorizationServer_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>630</x>
   <y>0</y>
   <w>110</w>
```

```
    <h>40</h>
   </coordinates>
   <panel_attributes>_Facebook__ContentServer_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>110</x>
    <y>30</y>
    <w>30</w>
    <h>490</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;470.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>220</x>
    <y>30</y>
    <w>30</w>
    <h>490</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;470.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>330</x>
    <y>30</y>
    <w>30</w>
    <h>490</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;470.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>510</x>
    <y>30</y>
    <w>30</w>
    <h>490</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;470.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>680</x>
    <y>30</y>
    <w>30</w>
    <h>490</h>
   </coordinates>
   <panel_attributes>lt=.</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;470.0</additional_attributes>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
```

```
    <x>110</x>
    <y>60</y>
    <w>20</w>
    <h>340</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>220</x>
    <y>60</y>
    <w>20</w>
    <h>120</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>220</x>
    <y>210</y>
    <w>20</w>
    <h>190</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>330</x>
    <y>80</y>
    <w>20</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>510</x>
    <y>130</y>
    <w>20</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>510</x>
    <y>210</y>
    <w>20</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
```

```
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>510</x>
  <y>260</y>
  <w>20</w>
  <h>60</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>330</x>
  <y>260</y>
  <w>20</w>
  <h>120</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>680</x>
  <y>310</y>
  <w>20</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>110</x>
  <y>450</y>
  <w>20</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>220</x>
  <y>410</y>
  <w>20</w>
  <h>70</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>330</x>
  <y>410</y>
  <w>20</w>
  <h>70</h>
 </coordinates>
```

```
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>80</x>
  <y>50</y>
  <w>200</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>1:getFB_Resource(int_a,int_b)lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>50.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>210</x>
  <y>70</y>
  <w>160</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>2:request_FB_accesslt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>30.0;20.0;120.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>230</x>
  <y>90</y>
  <w>120</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>3:http_redirectlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;100.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>230</x>
  <y>110</y>
  <w>310</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>4:authorize(int_a)lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;290.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>230</x>
  <y>130</y>
  <w>300</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>5:permission_form(int_a,int_b,int_c)lt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;280.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>150</y>
```

```
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>6:permission_formlt=&lt;.</panel_attributes>
   <additional_attributes>20.0;20.0;110.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>90</x>
    <y>200</y>
    <w>180</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>7:user_permission(int_a)lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>40.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>230</x>
    <y>200</y>
    <w>300</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>8:process_permissionlt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;280.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>230</x>
    <y>220</y>
    <w>300</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>9:http_redirectlt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;280.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>200</x>
    <y>240</y>
    <w>180</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>10:FB_authorization_codelt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>40.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>290</x>
    <y>260</y>
    <w>290</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>11:FB_authorization_code(int_a,int_b,int_c)lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>60.0;20.0;220.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
```

```
  <coordinates>
    <x>340</x>
    <y>280</y>
    <w>190</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>12:access_tokenlt=&lt;.</panel_attributes>
  <additional_attributes>10.0;20.0;170.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>340</x>
    <y>310</y>
    <w>360</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>13:access_FB_user_protected_resourcelt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;340.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>340</x>
    <y>330</y>
    <w>360</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>14:user_protected_resourcelt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;340.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>190</x>
    <y>350</y>
    <w>200</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>15:user_protected_resourcelt=&lt;.</panel_attributes>
  <additional_attributes>50.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>80</x>
    <y>370</y>
    <w>200</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>16:user_protected_resourcelt=&lt;.</panel_attributes>
  <additional_attributes>50.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>150</x>
    <y>400</y>
    <w>270</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>17:no_authorization(int_a,int_b,int_c,intd)lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>80.0;20.0;180.0;20.0</additional_attributes>
```

```
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>190</x>
   <y>420</y>
   <w>200</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>18:FB_resource_not_availableIt=&amp;lt;.</panel_attributes>
  <additional_attributes>50.0;20.0;140.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>70</x>
   <y>440</y>
   <w>220</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>19:FB_resource_not_availableIt=&amp;lt;.</panel_attributes>
  <additional_attributes>60.0;20.0;150.0;20.0</additional_attributes>
 </element>
</diagram>
```

Appendix F – XML Format for Sequence Diagram in Example6

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <zoom_level>10</zoom_level>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>30</x>
   <y>60</y>
   <w>100</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>_OperatorPanel_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>160</x>
   <y>60</y>
   <w>100</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>_ATM_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>290</x>
   <y>60</y>
   <w>100</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>_CashDispenser_</panel_attributes>
  <additional_attributes/>
 </element>
```

```
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>420</x>
  <y>60</y>
  <w>100</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>_Bank_</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>70</x>
  <y>80</y>
  <w>30</w>
  <h>280</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>200</x>
  <y>80</y>
  <w>30</w>
  <h>280</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>330</x>
  <y>80</y>
  <w>30</w>
  <h>280</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>460</x>
  <y>80</y>
  <w>30</w>
  <h>280</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>70</x>
  <y>200</y>
  <w>20</w>
  <h>60</h>
 </coordinates>
```

```
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>200</x>
   <y>110</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>200</x>
   <y>180</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>200</x>
   <y>250</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>330</x>
   <y>270</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>460</x>
   <y>270</y>
   <w>20</w>
   <h>60</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>70</x>
   <y>100</y>
```

```
    <w>150</w>
    <h>50</h>
   </coordinates>
   <panel_attributes>1:Switch_on()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>200</x>
    <y>130</y>
    <w>130</w>
    <h>80</h>
   </coordinates>
   <panel_attributes>2:PerformStartUp()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;60.0;60.0;60.0;60.0;10.0;20.0;10.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>80</x>
    <y>190</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>3:GetInitialCash()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>80</x>
    <y>240</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>4:LoadingCash()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>210</x>
    <y>260</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>5:SetCash()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>340</x>
    <y>280</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>6:OpenConnection()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
</diagram>
```

Appendix G – XML Format for Sequence Diagram in Example7

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
  <zoom_level>10</zoom_level>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>10</x>
      <y>70</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_Customer_</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>120</x>
      <y>70</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_:BookStore_</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>230</x>
      <y>70</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_:Inventory_</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>340</x>
      <y>70</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_:Payment_</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>450</x>
      <y>70</y>
      <w>100</w>
      <h>30</h>
    </coordinates>
    <panel_attributes>_:Delivery_</panel_attributes>
    <additional_attributes/>
  </element>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
```

```
    <x>560</x>
    <y>70</y>
    <w>100</w>
    <h>30</h>
  </coordinates>
  <panel_attributes>_:Message_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>50</x>
    <y>90</y>
    <w>30</w>
    <h>280</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>160</x>
    <y>90</y>
    <w>30</w>
    <h>280</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>270</x>
    <y>90</y>
    <w>30</w>
    <h>280</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>380</x>
    <y>90</y>
    <w>30</w>
    <h>280</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
    <x>490</x>
    <y>90</y>
    <w>30</w>
    <h>280</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
 </element>
```

```
<element>
 <id>Relation</id>
 <coordinates>
  <x>600</x>
  <y>90</y>
  <w>30</w>
  <h>280</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;260.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>20</x>
  <y>110</y>
  <w>200</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>1:Search(bookname,author)lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>40.0;20.0;150.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>160</x>
  <y>130</y>
  <w>140</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>2:checkStock(ISBN)lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>50</x>
  <y>150</y>
  <w>140</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>3:buy(id,ISBN,gty)lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>160</x>
  <y>180</y>
  <w>250</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>4:makePayment(id,totalprice)lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>130</x>
  <y>210</y>
  <w>200</w>
  <h>40</h>
 </coordinates>
```

```
  <panel_attributes>5:ship(ISBN,decrease_NUM)lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>40.0;20.0;150.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>270</x>
   <y>240</y>
   <w>250</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>6:deliver()lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>270</x>
   <y>280</y>
   <w>360</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>7:sendMessage()lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;340.0;20.0</additional_attributes>
 </element>
</diagram>
```

Appendix H – XML Format for Sequence Diagram in Example8

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <help_text/>
 <zoom_level>10</zoom_level>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>30</x>
   <y>0</y>
   <w>100</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>_CardReader_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>140</x>
   <y>0</y>
   <w>110</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>_DisplayManager_</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>260</x>
```

```
    <y>0</y>
    <w>110</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_SessionManager_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>380</x>
    <y>0</y>
    <w>120</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_PasswordReader_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>510</x>
    <y>0</y>
    <w>100</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_UserAcount_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>620</x>
    <y>0</y>
    <w>120</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_LibraryDatabase_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>70</x>
    <y>20</y>
    <w>30</w>
    <h>670</h>
   </coordinates>
   <panel_attributes>lt=-</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;650.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>20</y>
    <w>30</w>
```

```
    <h>670</h>
   </coordinates>
   <panel_attributes>lt=-</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;650.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>310</x>
    <y>20</y>
    <w>30</w>
    <h>670</h>
   </coordinates>
   <panel_attributes>lt=-</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;650.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>430</x>
    <y>20</y>
    <w>30</w>
    <h>670</h>
   </coordinates>
   <panel_attributes>lt=-</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;650.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>550</x>
    <y>20</y>
    <w>30</w>
    <h>670</h>
   </coordinates>
   <panel_attributes>lt=-</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;650.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>670</x>
    <y>20</y>
    <w>30</w>
    <h>670</h>
   </coordinates>
   <panel_attributes>lt=-</panel_attributes>
   <additional_attributes>10.0;10.0;10.0;650.0</additional_attributes>
  </element>
  <element>
   <id>UMLFrame</id>
   <coordinates>
    <x>10</x>
    <y>50</y>
    <w>750</w>
    <h>600</h>
   </coordinates>
```

```
 <panel_attributes>Alt</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>70</x>
  <y>20</y>
  <w>270</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>1:CardValueM1()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>40</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>2:InvalidCardM2()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>50</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>3:PWValueM3()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>60</y>
  <w>150</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>UMLFrame</id>
 <coordinates>
  <x>20</x>
  <y>90</y>
  <w>730</w>
  <h>550</h>
 </coordinates>
 <panel_attributes>Alt</panel_attributes>
 <additional_attributes/>
```

```
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>80</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>4:InvalidPWM4()lt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>80</y>
  <w>150</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>90</y>
  <w>390</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>5:AccountValueM5()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;370.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>100</y>
  <w>510</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;490.0;20.0</additional_attributes>
</element>
<element>
 <id>UMLFrame</id>
 <coordinates>
  <x>30</x>
  <y>130</y>
  <w>710</w>
  <h>500</h>
 </coordinates>
 <panel_attributes>Alt</panel_attributes>
 <additional_attributes/>
</element>
<element>
```

```
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>120</y>
  <w>150</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>6:AccountCreatedM6()lt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>UMLFrame</id>
 <coordinates>
  <x>40</x>
  <y>150</y>
  <w>690</w>
  <h>310</h>
 </coordinates>
 <panel_attributes>Alt</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>140</y>
  <w>510</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>7:CloseAccountM7()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;490.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>150</y>
  <w>510</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;490.0;20.0</additional_attributes>
</element>
<element>
 <id>UMLFrame</id>
 <coordinates>
  <x>50</x>
  <y>180</y>
  <w>670</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>Par</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
```

```
    <x>310</x>
    <y>180</y>
    <w>390</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>8:DestroyUserAccountM8()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;370.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>310</x>
    <y>190</y>
    <w>270</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>9:SuspendICM9()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>210</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>10:ContinueM10()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>220</y>
    <w>150</w>
    <h>50</h>
   </coordinates>
   <panel_attributes>ReturnIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>UMLFrame</id>
   <coordinates>
    <x>50</x>
    <y>250</y>
    <w>670</w>
    <h>200</h>
   </coordinates>
   <panel_attributes>Alt</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>240</y>
```

```
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>11:ReturnBookM11()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>UMLFrame</id>
   <coordinates>
    <x>70</x>
    <y>270</y>
    <w>640</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>Par</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>310</x>
    <y>260</y>
    <w>390</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>12:UpdateLibraryDatabaseM12()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;370.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>310</x>
    <y>280</y>
    <w>270</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>13:UpdateUserAcountM13()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>300</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>14:IssueBookM14()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>310</x>
    <y>310</y>
    <w>390</w>
    <h>40</h>
```

```
 </coordinates>
 <panel_attributes>15:BOOKIdM15()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;370.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>320</y>
  <w>390</w>
  <h>50</h>
 </coordinates>
 <panel_attributes>Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;370.0;20.0</additional_attributes>
</element>
<element>
 <id>UMLFrame</id>
 <coordinates>
  <x>60</x>
  <y>350</y>
  <w>650</w>
  <h>90</h>
 </coordinates>
 <panel_attributes>Alt</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>170</x>
  <y>340</y>
  <w>190</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>16:BookNotAvailableM16()lt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>30.0;20.0;150.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>180</x>
  <y>360</y>
  <w>170</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>17:BookAvailableM17()lt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>20.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>UMLFrame</id>
 <coordinates>
  <x>70</x>
  <y>390</y>
  <w>630</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>Par</panel_attributes>
```

```
  <additional_attributes/>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>310</x>
    <y>380</y>
    <w>390</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>18:UpdateLibraryDatabaseM18()lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;370.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>310</x>
    <y>400</y>
    <w>270</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>19:UpdateUserAcountM19()lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>170</x>
    <y>450</y>
    <w>190</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>20:AcountNotCreatedM20()lt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>30.0;20.0;150.0;20.0</additional_attributes>
</element>
<element>
  <id>Relation</id>
  <coordinates>
    <x>190</x>
    <y>460</y>
    <w>150</w>
    <h>50</h>
  </coordinates>
  <panel_attributes>Returnlt=.&gt;</panel_attributes>
  <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
  <id>UMLFrame</id>
  <coordinates>
    <x>40</x>
    <y>490</y>
    <w>690</w>
    <h>130</h>
  </coordinates>
  <panel_attributes>Par</panel_attributes>
  <additional_attributes/>
</element>
```

```
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>490</y>
  <w>270</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>21:GetNameM21()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>510</y>
  <w>270</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>22:GetF/MM22()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>530</y>
  <w>270</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>23:GetDobM23()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>550</y>
  <w>270</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>24:GetAddressM24()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>570</y>
  <w>270</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>25:GetPhnoM25()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
```

```
  <coordinates>
    <x>310</x>
    <y>590</y>
    <w>270</w>
    <h>40</h>
  </coordinates>
  <panel_attributes>26:GetAffiliationM26()lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;250.0;20.0</additional_attributes>
 </element>
</diagram>
```

Appendix I – XML Format for Sequence Diagram in Example9

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <help_text/>
 <zoom_level>10</zoom_level>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>140</x>
   <y>0</y>
   <w>100</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>C:Customer</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>270</x>
   <y>0</y>
   <w>100</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>A:ATM</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>400</x>
   <y>0</y>
   <w>100</w>
   <h>30</h>
  </coordinates>
  <panel_attributes>Accnt:Acount</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>180</x>
   <y>20</y>
   <w>30</w>
   <h>410</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;390.0</additional_attributes>
 </element>
 <element>
```

```
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>20</y>
  <w>30</w>
  <h>410</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;390.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>440</x>
  <y>20</y>
  <w>30</w>
  <h>410</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;390.0</additional_attributes>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>530</x>
  <y>0</y>
  <w>160</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>CKA:Checking Acount</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>600</x>
  <y>20</y>
  <w>30</w>
  <h>410</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;390.0</additional_attributes>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>180</x>
  <y>40</y>
  <w>20</w>
  <h>360</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>310</x>
  <y>40</y>
  <w>20</w>
  <h>360</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
```

```
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>440</x>
  <y>70</y>
  <w>20</w>
  <h>60</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>440</x>
  <y>190</y>
  <w>20</w>
  <h>70</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>600</x>
  <y>190</y>
  <w>20</w>
  <h>70</h>
 </coordinates>
 <panel_attributes>//Line</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>30</y>
  <w>140</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>1:InsertAtmCard()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>50</y>
  <w>140</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>2:RequestPin()lt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>190</x>
  <y>70</y>
  <w>140</w>
```

```
    <h>40</h>
   </coordinates>
   <panel_attributes>3:PinEntered()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>320</x>
    <y>80</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>4:VerifyPin()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>320</x>
    <y>100</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>5:PinOK()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>110</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>6:RequestOption()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>130</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>7:OptionEntered()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>150</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>8:RequestAmount()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
```

```
    <x>190</x>
    <y>170</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>9:AmountEntered()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>310</x>
    <y>180</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>10:ProcessTransaction()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>430</x>
    <y>200</y>
    <w>210</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>11:WithdrawCheckingAcount()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;170.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>450</x>
    <y>220</y>
    <w>170</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>12:WithdrawSuccessful()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;150.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>300</x>
    <y>230</y>
    <w>180</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>13:TransactionSuccessful()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;140.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>190</x>
    <y>240</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>14:DispenseCash()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
```

```
 <element>
  <id>Relation</id>
  <coordinates>
   <x>180</x>
   <y>260</y>
   <w>160</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>15:RequestTakeCash()lt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>190</x>
   <y>280</y>
   <w>140</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>16:TakeCash()lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>170</x>
   <y>300</y>
   <w>180</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>17:RequestContinuation()lt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>30.0;20.0;140.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>190</x>
   <y>320</y>
   <w>140</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>18:Terminate()lt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>190</x>
   <y>340</y>
   <w>140</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>19:PrintReceipt()lt=&lt;&lt;-//syncMsg</panel_attributes>
  <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
 </element>
</diagram>
```

Appendix J – XML Format for Sequence Diagram in Example10

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
 <zoom_level>10</zoom_level>
 <element>
```

```
 <id>UMLGeneric</id>
 <coordinates>
  <x>70</x>
  <y>0</y>
  <w>100</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>_student_</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>180</x>
  <y>0</y>
  <w>100</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>_browser_</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>290</x>
  <y>0</y>
  <w>100</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>_webServer_</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>400</x>
  <y>0</y>
  <w>110</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>_registrationPage_</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>UMLGeneric</id>
 <coordinates>
  <x>640</x>
  <y>0</y>
  <w>100</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>_dB_</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>20</y>
  <w>30</w>
  <h>600</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
```

```
  <additional_attributes>10.0;10.0;10.0;580.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>220</x>
   <y>20</y>
   <w>30</w>
   <h>600</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;580.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>330</x>
   <y>20</y>
   <w>30</w>
   <h>600</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;580.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>440</x>
   <y>20</y>
   <w>30</w>
   <h>600</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;580.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>680</x>
   <y>20</y>
   <w>30</w>
   <h>600</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;580.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>100</x>
   <y>30</y>
   <w>160</w>
   <h>40</h>
  </coordinates>
  <panel_attributes>1:browse_home_pagelt=-&gt;&gt;//syncMsg</panel_attributes>
  <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>110</x>
   <y>50</y>
   <w>140</w>
```

```
    <h>40</h>
   </coordinates>
   <panel_attributes>2:abortIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>100</x>
    <y>80</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>3:browse_home_pageIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>210</x>
    <y>100</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>4:request_home_pageIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>220</x>
    <y>120</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>5:abortIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>100</x>
    <y>130</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>6:browse_home_pageIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>210</x>
    <y>150</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>7:request_home_pageIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>20.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
```

```
    <x>220</x>
    <y>170</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>8:responseIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>90</x>
    <y>180</y>
    <w>180</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>9:select_registration_pageIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;140.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>180</x>
    <y>200</y>
    <w>220</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>10:requesting_registration_pageIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>50.0;20.0;160.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>110</x>
    <y>220</y>
    <w>250</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>11:ask_user_to_sign_upIt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>110</x>
    <y>240</y>
    <w>250</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>12:user_sign_upIt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>520</x>
    <y>160</y>
    <w>100</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_aStudent_</panel_attributes>
   <additional_attributes/>
  </element>
```

```
<element>
 <id>Relation</id>
 <coordinates>
  <x>560</x>
  <y>180</y>
  <w>30</w>
  <h>390</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;370.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>260</y>
  <w>250</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>13:abortlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>280</y>
  <w>250</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>14:user_sign_uplt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>300</y>
  <w>250</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>15:abortlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>320</y>
  <w>250</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>16:user_sign_uplt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>340</y>
  <w>250</w>
  <h>40</h>
 </coordinates>
```

```
 <panel_attributes>17:abortIt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>110</x>
  <y>360</y>
  <w>250</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>18:user_sign_upIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;230.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>330</x>
  <y>380</y>
  <w>140</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>19:new_pageIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>310</x>
  <y>400</y>
  <w>180</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>20:prepare_and_initializeIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>30.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>420</x>
  <y>420</y>
  <w>190</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>21:create_new_student_IDIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>30.0;20.0;150.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>430</x>
  <y>440</y>
  <w>170</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>22:get_registration_infoIt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>20.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>540</x>
  <y>460</y>
```

```
    <w>190</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>23:get_student_registrationlt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;150.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>560</x>
    <y>480</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>24:return_datalt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;130.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>420</x>
    <y>500</y>
    <w>190</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>25:return_registration_datalt=&lt;.</panel_attributes>
   <additional_attributes>30.0;20.0;150.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>330</x>
    <y>510</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>26:returnlt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;120.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>90</x>
    <y>520</y>
    <w>290</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>27:return_registration_data_and_student_IDlt=&lt;.</panel_attributes>
   <additional_attributes>30.0;20.0;250.0;20.0</additional_attributes>
  </element>
</diagram>
```

Appendix K – XML Format for Sequence Diagram in Figure 20

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diagram program="umlet" version="13.0">
  <help_text/>
  <zoom_level>10</zoom_level>
  <element>
    <id>UMLGeneric</id>
    <coordinates>
      <x>50</x>
```

```
    <y>0</y>
    <w>80</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_Doctor_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>140</x>
    <y>0</y>
    <w>80</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_Patient_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>230</x>
    <y>0</y>
    <w>90</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_Receptionist_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>330</x>
    <y>0</y>
    <w>90</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_Department_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>430</x>
    <y>0</y>
    <w>80</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_Rooms_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>520</x>
    <y>0</y>
    <w>80</w>
    <h>30</h>
   </coordinates>
   <panel_attributes>_Person_</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
```

```
 <id>UMLGeneric</id>
 <coordinates>
  <x>610</x>
  <y>0</y>
  <w>40</w>
  <h>30</h>
 </coordinates>
 <panel_attributes>_Bill_</panel_attributes>
 <additional_attributes/>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>80</x>
  <y>20</y>
  <w>30</w>
  <h>610</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;590.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>170</x>
  <y>20</y>
  <w>30</w>
  <h>610</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;590.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>270</x>
  <y>20</y>
  <w>30</w>
  <h>610</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;590.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>370</x>
  <y>20</y>
  <w>30</w>
  <h>610</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
 <additional_attributes>10.0;10.0;10.0;590.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>460</x>
  <y>20</y>
  <w>30</w>
  <h>610</h>
 </coordinates>
 <panel_attributes>lt=.</panel_attributes>
```

```
  <additional_attributes>10.0;10.0;10.0;590.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>550</x>
   <y>20</y>
   <w>30</w>
   <h>610</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;590.0</additional_attributes>
 </element>
 <element>
  <id>Relation</id>
  <coordinates>
   <x>620</x>
   <y>20</y>
   <w>30</w>
   <h>610</h>
  </coordinates>
  <panel_attributes>lt=.</panel_attributes>
  <additional_attributes>10.0;10.0;10.0;590.0</additional_attributes>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>80</x>
   <y>40</y>
   <w>20</w>
   <h>570</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>170</x>
   <y>40</y>
   <w>20</w>
   <h>570</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>270</x>
   <y>40</y>
   <w>20</w>
   <h>570</h>
  </coordinates>
  <panel_attributes>//Line</panel_attributes>
  <additional_attributes/>
 </element>
 <element>
  <id>UMLGeneric</id>
  <coordinates>
   <x>370</x>
   <y>40</y>
   <w>20</w>
```

```
    <h>570</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>460</x>
    <y>40</y>
    <w>20</w>
    <h>570</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>550</x>
    <y>40</y>
    <w>20</w>
    <h>570</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>UMLGeneric</id>
   <coordinates>
    <x>620</x>
    <y>40</y>
    <w>20</w>
    <h>570</h>
   </coordinates>
   <panel_attributes>//Line</panel_attributes>
   <additional_attributes/>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>60</x>
    <y>30</y>
    <w>160</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>1:PrescribeMedicines()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>40.0;20.0;110.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>70</x>
    <y>60</y>
    <w>140</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>2:PrescribeTests()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;100.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
```

```
    <x>80</x>
    <y>90</y>
    <w>120</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>3:CheckReports()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>20.0;20.0;90.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>160</x>
    <y>110</y>
    <w>150</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>4:RequestForRoom()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;110.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>280</x>
    <y>120</y>
    <w>200</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>5:CheckRoomAvailability()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;180.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>280</x>
    <y>150</y>
    <w>200</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>6:ReturnStatuslt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;180.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>180</x>
    <y>160</y>
    <w>110</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>7:ReturnStatuslt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;90.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>280</x>
    <y>180</y>
    <w>200</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>8:IfAvailableBookRoom()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;180.0;20.0</additional_attributes>
  </element>
```

```
<element>
 <id>Relation</id>
 <coordinates>
  <x>280</x>
  <y>210</y>
  <w>200</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>9:ReturnRoomNolt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;180.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>170</x>
  <y>220</y>
  <w>130</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>10:ReturnRoomNolt=&lt;.</panel_attributes>
 <additional_attributes>20.0;20.0;100.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>140</x>
  <y>250</y>
  <w>190</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>11:MaintainPatientDetails()lt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>50.0;20.0;130.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>280</x>
  <y>260</y>
  <w>360</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>12:GenerateBills()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;340.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>170</x>
  <y>280</y>
  <w>130</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>13:GenerateBills()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>20.0;20.0;100.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>130</x>
  <y>310</y>
  <w>210</w>
  <h>40</h>
 </coordinates>
```

```
 <panel_attributes>14:ReturnDischargeSummarylt=&lt;.</panel_attributes>
 <additional_attributes>60.0;20.0;140.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>180</x>
  <y>320</y>
  <w>390</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>15:AddPerson()lt=&lt;&lt;-//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;370.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>90</x>
  <y>350</y>
  <w>550</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>16:DrawSalary()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;530.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>90</x>
  <y>380</y>
  <w>550</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>17:Return()lt=&lt;&lt;-//asyncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;530.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>280</x>
  <y>400</y>
  <w>360</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>18:DrawSalary()lt=-&gt;&gt;//syncMsg</panel_attributes>
 <additional_attributes>10.0;20.0;340.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>280</x>
  <y>430</y>
  <w>360</w>
  <h>40</h>
 </coordinates>
 <panel_attributes>19:Returnlt=&lt;.</panel_attributes>
 <additional_attributes>10.0;20.0;340.0;20.0</additional_attributes>
</element>
<element>
 <id>Relation</id>
 <coordinates>
  <x>540</x>
  <y>450</y>
```

```
    <w>120</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>20:DrawSalary()lt=-&gt;&gt;//syncMsg</panel_attributes>
   <additional_attributes>30.0;20.0;80.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>560</x>
    <y>480</y>
    <w>80</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>21:Return()lt=&lt;.</panel_attributes>
   <additional_attributes>10.0;20.0;60.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>90</x>
    <y>490</y>
    <w>480</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>22:EditPerson()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;460.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>90</x>
    <y>520</y>
    <w>480</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>23:DeletePerson()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;460.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>90</x>
    <y>550</y>
    <w>300</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>24:AddDoctor()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;280.0;20.0</additional_attributes>
  </element>
  <element>
   <id>Relation</id>
   <coordinates>
    <x>90</x>
    <y>580</y>
    <w>300</w>
    <h>40</h>
   </coordinates>
   <panel_attributes>25:DeleteDoctor()lt=&lt;&lt;-//syncMsg</panel_attributes>
   <additional_attributes>10.0;20.0;280.0;20.0</additional_attributes>
  </element>
</diagram
```

Appendix L – Automatic Testing Junit Code (‘*JUnitCode*’)

```
package test;

import java.util.ArrayList;
import java.util.List;
import junit.framework.TestCase;

public class JUnitCode extends TestCase {
public void testPossibleTestPaths() {
Table name = new Table();
List<String> numdata = new ArrayList<String>();
List<String> numdatalist2 = new ArrayList<String>();
int i = 1;
int blank_value = 0;
System.out.println("");
System.out.println("TEST PATHS");
System.out.println("");
System.out.println("All Possible Test Paths...");
System.out.println("");
for (int count = 0; count < name.getRowCount(); count++){
if( name.getValueAt(count, 0).equals("Relation") && name.getValueAt(count,
1).toString().contains("lt=->>")){ //lt=->> is syncMsg
String[] parts = name.getValueAt(count, 1).toString().split(":");
String parts1 = parts[0]; //Desired Message Part
//String parts2 = parts[1]; //Undesired Message Part
numdata.add(parts1);}
if( name.getValueAt(count, 0).equals("Relation") && name.getValueAt(count,
1).toString().contains("lt=<<-")){ //lt=->> is syncMsg
String[] string = name.getValueAt(count, 1).toString().split(":");
String string1 = string[0]; //Desired Message Part
//String string2 = string[1]; //Undesired Message Part
numdata.add(string1);}
if( name.getValueAt(count, 0).equals("Relation") && name.getValueAt(count,
1).toString().contains("lt=<.")){ //lt=<. is return message
System.out.print(i +". ");
String[] part = name.getValueAt(count, 1).toString().split(":");
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
numdata.add(par1);
numdatalist2.add(par1);
if( ! name.getValueAt(count, 1).toString().contains(":")){
numdata.remove(par1);
numdatalist2.remove(par1);
String value = Integer.toString(blank_value);
numdatalist2.add(value);}
par1 = numdata.toString().replace(", ", " - ");
System.out.println(par1);
i++;}
if( name.getValueAt(count, 0).equals("Relation") && name.getValueAt(count,
1).toString().contains("lt=.>")){ //lt=.> is return message
System.out.print(i +". ");
String[] part_no = name.getValueAt(count, 1).toString().split(":");
String part_no1 = part_no[0]; //Desired Message Part
//String part_no2 = part_no[1]; //Undesired Message Part
numdata.add(part_no1);
numdatalist2.add(part_no1);
if( ! name.getValueAt(count, 1).toString().contains(":")){
numdata.remove(part_no1);
numdatalist2.remove(part_no1);
String value = Integer.toString(blank_value);
numdatalist2.add(value);}
part_no1 = numdata.toString().replace(", ", " - ");
System.out.println(part_no1);
i++;}}
String lastElement_numdata=null;
lastElement_numdata=numdata.get(numdata.size()-1); //last element of numdata list.
if(numdatalist2.contains(lastElement_numdata)){
System.out.println("");}
else{
String list_no = numdata.toString().replace(", ", " - ");
System.out.println(i + ". " + list_no);}}
```

```
public void testAllPaths() {
Table name = new Table();
List<String> listdata = new ArrayList<String>();
List<String> listdata2 = new ArrayList<String>();
int i=1;
System.out.println("");
for (int count = 0; count < name.getRowCount(); count++){
if(  name.getValueAt(count, 0).equals("Relation") && name.getValueAt(count,
1).toString().contains("lt=->>")){
String[] parts = name.getValueAt(count, 1).toString().split("lt=->>");
String parts1 = parts[0]; //Desired Message Part
//String parts2 = parts[1]; //Undesired Message Part
listdata.add(parts1);}
if(  name.getValueAt(count, 0).equals("Relation") && name.getValueAt(count,
1).toString().contains("lt=<<-")){
String[] parts = name.getValueAt(count, 1).toString().split("lt=<<-");
String parts1 = parts[0]; //Desired Message Part
//String parts2 = parts[1]; //Undesired Message Part
listdata.add(parts1);}
if( name.getValueAt(count, 1).toString().contains("lt=<.")){
System.out.print(i +". ");
String[] part = name.getValueAt(count, 1).toString().split("lt=<.");
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
listdata.add(par1);
listdata2.add(par1);
String str1 = listdata.toString().replace(", ", " - ");
System.out.println(str1);
i++;}
if( name.getValueAt(count, 1).toString().contains("lt=.>")){
System.out.print(i +". ");
String[] part = name.getValueAt(count, 1).toString().split("lt=.>");
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
listdata.add(par1);
listdata2.add(par1);
String str1 = listdata.toString().replace(", ", " - ");
System.out.println(str1);
i++;}}
String lastElement_listdata=null;
String lastElement_listdata2=null;
lastElement_listdata=listdata.get(listdata.size()-1);
lastElement_listdata2=listdata2.get(listdata2.size()-1);
if(lastElement_listdata == lastElement_listdata2){
System.out.println("");}
else{
String list_no = listdata.toString().replace(", ", " - ");
System.out.println(i + ". " + list_no);}}
public void testBestPath() {
Table table = new Table();
List<String> numdata_best = new ArrayList<String>();
List<String> numdatalist = new ArrayList<String>();
int path_count=0;
for (int count = 0; count < table.getRowCount(); count++){
if(  table.getValueAt(count, 0).equals("Relation") && table.getValueAt(count,
1).toString().contains("lt=->>")){
String[] parts = table.getValueAt(count, 1).toString().split("lt=->>");
String part1 = parts[0]; //Desired Message Part
//String part2 = parts[1]; //Undesired Message Part
numdata_best.add(part1);}
if(  table.getValueAt(count, 0).equals("Relation") && table.getValueAt(count,
1).toString().contains("lt=<<-")){
String[] s = table.getValueAt(count, 1).toString().split("lt=<<-");
String s1 = s[0]; //Desired Message Part
//String s2 = s[1]; //Undesired Message Part
numdata_best.add(s1);
if(table.getValueAt(count, 1).toString().contains("lt=<.")){
String[] part = table.getValueAt(count, 1).toString().split("lt=<.");
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
numdata_best.add(par1);
numdatalist.add(par1);
path_count++;}
if(table.getValueAt(count, 1).toString().contains("lt=.>")){
String[] part = table.getValueAt(count, 1).toString().split("lt=.>");
```

```
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
numdata_best.add(par1);
numdatalist.add(par1);
path_count++;}}
String lastElement_numdata_best=null;
String lastElement_numdatalist=null;
lastElement_numdata_best=numdata_best.get(numdata_best.size()-1);
lastElement_numdatalist=numdatalist.get(numdatalist.size()-1);
if(lastElement_numdata_best == lastElement_numdatalist){
System.out.println("");}
else{
path_count++;}
String list_no = numdata_best.toString().replace(", ", " - ");
System.out.println("BEST PATH");
System.out.println("");
System.out.println("The Best Path: " + path_count + ". " + " Path " + list_no);
System.out.println("");}
public void testNames() {
Table data = new Table();
int index = 0;
assertNotNull(data.getColumnName(index));}
public void testCount() {
Table table = new Table();
int columnIndex = 0;
int rowIndex = 0;
assertNotNull(table.getValueAt(rowIndex, columnIndex));
assertNotNull(table.getColumnCount());
assertNotNull(table.getRowCount());}
public void testReadMessages() {
Table message = new Table();
List<String> data = new ArrayList<String>();
String string=null;
for (int count = 0; count < message.getRowCount(); count++){
if(message.getValueAt(count, 0).equals("Relation") && message.getValueAt(count,
1).toString().contains("lt=->>")){
String[] list = message.getValueAt(count, 1).toString().split("lt=->>//");
String list1 = list[0]; //Desired Message Part
//String list2 = list[1]; //Undesired Message Part
data.add(list1);
string = data.toString().replace(", ", " -> ");}
if(message.getValueAt(count, 0).equals("Relation") && message.getValueAt(count,
1).toString().contains("lt=<<-")){
String[] list1 = message.getValueAt(count, 1).toString().split("lt=<<-//");
String list_1 = list1[0]; //Desired Message Part
//String list_2 = list1[1]; //Undesired Message Part
data.add(list_1);
string = data.toString().replace(", ", " -> ");}
if(message.getValueAt(count, 1).toString().contains("lt=<.")){
String[] start = message.getValueAt(count, 1).toString().split("lt=<.");
String start1 = start[0]; //Desired Message Part
//String start2 = start[1]; //Undesired Message Part
data.add(start1);
string = data.toString().replace(", ", " - ");}
if(message.getValueAt(count, 1).toString().contains("lt=.>")){
String[] start = message.getValueAt(count, 1).toString().split("lt=.>");
String start1 = start[0]; //Desired Message Part
//String start2 = start[1]; //Undesired Message Part
data.add(start1);
string = data.toString().replace(", ", " - ");}}
System.out.println("ALL MESSAGES");
System.out.println("");
System.out.println("All messages of sequence diagram: " + string);
System.out.println("");
System.out.println("Total message number of sequence diagram: " + data.size());}
}
```

Appendix M – Improved Automatic Testing Junit Code (‘*JunitCode2*’)

```
package test2;

import java.util.ArrayList;
import java.util.List;
import junit.framework.TestCase;

public class JunitCode2 extends TestCase {

public void testPossibleTestPaths() { //This function presents all possible path
sequence in the sequence diagram.
Table name = new Table();
List<String> numdata = new ArrayList<String>();
List<String> numdata2 = new ArrayList<String>();
int i = 1;
System.out.println("");
System.out.println("TEST PATHS");
System.out.println("");
System.out.println("All Possible Test Paths...");
System.out.println("");

for (int count = 0; count < name.getRowCount(); count++){
if(name.getValueAt(count,0).equals("Relation")&&name.getValueAt(count,1).toString(
).contains("lt=->>")){ //lt=->> is syncMsg
String[] parts = name.getValueAt(count, 1).toString().split(":");
String parts1 = parts[0]; //Desired Message Part
//String parts2 = parts[1]; //Undesired Message Part
numdata.add(parts1);}
if(name.getValueAt(count,0).equals("Relation")&&name.getValueAt(count,1).toString(
).contains("lt=<<-")){ //lt=->> is syncMsg
System.out.print(i +". ");
String[] string = name.getValueAt(count, 1).toString().split(":");
String string1 = string[0]; //Desired Message Part
//String string2 = string[1]; //Undesired Message Part
numdata.add(string1);
numdata2.add(string1);
string1 = numdata.toString().replace(", ", " - ");
System.out.println(string1);
i++;}
if(name.getValueAt(count,0).equals("Relation")&&name.getValueAt(count,1).toString(
).contains("lt=<.")){ //lt=<. is return message
System.out.print(i +". ");
String[] part = name.getValueAt(count, 1).toString().split(":");
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
numdata.add(par1);
numdata2.add(par1);
if( ! name.getValueAt(count, 1).toString().contains(":")){
numdata.remove(par1);
numdata2.remove(par1);}
par1 = numdata.toString().replace(", ", " - ");
System.out.println(par1);
i++;}
if(name.getValueAt(count,0).equals("Relation")&&name.getValueAt(count,1).toString(
).contains("lt=.>")){ //lt=.> is return message
System.out.print(i +". ");
String[] part_no = name.getValueAt(count, 1).toString().split(":");
String part_no1 = part_no[0]; //Desired Message Part
//String part_no2 = part_no[1]; //Undesired Message Part
numdata.add(part_no1);
numdata2.add(part_no1);
if( ! name.getValueAt(count, 1).toString().contains(":")){
numdata.remove(part_no1);
numdata2.remove(part_no1);}
part_no1 = numdata.toString().replace(", ", " - ");
System.out.println(part_no1);
i++;}
}
String lastElement_numdata=null;
lastElement_numdata=numdata.get(numdata.size()-1); //last message in numdata list.
if(numdata2.contains(lastElement_numdata)){
System.out.println("");}
else{
String last_list = numdata.toString().replace(", ", " - ");
System.out.println(i + ". " + last_list);}
}
```

```
public void testAllPaths() { //This function reads messages of all possible path
in the sequence diagram.
Table name = new Table();
List<String> listdata = new ArrayList<String>();
List<String> listdata2 = new ArrayList<String>();
int i=1;
int best_path_no=0;
System.out.println("");
for (int count = 0; count < name.getRowCount(); count++){
if(name.getValueAt(count,0).equals("Relation")&&name.getValueAt(count,1).toString(
).contains("lt=->>")){
String[] parts = name.getValueAt(count, 1).toString().split("lt=->>");
String parts1 = parts[0]; //Desired Message Part
//String parts2 = parts[1]; //Undesired Message Part
listdata.add(parts1);}
if(name.getValueAt(count,0).equals("Relation")&&name.getValueAt(count,1).toString(
).contains("lt=<<-")){
System.out.print(i +". ");
String[] parts = name.getValueAt(count, 1).toString().split("lt=<<-");
String parts1 = parts[0]; //Desired Message Part
//String parts2 = parts[1]; //Undesired Message Part
listdata.add(parts1);
listdata2.add(parts1);
parts1 = listdata.toString().replace(", ", " - ");
System.out.println(parts1);
i++;
best_path_no++;}
if( name.getValueAt(count, 1).toString().contains("lt=<.")){
System.out.print(i +". ");
String[] part = name.getValueAt(count, 1).toString().split("lt=<.");
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
listdata.add(par1);
listdata2.add(par1);
par1 = listdata.toString().replace(", ", " - ");
System.out.println(par1);
i++;
best_path_no++;}
if( name.getValueAt(count, 1).toString().contains("lt=.>")){
System.out.print(i +". ");
String[] part = name.getValueAt(count, 1).toString().split("lt=.>");
String par1 = part[0]; //Desired Message Part
//String par2 = part[1]; //Undesired Message Part
listdata.add(par1);
listdata2.add(par1);
par1 = listdata.toString().replace(", ", " - ");
System.out.println(par1);
i++;
best_path_no++;}}
String lastElement_numdata=null;
lastElement_numdata=listdata.get(listdata.size()-1);
if(listdata2.contains(lastElement_numdata)){
System.out.println("");}
else{
String last_list = listdata.toString().replace(", ", " - ");
System.out.println(i + ". " + last_list);
best_path_no++;}
String best_list = listdata.toString().replace(", ", " - ");
System.out.println("BEST PATH");
System.out.println("");
System.out.println("The Best Path:" + best_path_no + ". " + " Path " + best_list);
System.out.println("");
}
public void testNames() {
Table data = new Table();
int index = 0;
assertNotNull(data.getColumnName(index));
}
public void testCount() {
Table table = new Table();
int columnIndex = 0;
int rowIndex = 0;
assertNotNull(table.getValueAt(rowIndex, columnIndex));
assertNotNull(table.getColumnCount());
assertNotNull(table.getRowCount());
}
```

```
public void testReadMessages() { //This function reads all messages of the
sequence diagram.
Table message = new Table();
List<String> data = new ArrayList<String>();
List<String> data2 = new ArrayList<String>();
String string=null;
for (int count = 0; count < message.getRowCount(); count++){
if(message.getValueAt(count, 0).equals("Relation") && message.getValueAt(count,
1).toString().contains("lt=->>")){
String[] list = message.getValueAt(count, 1).toString().split("lt=->>//");
String list1 = list[0]; //Desired Message Part
String list2 = list[1]; //Undesired Message Part
data.add(list1);
string = data.toString().replace(", ", " -> ");
data2.add(list1 + "->" + list2);
}
if(message.getValueAt(count, 0).equals("Relation") && message.getValueAt(count,
1).toString().contains("lt=<<-")){
String[] list1 = message.getValueAt(count, 1).toString().split("lt=<<-//");
String list_1 = list1[0]; //Desired Message Part
String list_2 = list1[1]; //Undesired Message Part
data.add(list_1);
string = data.toString().replace(", ", " -> ");
data2.add(list_1 + "->" + list_2);}
if(message.getValueAt(count, 1).toString().contains("lt=<.")){
String[] start = message.getValueAt(count, 1).toString().split("lt=<.");
String start1 = start[0]; //Desired Message Part
//String start2 = start[1]; //Undesired Message Part
data.add(start1);
string = data.toString().replace(", ", " - ");
data2.add(start1 + "->returnMsg");}
if(message.getValueAt(count, 1).toString().contains("lt=.>")){
String[] start = message.getValueAt(count, 1).toString().split("lt=.>");
String start1 = start[0]; //Desired Message Part
//String start2 = start[1]; //Undesired Message Part
data.add(start1);
string = data.toString().replace(", ", " - ");
data2.add(start1 + "->returnMsg");}}
System.out.println("ALL MESSAGES");
System.out.println("");
System.out.println("All messages of sequence diagram: " + string);
System.out.println("");
System.out.println("Total message number of sequence diagram: " + data.size());
System.out.println("");
System.out.println("MESSAGES RELATIONS");
System.out.println("");
System.out.println("All messages relations of sequence diagram: " + data2);
System.out.println("");
}
public void testNameofDiagram() { //Return the sequence diagram frames.
Table object = new Table();
List<String> nameofdiagram = new ArrayList<String>();
for (int count = 0; count < object.getRowCount(); count++){
if(object.getValueAt(count, 0).equals("UMLFrame")){
nameofdiagram.add(object.getValueAt(count, 1).toString());}}
System.out.println("INTERACTION FRAMES OF DIAGRAM");
System.out.println("");
System.out.println("Interaction Frames names of diagram: " + nameofdiagram);
}
public void testObjects() { //Return the sequence diagram objects.
Table object = new Table();
int total_number=0;
List<String> number = new ArrayList<String>();
for (int count = 0; count < object.getRowCount(); count++){
if(object.getValueAt(count, 0).equals("UMLGeneric")){
number.add(object.getValueAt(count, 1).toString());
total_number++;}
}
System.out.println("OBJECTS");
System.out.println("");
System.out.println("All objects of sequence diagram: " + number);
System.out.println("");
System.out.println("Total objects number of sequence diagram: " + total_number);}
}
```