International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Issue 8, Volume 2 (August 2015) www.ijirae.com

Functional Test Cases Generation Based on Automated Generated Use Case Diagram

Arjinder Singh
Chandigarh University

Er. Sumit Sharma
Chandigarh University

Abstract— Software testing is used to test the developed system to find the errors and defects in the software. To find the errors and defects in the software system test cases are generated. Test cases generation based on three parts i.e. coding, design and specification. Specification based testing deals with the generation of test cases from the functional requirements of Software Requirement Specification. This approach deals with the automated generation of functional test cases. This approach proposes a framework for the automated generation of use case diagram. Use cases of automated generated use case diagram are used to develop one or more use case activity diagrams. By using depth first search technique, functional test cases from the use case activity diagrams are generated. Automated generation of functional test cases and use case diagrams saves the maximum effort and time consumed by software tester to test the software system.

Keywords— SRS, UAD, UADT, UADG, DFS

I. INTRODUCTION

Decades of work by researchers and practitioners in the field of numerous software quality assurance techniques, testing remains one of the most widely practiced and studied approaches for assessing and improving software quality [1].

Testing is one of the salient phases of the software developing model. Testing phase comes after the coding phase. The major purpose of testing the software is to find and correct the errors to ensure the error free and reliable delivery of the software to the customer or client. In the current era of software development, testing considered as a very crucial phase of the development life cycle, so a separate testing department has been made in the organization which is completely different from the other development department. For testing the software, software testing teams have numerous roles to play.

Every developed software project has its target audience. While developing the software it is clearly kept in mind about the type of audience for whom the software is going to be developed. There can be large number of types of targeted audience so while developing the system it is necessary to know clearly about the audience. For example, the audience of merchant navy system is completely different from a gaming system. And audience of banking system is completely different from university system. Therefore it is necessary for the developing organizations to clearly know about that customer will accept the software or not.

II. RELATED WORK

Nebut and Clementine [2] presented the approach of test cases generation directly from the software requirement specification. This approach is mainly divided into two main processes. First process is to generate use cases from the requirement and second process is to create test cases from the generate use cases of the system. This approach assumed that the formalization of requirements has to be taken out manually. For the evaluation of this approach author has considered three case studies generated by this proposed approach.

Schweighofer et al. [3] presented an overview about different UML diagrams that can be used for the creation of test cases. This paper provided an overview of use case diagram, activity diagram and state chart diagram that can be used to create the test cases. This paper also presented the lack of automation issue in the generation of test cases. This paper also presented the techniques that are used by these diagrams to generate test cases like trees graphs, tables and genetic algorithm.

Offutt et al. [4] presented the approach for creation of test cases from specification by using UML state chart diagram. This approach provided automation of test generation by embedding with rational rose tool. The major drawback of this paper is it only considers the state chart diagram for test case generation. This approach is limited to just one single program and a single set of test data.

Gemino et al. [5] addressed the issues of using use cases with or without incorporating use case diagrams. This paper described the effectiveness of using use case diagrams rather than simply using only use cases. This paper used cognitive theory of multimedia learning that helps in improving the impact of use cases. This paper showed that the user which uses use case concept with the use case diagram becomes more capable of understanding the system rather than those who just used the only use cases.

Hadar et al. [6] presented a survey which states the importance of different UML diagram for the generation of test cases. This approach works in two groups, first group consists of senior students who were asked to check the method that how they retrieve information from different UML diagrams. And second group consists of another senior who were asked to fill the questionnaire in which they have to rank the UML diagrams according to their importance. This paper concluded up that each UML diagrams are equally important. Muhairat et al. [7] presented an approach for generation of use case diagram based on event table which is built by considering the software requirement specification.

This paper addressed the time consuming issue of developing use case diagram from the requirements. This approach is completely depended on the event table for generation of use case diagrams. The limitation of this approach is the lack of automation capability of generating use case diagram. Khan et al. [8] presented a comparative study between three different techniques used in testing. The paper presented that each technique is effective if carefully chosen. This paper provided a comparison table based on which we can opt our testing strategy which satisfies our problem. Prasanna [9] presented a survey of various techniques available for the generation of test data automatically. This paper also addressed the problems associated with each technique. This paper proposed the model based testing strategy as an optimum one for the creation of test cases among the various other techniques available. Nanda et al. [10] presented strategy for the creation of test cases from the activity diagram. This paper addressed the issue of handling parallel activities with the help of fork-join pairs. This approach provided major emphasis on the reuse of design models which results in low cost for developing the test cases. This approach also presented the defects analysis scenarios for findings the defects during the analysis of models. Jena et al. [11] presented an approach for the creation of test cases by using genetic algorithms. This paper also addressed the issue of detection of faults in the early stages of software development that results in reducing time, cost and effort takes in development of any software system. This paper proposed an approach of generation of activity flow table from activity diagram and then generation of activity flow graph by using activity coverage criteria.

III. PROPOSED METHODOLOGY

This methodology discusses the steps used to generate functional test cases from the software requirement specification. Numbers of steps are involved in the generation of test cases as shown in fig1.

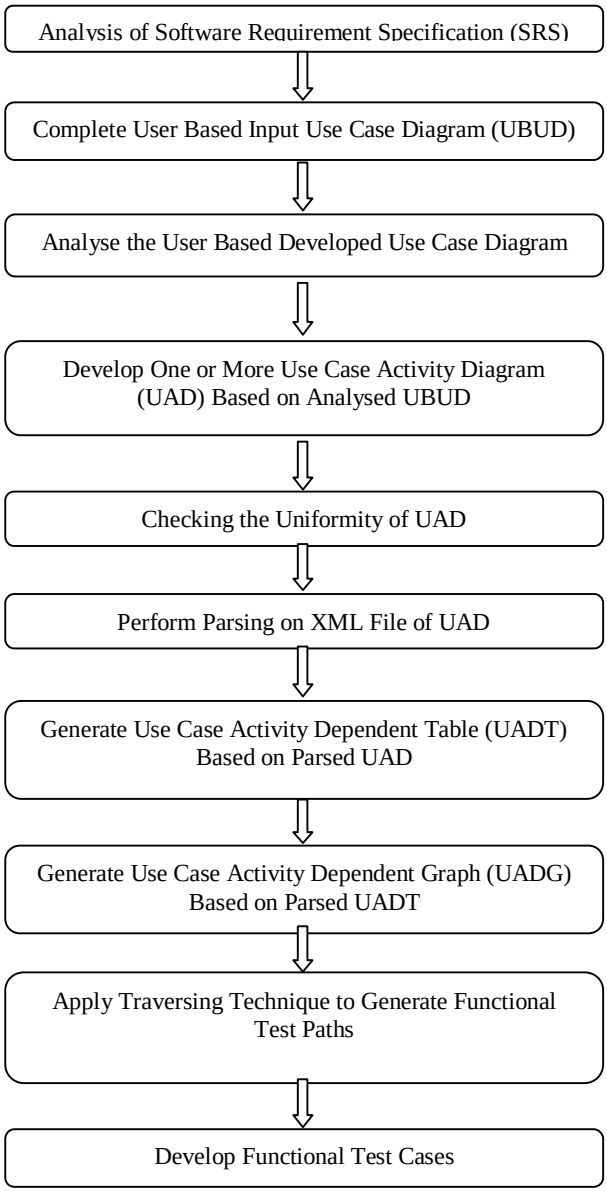

Fig. 1 Methodology for Functional Test Cases Generation

A. Analysis of Software Requirement Specification

In manual analysis of SRS, the analyst uses their expertise skills to identify use cases and actors from the functional requirements of the SRS. Analyst thoroughly read the functional requirements and identifies appropriate use cases and their corresponding actors. Each actor interacts with systems by having association with their corresponding functionalities. For the generation of functional test cases this research will use the SRS document of ATM system. After analysis of SRS document

B. Complete User Based Input Use Case Diagram (UBUD)

After the analysis of SRS document, the next step is to enter the name of identified actors and use cases in the proposed framework of this research. This framework will automatically generate the XML file of use case diagram based on input names.

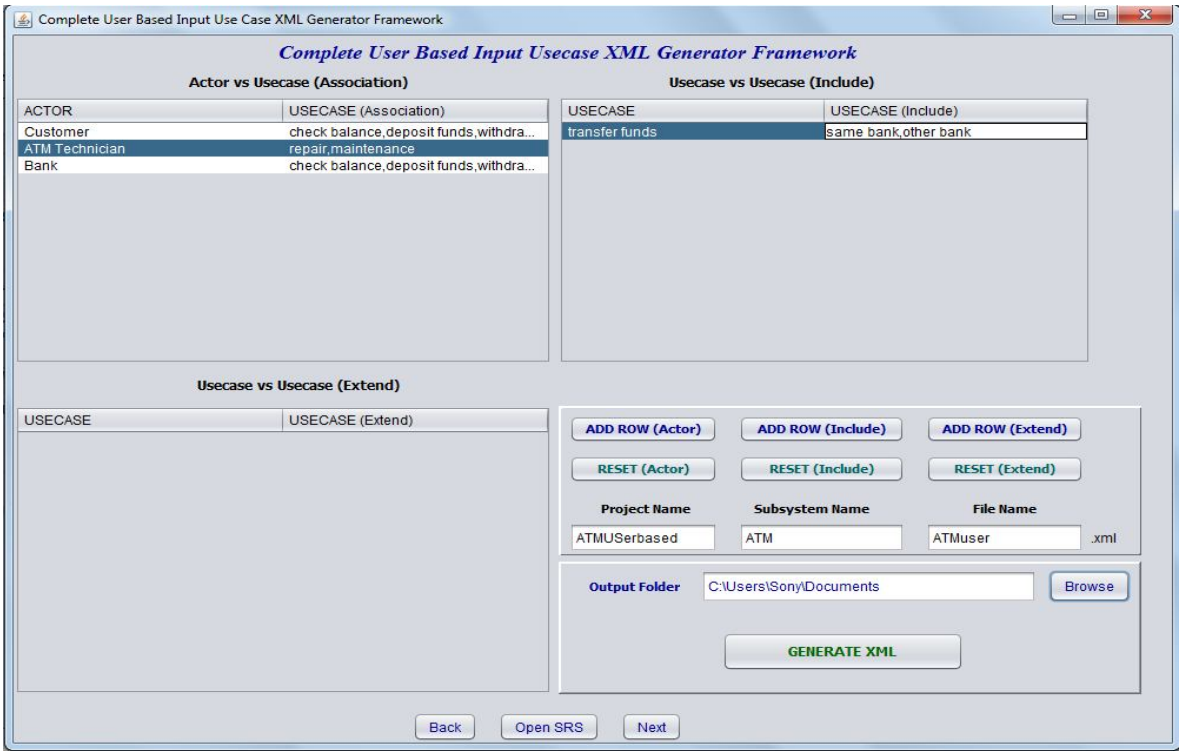

Fig.2 Framework for Use Case XML Generator

After filling the data in the proposed framework for the generation of XML file of use case diagram, open the XML file of use case diagram with the help of Magic Draw UML tool.

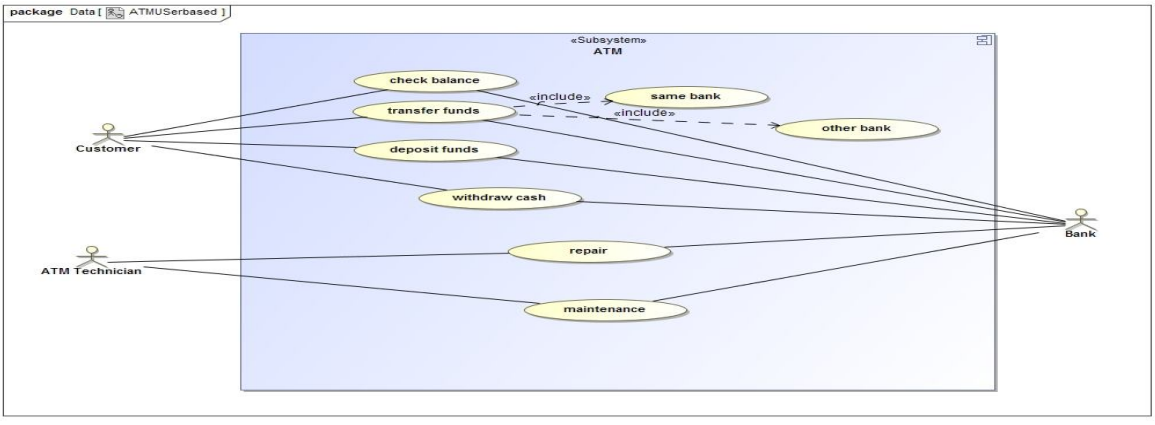

Fig. 3 Generated Use Case Diagram with Proposed Framework

C. Analyse the User Based Developed Use Case Diagram

In this step use case diagram generated is analyzed to get ensured about the coverage of all functional requirements specified in the SRS. If any requirement is left out in the user based input generation of use case diagram, that use case diagram is updated with the new information or data.

D. Develop One or More Use Case Activity Diagram (UAD) Based on UBUD

In this step use case activity diagrams are generated based on proposed framework of use case XML generator. There is possibility of number of use case activity diagram generation depends on the number of use cases available in the system. UAD corresponding to each use case can be drawn. Activity diagrams are important for the generation of test cases efficiently. Use case diagram for ATM balance enquiry module is drawn to further generate functional test cases. UAD is developed with the help of magic draw UML tool.

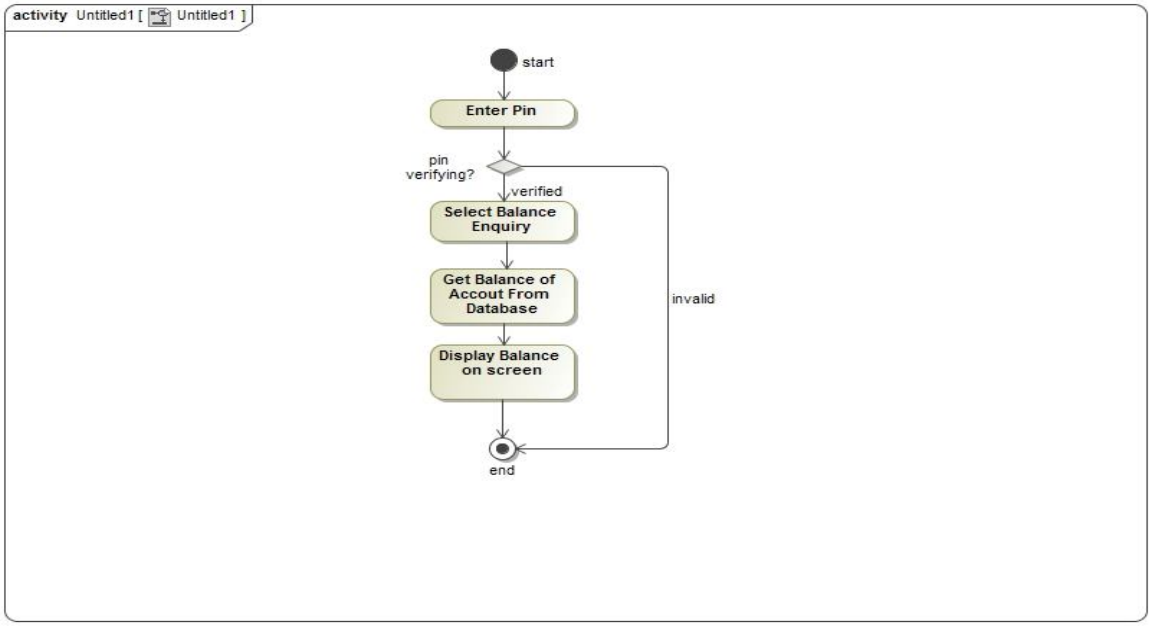

Fig.4 Use Case Activity Diagram of Balance Enquiry

After the generation of use case activity diagram, the next step is to save the diagram in the form of XML file.

Fig.5 XML File of Use Case Activity Diagram of Balance Enquiry

E. Perform Parsing on XML File of UAD

After the conversion of activity diagram into XML file, the next step is to perform DOM parsing on the XML file. With the help of DOM parsing various fields of activity diagrams are identified like name of the node, in degree and out degree of nodes, dependency and dependent nodes.

F. Generate Use Case Activity Dependent Table (UADT) Based on Parsed Data

The parsed data of XML file are used to generate UADT. The use case activity dependent table includes the various fields which are further helpful in generating graph for the UAD.

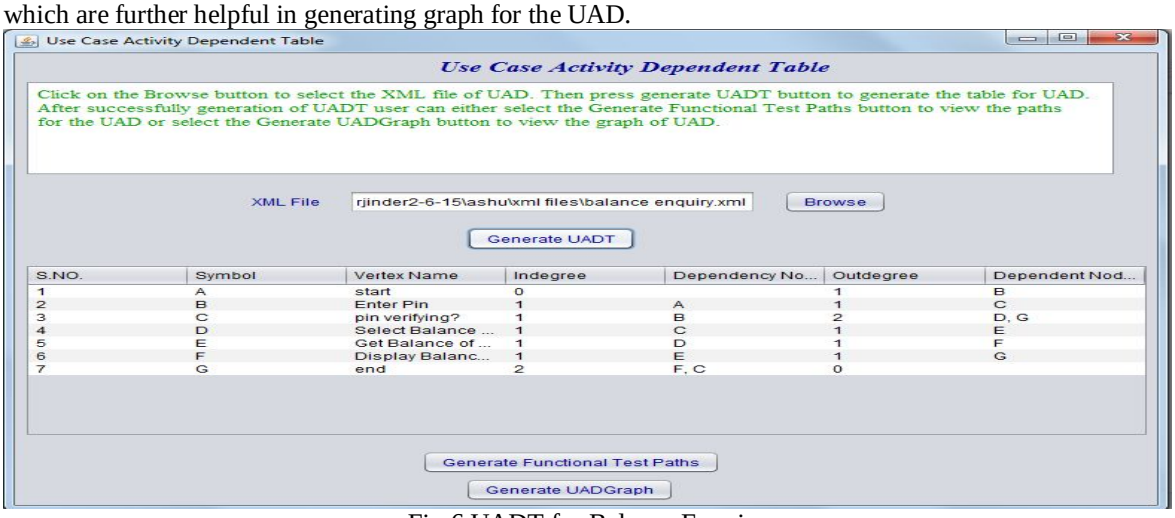

Fig.6 UADT for Balance Enquiry

G. Generate Use Case Activity Dependent Graph (UADG) Based on UADT

A graph for use case activity diagram is drawn with the data generated for the UADT. For the generation of functional test cases graph is the important element. Traversing techniques can be easily applied on the graph.

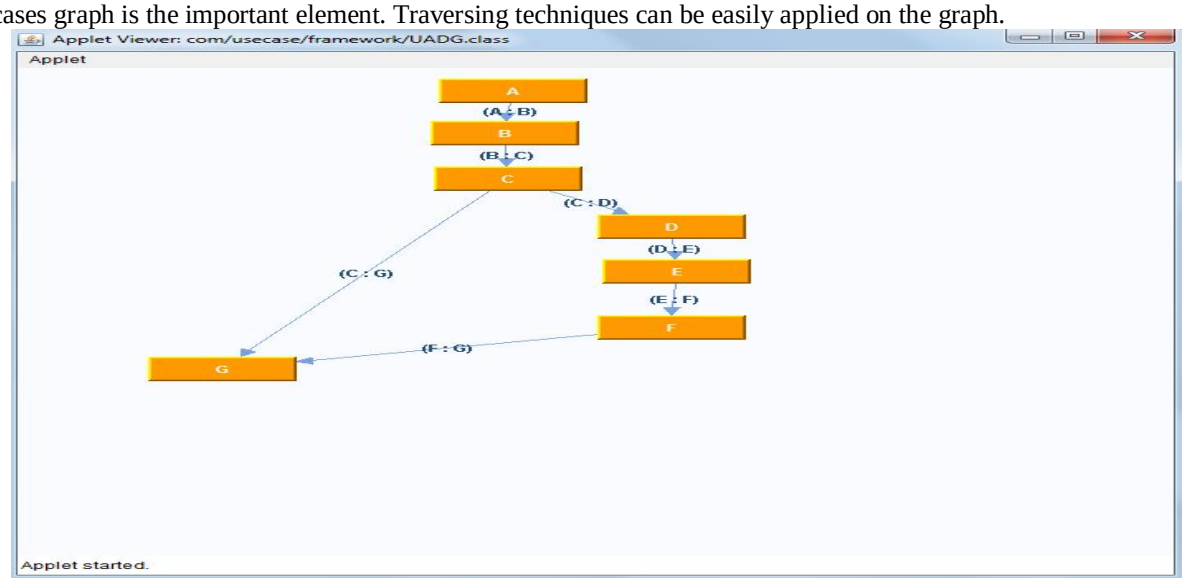

Fig.7 UADG for Balance Enquiry

H. Apply Traversing Technique to Generate Functional Test Paths

Depth first search technique is applied to traverse the graph in order to find all the possible paths available in the use case activity dependent graph

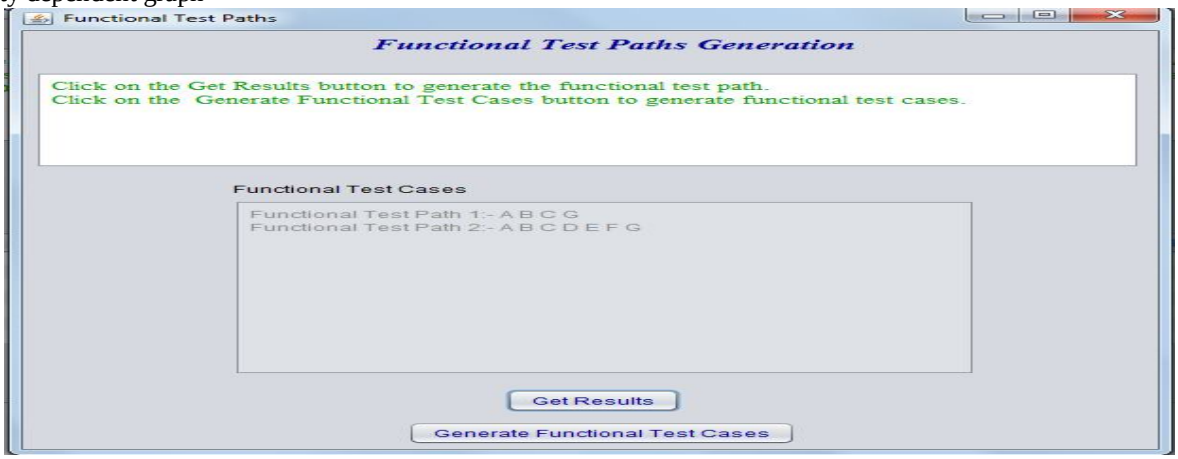

Fig.8 Functional Test Paths for Balance Enquiry

I. Develop Functional Test Cases

After the generation of functional test paths, the functional test cases are generated. Test case template to be followed includes the input condition and output condition of the path.

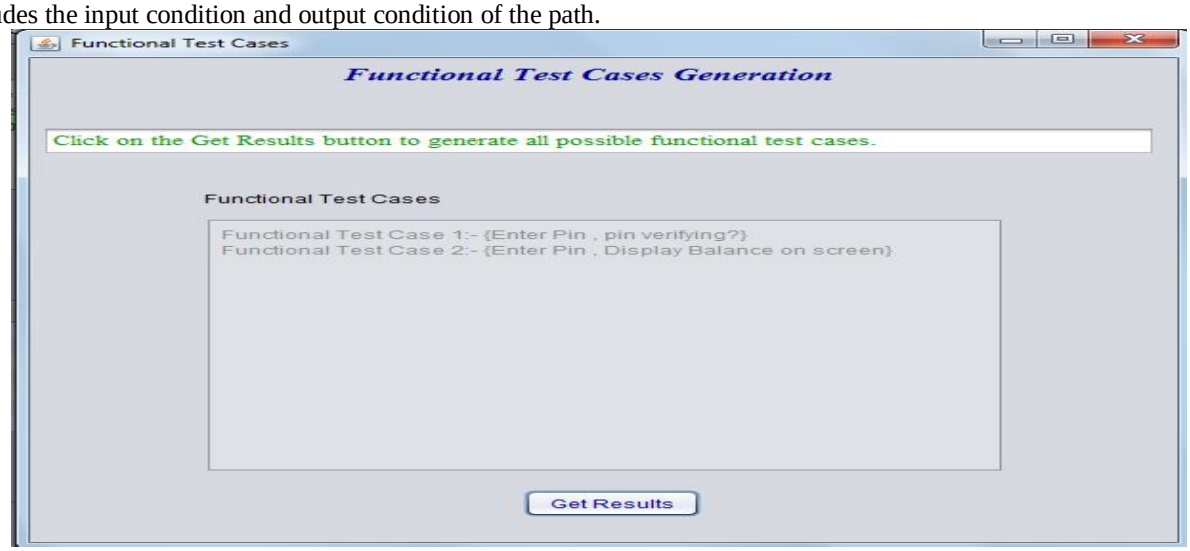

Fig.9 Functional Test Cases for Balance Enquiry

Similarly with the help of this technique functional test cases for other use cases are also developed. Functional test cases for the number of SRS documents can be developed with this approach. SRS should be written in IEEE 830 format.

IV. CONCLUSIONS

Functional test cases from software requirement specification are generated by developing use case diagram and activity diagram. Automation approach for the generation of use case diagram and generation of functional test cases has been introduced. Automation results helpful in saving maximum effort consumed in drawing of use case diagram manually with the help of any UML tool or by using pen or paper. Automation also results in saving cost included in the testing of the software application. Mostly in code based testing, we are able to find errors in coding or logics, we are not able to ensure the covering of all requirements specified in the SRS. Our automation approach ensures the maximum coverage of all requirements specified in the SRS. Hence possibility of reliable system delivery to the customer increases.

REFERENCES

[1] Alessandro Orso and Gregg Rothermel, "Software Testing: A Research Travelogue (2000–2014)", ACM (2014).
[2] Nebut, Clementine, "Automatic test generation: A use case driven approach." Software Engineering, IEEE Transactions on 32.3 (2006)
[3] SCHWEIGHOFER, TINA, and MARJAN HERIČKO. "Approaches for Test Case Generation from UML Diagrams ." Third Workshop on Software Quality Analysis, Monitoring, Improvement and Applications SQAMIA (2014).
[4] Offutt, Jeff, and AynurAbdurazik. "Generating tests from UML specifications."«UML»'99—The Unified Modeling Language. Springer Berlin Heidelberg,. 416-429. (1999)
[5] Gemino, Andrew, and Drew Parker. "Use case diagrams in support of use case modeling: Deriving understanding from the picture." Journal of Database Management (JDM) 20.1 1-24. (2009)
[6] Hadar, Irit, and Orit Hazzan. "On the contribution of UML diagrams to software system comprehension." Journal of Object Technology 3.1 143-156. (2004)
[7] Muhairat, Mohammad I., and Rafa E. Al-Qutaish. "An approach to derive the use case diagrams from an event table." Proceedings of the 8th WSEAS International Conference on Software engineering, parallel and distributed systems. World Scientific and Engineering Academy and Society (WSEAS), (2009).
[8] Khan, MohdEhmer, and Farmeena Khan. "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques." International Journal of Advanced Computer Science and Applications (IJACSA) 3.6 (2012).
[9] Prasanna "A survey on automatic test case generation." Academic Open Internet Journal 15.part 6 (2005).
[10] Nanda, P., Dr DP Mohapatra, and S. K. Swain. "Generation of Test Scenarios Using Activity Diagram." Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India. Vol. 4. (2008).
[11] Jena, Ajay Kumar, Santosh Kumar Swain, and Durga Prasad Mohapatra. "A novel approach for test case generation from UML activity diagram." Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on. IEEE, (2014).

