TEST CASE GENERATION USING USER STORIES

Talal Bin Afzal

Enrollment No: 01-241172-040

Supervisor: Dr Tamim Ahmed Khan

A thesis submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the requirements of a Master degree in Software Engineering

June 2020

Abstract

Agile software development methodology is being used widely for software development with the passage of time because of its quick delivery of projects to the market. In the process of software development, software testing plays a vital role as it is used to validate the software product and to make sure that the required system functionalities are available or not. It is also performed to identify the bugs or errors in the system and to ensure the proper functioning of the system. Several tools, practices and methodologies are existed to generate test cases. All these methodologies are generating the test cases based on specifications or models. So, there’s a need of a technique which can generate test cases from user stories instead of specifications based or model based. In this research a test case generation technique and a coverage metric is proposed which generates test cases from user stories. In the first step user story components i.e. user roles, goals and reasons are identified from a user story. After identifying the user story components, tasks are also generated from the user stories. On the basis of these tasks and components test cases are generated. These components and tasks also help to map test cases to their respective user story. A coverage metric is also proposed in this research which helps to calculate the coverage of the user stories based on the test cases generated. The proposed test case generation technique is implemented on four moblie applications to generate the test cases. Test cases generated for these applications based on the proposed technique covers all the user stories and are useful to identify the bugs and issues in the applications.

Dedication

I dedicate my research thesis to my parents, my supervisor and Aneela Idrees who always remember me in their prayers, to my respected teachers and to my respected colleagues who always helped me to make this possible.

Acknowledgments

I am very grateful to ALMIGHTY ALLAH the most courteous and generous for providing me the opportunity and knowledge in concluding this thesis. Without His generous support, I would have never done this.

I would also like to pay my regards to my supervisor ***Dr. Tamim Ahmed Khan,*** *for his support and supervision. His experience and ability to make things understand in a unique way have increased my knowledge and helped me clear my ambiguities. We had various meetings over the making of the research proposal and the model about its implementations. He assisted me in every phase of my research development. I would also like to acknowledge* ***Dr. Huma Hayyat*** *and* ***Mr. Mohsin Munir*** *and* ***Ms. Aleena Ejaz*** *for their assistance and support throughout this thesis.*

I would like to acknowledge the Department of Software Engineering at Bahria University Islamabad. My post-graduate friend who supported me during my post-graduation study.

Contents

List of Figures

List of tables

Chapter 1

INTRODUCTION

There are more than 200 giant software organizations that have embraced agile for software development [1]. Quick delivery of the product to the market, managing fresh requirements in an efficient way and surge in development teams productivity are the significant inspirations for the acceptance of agile development as a software development methodology [1]. Agile methodology is based on the approaches like iterations and increments which is a result of a combined teamwork[2].

Scrum is a widely used method among all the agile methodologies. Scrum deals with the methodology of management and is used to maintain and improve an existing system or prototype [3]. In scrum, a project is distributed into different iterations known as “sprints” [4]. Each sprint implements a feature which is mentioned in a user story (U/St.), which is basically a requirement written in a simple language [5]. Each user story is distributed into different tasks, which can be implemented within 1 or 2 days. Exit criteria for tasks and user story is defined by the team [5]. Scrum focuses on the implementation instead of the documentation [4].

In agile development the software team gets the requirements in the form of user stories [1]. User story is an elite definition of prerequisite, which is quite helpful to estimate a reasonable amount of effort which is required to implement that user story [1].

Developers implement the functionality mentioned in the user story and also write their respective unit and integration test cases. Testers also write the required test scripts independently which describes the functionality usage in a step-by-step manner to highlight the desired system behaviour [6].

With the passage of time the size and complexity of software systems grow which results in an increased effort for testing a software. Almost 50% cost of the software development is devoted to the software testing [7]. Test case development is one of the most vital issues as designing effective test cases and their execution takes a lot of time and labour [8]. Test cases can be generated either by using the requirement specifications or by using the code [9].

1.1. Motivation

There are various tools and approaches for generating test cases but they drive them from UML diagrams, Use Case specifications and Model based specifications [8], [10]–[13]. So a technique is compulsory which assists to generate test cases from user stories and map user stories to the corresponding test cases to provide a widespread coverage. In this research our aim is to generate test cases from user stories instead of use cases and UML diagrams and to provide a complete coverage of user stories and test cases on the basis of a coverage metric.

1.2. Problem statement or research questions

There are a number of tools and approaches which create test cases based on UML diagrams and related artefacts such as, use case specifications and model-based specifications. Since the industry is moving towards agile methods where user stories are a widely accepted requirements specification technique, we require a test case generation technique that generates test cases from user stories accompanied with a coverage metric.

1.3. The aims and objectives

Our research objective is to propose such a technique which will be helpful to

1. Generate test cases from user stories.
2. Provides a complete coverage of user stories and test cases based on a coverage metric.

1.4. Main contributions

The major offerings of this research are as follows:

1. It provides a model which is quite helpful for generating test cases from user stories.
2. Provides a coverage metric which makes sure that all the user stories and their test cases are implemented.

1.5. Thesis organisation

Chapter 2 discusses the available work related to the test case generation techniques. It reflects the different methodologies related to the specification based and model based testing. It also discusses the traditional software development approaches and agile development in context of scrum and user story. It also involves different user story templates and the research design of our thesis.

Chapter 3 discusses the proposed methodology in which test cases are generated from the user stories based on the tasks and user story components. It also involves a coverage metric for test cases.

Chapter 4 evaluates and validates the proposed technique by implementing it to generate test cases from user stories for a an application named.

Chapter 5 concludes the thesis based on the work done and explains the future work

Chapter 2

LITERATURE REVIEW

2.1.Testing

Software testing is very helpful for detecting errors and bugs in a software [14]. Testing is basically a way to validate the software system whether it works as expected or not. Test cases are implemented to check that the system which is to be tested is working as specified in the requirement documents or not [15]. Testing is quite challenging and hardworking practice in the absence of an active automation. It is estimated that testing consumes almost 50% of the efforts required to develop a software [15]. Software testing has a limitation that it is able to show the errors and bugs in a software system but it can never validate the absence of errors, in spite of this limitation software testing has always been an important technique to validate the system under test [16].

2.1.1. Specification based testing

The methods used for software testing are solely dependent on the application of information which is used to select test data while specifications are accountable for valuable information for testing [17]. The proposed behaviour is overlooked completely as the techniques of testing which are based on the implementation emphasize on the real-time performance of the implementation [17]. Based on a requirements specification, this case selection technique consists of test cases selected by the user. As too often in any case, when the specification is informal, the whole thing can be done efficiently [17].

Test case selection should purely base on source code which has long been recognised [18]. Gourlay has represented a mathematical framework and he acknowledged the need for specification-based testing [19]. A well written and well-defined semantics shall be used to attain results from testing which is based on specification and the semantics must be written in an official language to obtain proper results. Informal specifications will not be able to expose errors; says Laski [20].

To divide the domain which is used for input data into equivalence classes and to select the data for testing from each of these classes the traditional functional testing method is used. Goodenough and Gerhart enhanced this over-all method. A table known as a condition table is used as it contains several information sources, where columns signifies the test case which is to be derived, which in turn is a blend of conditions which are to be tested [18].

A technique is provided by Ostrand and Balcer for category-partition in which the person performing the test examines specification and classifies distinct functional units which are testable, inputs of each function are labelled and then it is classified into different categories as per their equivalence [21]. Test case selection approach involves activities related to the documentation interpretation by the testers.

Scholars recommended different practises which concentrates on the specification to develop the test cases. It was also suggested that the path domains can be subdivided to construct the subdomains which are likely to be error centred, which may be based on the specifications [22].

Partition analysis method was devised by Richardson and Clarke which progresses a division by overlaying a partition based on implementation and a specification [23], [24]. Functional testing hires functional decomposition information of design and specification and applies its guidelines to adopt the test cases for several functional classes [25]. From algebraic specifications, an approach is introduced which is used to select the description of categorically enhancing collection of test cases [26], [27].

Constructed on the basis of modification of a predicate calculus specification, a test adequacy technique was proposed by Gopal and Budd [28]. But the mentioned selection of test case tactics has never been defined appropriately which can be applicable in general. The main focus is on testing the specifications rather than their implementation. Goguen and Tardo is in favour of testing the algebraic specifications [29]. Neither of these techniques are focused on the selection of real test cases for the specification.

A tool set used to provide writing capabilities and declarations which are assembled into run-time checks for debugging is called The Anna tool set [30]. To identify variations in the code Velasco presented a technique that uses assertions supplied by the programmer, which in turn used to choose data for testing [31].

2.1.2. Model based testing

Testing which is based on model, initiates with a model that is considered to be correct and usable. This model depicts exactly how the system under test should behave and how it shouldn’t behave. Therefore it forms the basis for generating the test cases systematically [15]. It can be difficult to build a model because of some factors including system's complexity, designers and specifiers disability to make a model, involvement of some third party tool or component, incomplete or missing documentation [15].

Model based testing is of various types depending upon the several model's requirements like type of model required, quality attributes needed to be tested, formality level involved, and degree to which a system is being tested regarding it's observability and accessibility. Therefore model based testing is considered as recognised, active and black box testing [15]. Development based on models use Unified Modelling Language (UML) results in leading several researchers to implement UML diagrams for designing the test cases [32].

These model based techniques are used for testing, results in increasing the eminence and efficiency of the software as it has shifted the software testing techniques to a prior stage of the development and made test case generation independent of the design implementation [33]. Unit tests can be generated automatically by utilizing the MDA initiative [33].

On the Basis of sequence diagrams a method is introduced which uses a model that is independent of any specific platform to generate test cases. First of all, system behaviour is modelled with the help of sequence diagrams and it is than transformed into a testing model automatically for unit testing, based on model-to-model guidelines of alteration. Additional transformation rules are than applied on generated test case model to extract executable and concrete test cases [33].

There are multiple model-based approaches used for testing which are somehow different from each other expressively depending upon their design specifics, testing goals, support for tools and strategies for evaluation [33]–[36]. Yuan introduced a method which create test cases automatically for web services [37]. Test case model is constructed on the basis of UML Testing Profile (UTP) [38] and Test Control Notation Version 3 (TTCN-3) [39]. The UTP standard applies the Model-Driven Architecture (MDA) approach to define a framework that is used to build brief test models which are used to generate the test cases automatically [40]. The test model that is produced can be customized to generate unit testing, system testing, component testing, efficiency testing, integration testing and performance testing [37].

Two automatic transformations were applied by Yuan to create executable test cases. In the initial step UTP and test case models assist to create Abstract Test Cases (ATC) and in the succeeding step the abstract test cases are transformed to generate the executable test cases [37].

A testing technique is presented in which a combination of UML and Object Constraint Language (OCL) is used to create the test cases automatically from the models [41].

Model-based regression testing approach is delivered in which the original model and the changed model is semantically compared, changes are detected and based on these modifications the original test suite is redefined and categorised as retestable, obsolete and reusable [42]. As the system changes test cases that can be tested again, should be executed for regression testing. Test cases related to the modules or elements which are no longer part of the system are invalid which are considered as obsolete test cases. Test cases related to the unaffected parts of the system are considered as reusable test cases which are not executed again during regression testing.

A technique is presented by Hartmann in which state machine and sequence diagrams are used to extract test cases to perform integration and unit testing. System level testing is done by extracting the test cases based on use cases and activity diagrams [43].

A business process model is defined by Trong that can be tested comprehensively and repetitively every time a change is made and is used to generate test cases that are executable. This approach contains three stages: to define the process which is to be tested, extracting nonfigurative test cases based on the under testing process, to produce the executable test cases from these nonfigurative test cases [44].

2.2. Established Software Development Methodologies

Right after the emergence of software development, many practices have been introduced to build a budget packed software which in future will help to cope up with user demands. [45], [46]. Agile methodology emerged as an alternative in 2001 [47]. The traditional software development methods are built on proper plans in general, the documentations for these methods and processes are generated from formal communication [48]–[51]. A predefined and broader plan is firmly followed to develop a new software in traditional development methods [52]. To use related information from the previous documents needs to be well defined at end of each stage so that they can be used at the next stage [49], [53]. Some of the basic methods are the Spiral Model [54], the Rapid Application Development (RAD) [55], or the Rational Unified Process (RUP) [56].

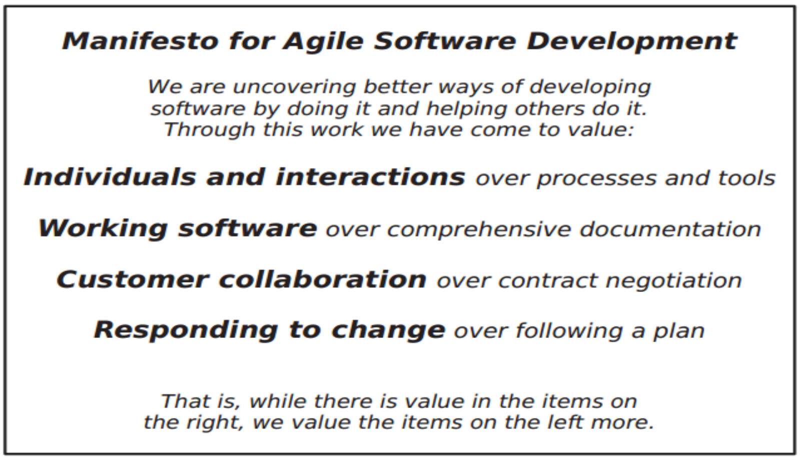
2.3. Agile

Numerous techniques have been offered which falls under the agile group, since the agile manifesto is introduced. Different perceptions regarding the meaning of agile or agility exists. However most of the agile methods are being used for software development [57].

The agile manifesto was designed as a result of a meeting which involved seventeen software developers who were the representatives of different agile methodologies [58]. The main objective was to argue on the applications of the approaches effective in developing software in late 90’s. This meeting was also focused on understanding the mutual grounds of each agile methodology.

Agile manifesto contains set of principles regarding software development using agile methods. It consists of twelve principles and four values [51].

The manifesto (fig. 2.1) contains the term “over” in the centre. Values on the right side are recognised by the agile software development as they are written in a regular style whereas the values on the left side are being emphasized more by the agile as they’re written in bold [59].



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions *over processes and tools*

Working software *over comprehensive documentation*

Customer collaboration *over contract negotiation*

Responding to change *over following a plan*

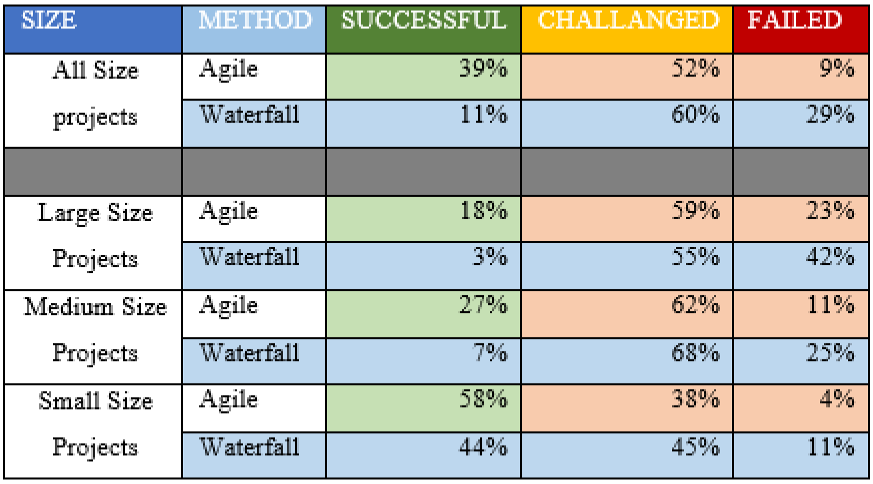
That is, while there is value in the items on the right, we value the items on the left more.

Figure. 2.1 Agile Manifesto values [60]

As it can be said by looking at the figure 2.1, that individuals and interactions are being focused than processes and tools. The success of a project generally depends more on the persons who are working on the project and their way of communication rather than on processes, methods and tools that are being used. In the agile practices developing a software is working is more essential than focusing on a broader document [61].

Agile methodologies consist of a series of a principles and processes created by experienced personals and being used for software development [62].

Because of their prominent success rate in software development, agile methods became quite famous once the creation of agile manifesto was done in 2001. As mentioned in the CHAOS report, the success rate of agile methodologies is more and the failing rate is less than the waterfall model. Figure 2.2, shows the report related to the project's success and failure based on using agile and waterfall in software development.



SIZE	METHOD	SUCCESSFUL	CHALLANGED	FAILED
All Size projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Figure. 2.2 The 2015 CHAOS report conducted by Sandish Group [63]

2.3.1. Scrum

As stated in a survey report of VersionOne conducted in 2016, Scrum is being used in 58% of the total agile projects which made it the most popular methodology of agile and XP has become unpopular over the time with a usage of 1% in agile development (Figure 2.3). XP has also been quite famous among agile methods as it covers several related engineering strategies for software development like User Stories, Planning Game, Pair Programming and Open Workspace. These practices are still famous and are used in several other agile methods [64].

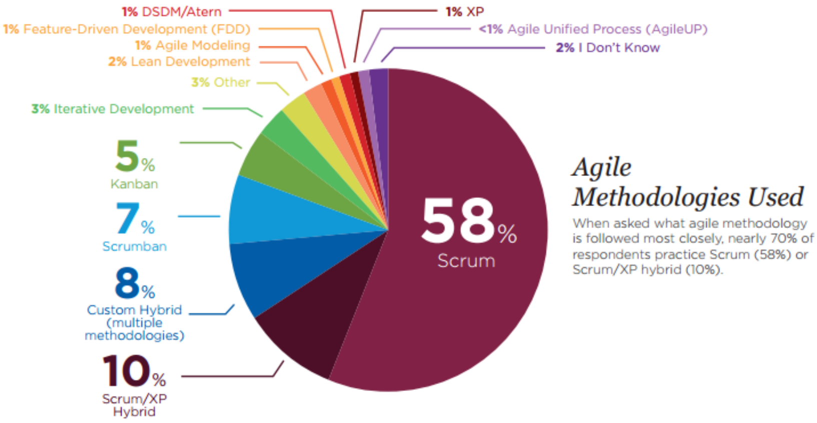


Figure. 2.3 Survey of most adapted agile methods [64]

Jeff Sutherland introduced the Scrum for software development and later on Mike Beedle and Ken Schwaber took interest in that in early 90's. This term "Scrum" is generated from a tactic used in the game of rugby [65].

In software development scrum is concentrated on the project organisation features [66]. It doesn't specify the software development methods to implement, it's up to the developers to use any technique or method for software development. Scrum can be used with XP and other agile methods and it works well [66].

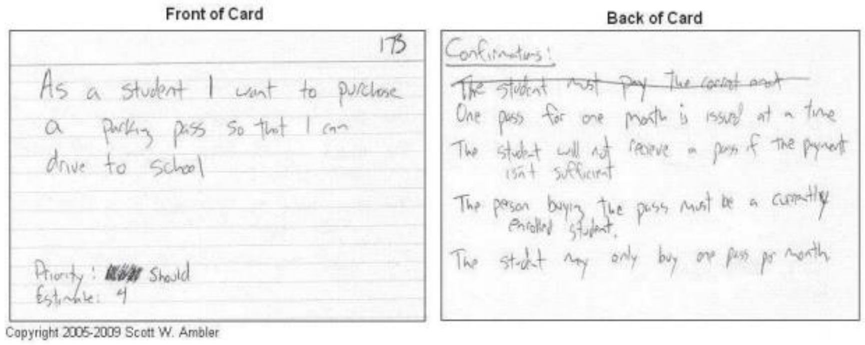
Scrum has three pillars which are transparency, inspection, and adaption [67]. Scrum has a common meaning of what a complete product is, which another important feature of scrum is [68]. Scrum basically works iteratively and incrementally and consists of relatively small iterations of 15 to 30 days which are known as "sprints". On the completion of every sprint a piece of practical software is produced.

Agile manifesto values inspired the authors to define the five values of scrum which was originally just consisted of practices only. According to Beedle and Schwaber the five values of scrum are courage, respect, commitment, openness and focus [69].

2.4.User Stories

User stories (U/St.) are the important requirement artefact in agile methodology. At first user stories have been offered by Kent Beck in eXtreme Programming (XP) [70]. Since than it has been embraced by other agile methods as well [59]. User story is considered as an element in an XP project as the progress is determined by the user story which is implemented on the basis of integrated and tested code [70]. User stories should be comprehensible and easy to understand for customers and developers. They should be testable, important to the customer and concise so that developers can build as many as possible user stories in a single iteration. User story is also descried as a small and simple narrative of functionality a system is expected to possess described from user's point of view [71].

In agile while writing the user stories, end-users are involved in that process as in End-User Development (EUD) which implies that the end-user is also a part of software development process [72]. A user story is considered to be best if it is concise and explains the important features of the system required by the user [70]. Conventionally user stories are used to be written on the index cards (see Figure 2.4) but nowadays tools such as JIRA 1, Excel, etc. are used to write user stories [73].



Front of Card

173

As a student I want to purchase a parking pass so that I can drive to school

Priority: Should
Estimate: 4

Back of Card

Confirmations:

~~The student must pay the correct amount~~
One pass for one month is issued at a time
The student will not receive a pass if the payment isn't sufficient
The person buying the pass must be a currently enrolled student.
The student may only buy one pass per month

Copyright 2005-2009 Scott W. Ambler

Figure. 2.4 User story index card [74]

There are some principles about how a user story should be written [70]. A user story should be clear, testable, independent, and not written by developers. Must be reasonable in size and it must include something that is important to the customer.

Cohn introduced a new way to write a user story as "A user can upload his/her resume". It can be seen that a user story is written from customer's perspective rather than the system. Cohn also suggested to involve "user role" as part of a user story [75]. By writing a user story in this way software developers have the feeling of satisfying the real user needs [75]. A user story can be categorized into two types named "positive and negative user story". Positive user story reveals only the positive meaning of a user story [76]. It represents the positive users and their positive actions involved in using the system.

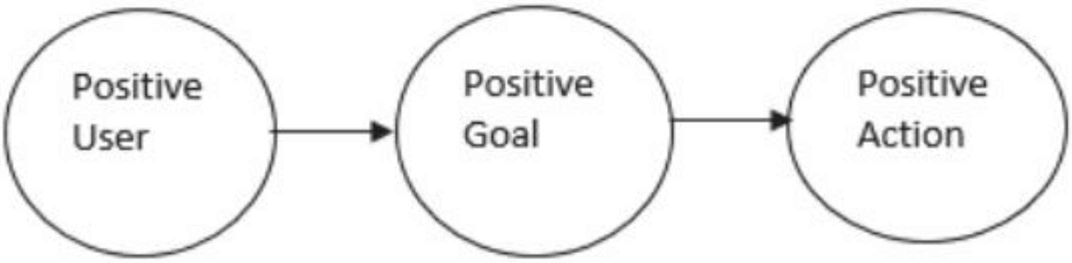


Figure. 2.5. Positive User Story Model [77]

Following are some of the positive user stories for an attendance monitoring system.

Sr. No.	Positive User stories
1	As an admin I want to login so that I can use the system
2	As an admin I want to add a faculty login so that the faculty record can be maintained in the system
3	As an admin I want to allocate courses to faculty members so that they can take the attendance

Table 2.1. Positive User Stories [77]

Negative user story is the one which reveals the negative meaning of a user story and identify the negative cases related to the user story from a point of view of security and safety of the system. It represents the positive user their negative actions and also the negative users their negative or positive actions involved [78].

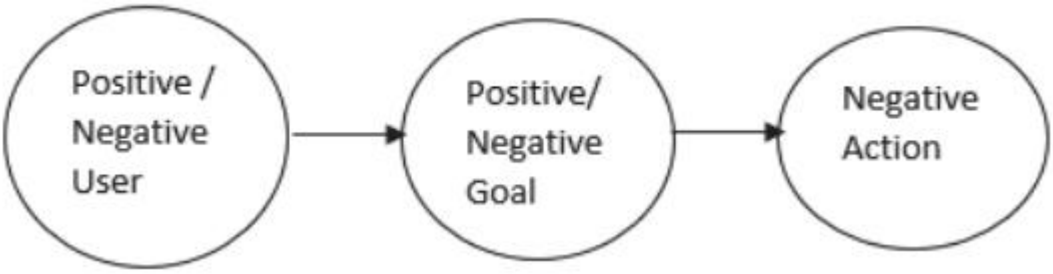


Figure. 2.6. Negative User Story Model [77]

Following are some of the negative user stories for an attendance monitoring system.

Sr. No.	Negative User stories
1	As a student I want to login so that I can edit my result
2	As a student I want to login so that I can edit other student's result
3	As a student I want to login so that I can edit/steal other student's information

Table 2.2. Negative User Stories [77]

It can be seen in the table that the negative user stories are mainly concentrated on the security instead of considering other side of the application.

2.4.1. User Story Templates

Cohn proposed a user story template that is organised as: I as a <role> want <function> so that <business value> [75]. Cohn proposed one more template structured as: As a <type of user> I want <capability> so that <business value> [79]. He offered several templates having the same structure but different keywords. Other agile experts suggested and proposed different templates of their own. Jeff Patton presented a user story template that is structured as: As a <user type> I want to <task to be performed> so that I can <achieve some goal>. These templates are also structured as the ones proposed by Cohn but a user story template is generally structured as: As [WHO], I want [WHAT] so that [WHY] [75]. A significantly different template was proposed which is structured as: In order to <receive benefit> as a <role>, I want <goal/desire>[75].

2.4.2. Epics and Themes

Epic and themes concept facilitate to plan, prioritize and organize a user story in agile methods which are driven by the user stories [75], [80]. A user story that is outsized and is difficult to be covered in one iteration is defined as an Epic [81]. A user story does not has a specific size to categorize it as an epic or not [81]. An epic user story is the one which is required to be decomposed into small user stories which can easily be implemented in one iteration. Some epics are large enough that they needed to be subdivided further into sub-epics [81]. A theme is a group of numerous smaller user stories that are related logically [81].

2.5.Research Design

This section discusses the proposed research design

2.5.1. Use case based testing related issues:

As we can see that UML diagrams play a vital role in many projects to develop test cases and use case diagrams are the ones used very frequently. Some of the teams faced little issues while implementing the use cases especially on a large scale. These problems are categorized and summarized into ten use case pitfalls which includes undefined or unreliable system boundary, the actor names are varying, large number of use cases, the relationship between actor and use case is similar to a spider's web, lengthy use case specifications, confusing use case specifications, ambiguous description of functional privilege, customer's inability to understand the use cases, use cases are always incomplete [82].

2.5.2. Components of user stories:

Cohn proposed a user story template that is generally structured as: As [WHO], I want [WHAT] so that [WHY] [75]. Based on this template it can be said that a user story consists of three main components which are as follows.

- User roles
- Goals
- Reasons

User roles define ‘Who’ which represents the different type of users that can interact with the system. Goals define ‘What’ which represents the purpose or aim of the user what he or she wants to achieve. Reasons define ‘Why’ which represents the motive of the user to perform a specific task or to achieve the goal.

2.5.3. Methodology

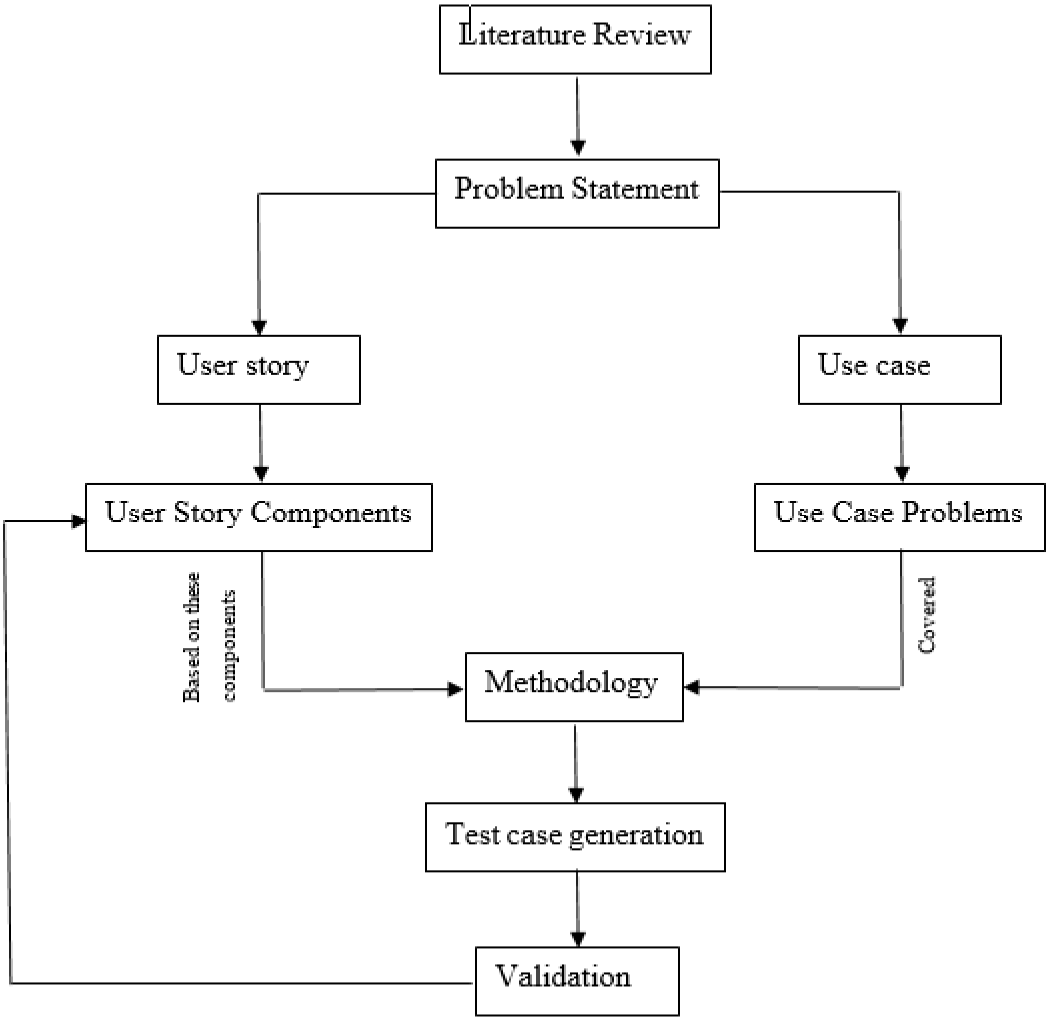


Figure. 2.7 Research Methodology

Chapter 3

OUR PROPOSAL

This chapter discusses the proposed methodology regarding generation test cases from user stories and a coverage metric which provides a complete coverage of user stories and test cases. Generation of test cases can be seen in figure 3.1.

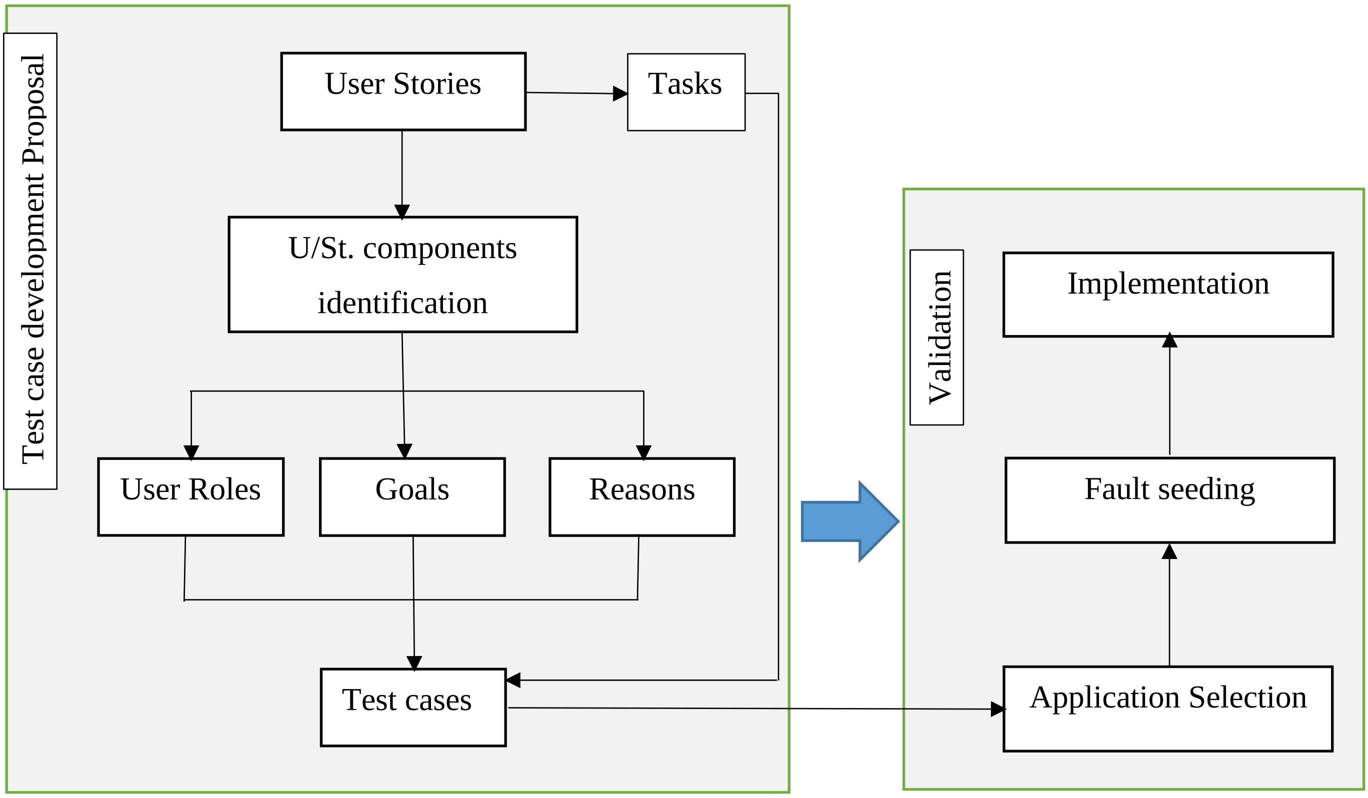


Figure 3.1 Our Proposal

Test cases will be generated from the user stories and fault seeding will be done in the application to validate the test cases against each user story.

3.1. Test Case Generation

A user story is a user requirement or need written in a single line on card. User stories will be used to generate test cases directly instead of driving them from specifications or models. As it is already mentioned in chapter 1 (section 1.1) that there are several approaches to design test cases from UML diagrams, Use case specifications and model based specifications [8,10-13]. So there's a need of a technique which is helpful in generating test cases from user stories and provides a complete mapping and coverage. In the above diagram a test case generation technique is given in which user stories will be used to generate the test cases. In case of user story it is quite rare that a single person can implement a full user story. In the process of achieving a user story multiple persons are involved like a programmer, tester, database designer, UI designer and analyst. Therefore these user stories are than divided into the different tasks which are tend to be done by a single person. A task can be something like designing a database, coding or designing a user interface. Meanwhile user story components are also identified in the user story which involves: (1) User Roles, (2) Goals and (3) Reasons. First of all user story components will be identified from the user story and meanwhile tasks will also be generated from the user stories. After identification of the components and tasks, test cases will be generated from these tasks as well as from the components. Test cases are developed by keeping these components and tasks in mind. So that a proper mapping can be done from test cases to the corresponding user stories by making sure that all the components and tasks are being covered. These components are used to make sure that the goals, roles and reasons of a user are met. Each user story may contain a single or multiple user roles, goals and reasons for achieving these goals and each task may generate one or more unit test cases. After generating the test cases a fault will be seeded in the application for which we are conducting the testing. After seeding the fault, test cases will be implemented to identify the fault and to validate the test case itself. Every test case represents a specific task or the component that needs to be achieved which in turn needs to be completed to achieve a user story. This fact will enable us to map the test cases to their respective user story and to decide which test cases are valid and which of them are invalid.

3.2. Test case generation steps

3.2.1. Identify user story components

First of all user story components (user roles, goals and reasons) are identified from the user stories.

3.2.2. Design test cases against components

Based on the components of user stories identified in the previous step, test cases will be generated which involves the following steps

3.2.2.1. Scenarios

Based on the user story components scenarios are generated.

3.2.2.2. Pre-condition and Post-condition

Pre-conditions and post-conditions are also identified

3.2.2.3. Concrete test cases

These scenarios, the pre and post-conditions will be used to generate the concrete test cases.

3.2.3. Identification of tasks

After test case generation based on the user story components tasks will be identified from the user stories.

3.2.4. Design test cases against tasks

Based on these tasks concrete test cases will be generated.

3.2.5. Identify positive and negative user stories

Positive and negative user stories will also be identified.

3.2.6. Design the negative user stories

User stories that generally identified are considered as positive user stories that's why we have to specifically design the negative user stories as the positive ones are already mentioned.

3.3. Running Example

Following is an example of test case generation for a mobile application based on the proposed technique. The application used is named as Numerical Method Solver (NMS). This is a mobile app which helps the user to solve the problems related to the Numerical Computing based on different numerical computing methods. There are several methods in Numerical Computing, but this application only contains ten of them. This example contains 5 user stories, based on these user stories components have been identified. It contains 14 tasks which are mentioned below and 46 test cases that are included in Appendix I.

U/St. 1: As a user I can see the list of methods upon accessing NMS app, so that I can select one of them.

User Role: Standard User

Goal: To see the list of methods

Reason: Able to select one of them

Scenario: Make sure to display the list of methods on the home screen so that user can select the desired method

Task 1: When user opens the application make sure that he/she is redirected to the landing/ home page.

Task 2: Make sure that the UI appears correctly on different screen sizes.

Task 3: Make sure that the following list of methods are available on the landing/home page.

- Bisection Method
- Iteration Method
- Newton-Raphson Method
- Regula-Falsi Method
- Jacobi’s Method
- Gauss-Seidel Method
- Lagrange’s Interpolation Formula
- Newton’s Forward Difference Formula
- Trapezoidal Rule
- Power Method

Task 4: Make sure that the user can select any of the methods appear on the home page.

U/St. 2: As a user I can see the method’s interface so that I can perform required calculation/task.

User Role: Standard User

Goal: Want to see method’s interface

Reason: To provide input values

Scenario: User should be able to see the method’s interface to perform the calculations

Task 1: Make sure that the user can see the method’s interface when the method is selected.

Task 2: Method’s interface UI must appear as per device screen and contains the required fields (input fields).

Task 3: Input fields must allow the user to enter input values.

U/St. 3: As a user I want the system to display an error message in case of invalid input so that I can provide a valid input.

User Role: Standard User

Goal: To get an error message in case of invalid inputs.

Reason: To provide valid input values.

Scenario: Make sure to display the error messages in case of an error

Task 1: System should check the input values against the standard input format and provide an appropriate error message to notify the user.
Task 2: User can edit/update the input values.

U/St. 4: As a user I want to have a solve/process button on every method's interface so that I can get the solution for the method.

User Role: Standard User

Goal: Want a solve button on every method's interface

Reason: To get a solution for the method.

Scenario: Make sure to have a solve button to perform the calculations.

Task 1: A solve button must be placed at the end/top of the method's interface
Task 2: Solve button must redirect the user to the solution / result page.

U/St. 5: As a user I want the result to be displayed so that I can see it.

User Role: Standard User

Goal: To display the results

Reason: want to see the results

Scenario: Make sure to display the results after required calculations

Task 1: Make sure that the result page is displayed to the user with text "solution for iteration 1, iteration 2 and so on".

Task 2: Results must be displayed correctly i.e. the whole solution.

Task 3: There is a home option available at the bottom/top right to result/solution page which redirect the user to landing/home page.

3.4. Test Coverage

A coverage metric is also proposed in this chapter which will be helpful to provide a complete information about coverage of user stories i.e. to ensure that all the user stories are covered and implemented. To make sure that all the user stories are tested completely and the system is fulfilling the user's need. The proposed coverage metric is as follows:

- All user story components are covered
- All tasks are covered
- Negative user stories are covered
- Positive user stories are covered

This coverage metric is proposed on the basis of tasks, user story components like user roles, goals, reasons, and positive and negative user stories. To make sure that a user story is fully covered and all the test cases are executed, this coverage metric will be very helpful.

3.5. Conclusion

Test case generation technique and coverage metric for user stories coverage is discussed in this chapter. It is useful to validate and map test cases to their respective user story to keep a track of valid and invalid test cases using fault seeding technique. Coverage metric is also defined in this chapter on the basis of which coverage of a user story is checked to make sure that it is fully covered.

Chapter 4

RESULTS AND DISCUSSION

This chapter discusses implementation of the proposed methodology on mobile applications named CariGari and Emergency Response. It also explains the model how it generates the test cases based on the user story components and the tasks with the implementation on a system under development. In the end this model is validated, and the results are displayed to support this validation.

4.1. CariGari

CariGari is an android based mobile application which will enable the user to control his/her vehicle using the smartphone. User will be able to control some of the components of the vehicle like start/stop the engine, lock/unlock the doors, estimating the fuel consumption, details of mileage covered against the fuel and history of places visited. The smartphone will be connected to an external Bluetooth device embedded in the vehicle's cabin using Bluetooth. After connecting the smartphone with the vehicle user will be able to control the vehicle and perform the required task.

4.2. Emergency Response

Emergency response is an android based mobile application which will enable a user to contact others in case of an emergency. It will initiate an emergency Email and SMS alert containing user's current location and this email and SMS will be sent to the contacts added in the application. It will keep a track of user's current location constantly. It will also allow the user to make an emergency call. Based on the current location it will be able to find out an optimum path, shortest distance and estimated time from source to destination. In case the user won't make it in time to the destination, it will generate an emergency scheduled SMS to inform the contacts. So that they can track the user to make sure of their safety.

4.3. Validation

For validation purposes test cases will be generated for CariGari and Emergency Response using the proposed methodology and then these test cases will be compared to the test cases generated by the development team of CariGari to validate that the proposed methodology can be used to generate more adequate test cases for software systems. To validate the proposed methodology based on the fault seeding, an application named "Car

Rental Management System" is selected. Following the fault types discussed in[83] faults are seeded in the application which are shown in the table 4.1.

Code Examples	
Correct Statement	**Mutant Statement**
readc();	// readc();
int number_of_daysc;	string number_of_daysc;
total_costc=(1000*number_of_daysc);	total_costc=0;
for(i=0;i<L;i++)	for(i=0;i>L;i++)
if(abc=='y'||abc=='Y')	if(abc=='end')

if(abc=='y'||abc=='Y')
{ u[i].displayc(); }
	if(abc=='y'||abc=='Y')
	{ //u[i].displayc(); }

fout1.open("accc.txt",ios::out|ios::app);	fout1.close("accc.txt",ios::out|ios::app);
fout1.close();	fout1.open();
fout1.open("accc.txt",ios::out|ios::app);	fout1.open("accc.txt",ios::out);

Table 4.1. Fault seeding in Car Rental Management System

After seeding the faults, manual testing has been done based on the test cases generated using the proposed methodology and all of the above-mentioned faults are detected by implementing the test cases (1,2,3,4,7,8,9,10,21,22, 27).

Following are some of the user stories, user story components and tasks generated from the user stories based on the proposed technique. Generated test cases based on the proposed methodology are included in the appendix II, appendix III and appendix IV.

4.4. Test case generation for CariGari

Sr. No.	**User Stories**	**User story components**			**Tasks based on user story**	**Scenarios based on user story**
		User Roles	**Goals**	**Reasons**		
1	As a user I want a GUI that helps a user to navigate through application to control the vehicle	Standard user	To have a GUI	To control the vehicle	**1.** Make sure that the dimensions of the elements are correct. **2.** Error messages must be displayed properly. **3.** Develop UI for different screens	Verify that the GUI of the application is correct and helpful in navigating the users

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
					(Different screen sizes). **4.** Make sure that UI appears correctly on different screen sizes. **5.** Design app icons for different screen sizes**.** **6.** Make sure that the font is readable. **7.** Text, buttons, text fields, icons and other elements must be aligned and in proper place	
2	As a user I want to login the system so that I can access the system	Standard User	To login the system	Want to access the system	**1.** Add input fields for username and password **2.** Make sure that the input fields allow the user to input the username and password **3.** Add a login button and make sure that it is working properly **4.** When login button is pressed it must verify the username and password **5.** When user enters the correct credentials login successfully	Check the login functionality

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
					6. When login credentials are invalid an error message must be generated **7.** A signup/ register button should be placed on the login page for unregistered users.	
3	As a user I want to register/ create login credentials so that I can login the system	Standard User	Registration	To login the system	**1.** Unregister user should see a signup/register form **2.** Add input fields for the required data on the signup form **3.** Make sure that the input fields allow the user to input data **4.** Input fields must verify the data entered by the user according to some standard pattern **5.** Place a submit button on the registration page **6.** On clicking the submit button the user must be registered successfully **7.** Login button on the registration page to redirect the user to the login page	Check the registration functionality

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
4	As a user I want a monthly report for the vehicle	Standard User	Monthly report	To keep a detailed history	**1.** Add a report option in the app **2.** Make sure that the report contains all the required details about vehicle like fuel consumption, distance covered and places visited	Verify the monthly report feature
5	As a user I want the system to generate an emergency SOS in case of any detected collision so that rescue and emergency contacts can be informed	Standard User	Emergency SOS	To inform the emergency contacts and rescue	**1.** User must be allowed to enter the details of emergency contact **2.** Make sure that an emergency SOS is generated **3.** Make sure that the SOS is sent to the specified contacts **4.** Make sure that the collision is detected	Verify the emergency SOS functionality
6	As a user I want the system to monitor the mileage and remaining fuel in the tank of vehicle	Standard User	Monitoring system	To estimate the mileage and remaining fuel in the tank	**1.** System must calculate the distance covered by the vehicle **2.** Make sure that an alert is generated when fuel is about to end **3.** Make sure that the system can estimate the distance that can be covered in the current amount of fuel	Validate the fuel monitoring feature

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
7	As a user I want the system to lock the doors of vehicle	Standard User	Lock the doors of vehicle	To secure the vehicle	**1.** Place a button named lock in the application interface **2.** Make sure that the lock button locks the vehicle's doors	Validate the door lock functionality
8	As a user I want the system to unlock doors of vehicle	Standard User	Unlock the doors of vehicle	To get in or get out of the vehicle	**1.** Place a button named unlock in the application interface **2.** Make sure that the unlock button unlocks the vehicle's doors	Validate the door unlock functionality
9	As a user I want the system to start the engine	Standard User	start the engine	want to drive the vehicle	**1.** Place a button named start in the application interface **2.** Make sure that the car's engine starts when the start button is pressed	verify the system's ability to start the engine
10	As a user I want the system to stop the engine	Standard User	To stop the engine	Not driving the car	**1.** Place a button named stop in the application interface **2.** Make sure that the car's engine stops when the stop button is pressed	Verify the system's ability to stop the engine
11	As a user I want the device to connect	Standard User	Connect the device with the vehicle	To control the vehicle with device	**1.** Make sure that the vehicle can be controlled from device	Verify that the device connects

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
	with the vehicle using Bluetooth		using Bluetooth		using Bluetooth **2.** Bluetooth should be visible to each other	to the vehicle via Bluetooth
12	As a user I want to view the history of places visited by the vehicle	Standard User	To view the history	To keep a record	**1.**Add an option to view the history of visited places **2.**The history must contain the places visited by the vehicle	Validate that the vehicle history can be viewed
13	As a user I want to delete the history of visited places	Standard User	To delete the history	Record not required any longer	**1.**Add an option to delete the history of visited places **2.**Make sure that the history is deleted **3.**Display a "deleted successfully" message on successfully deleting the history	Validate that the vehicle history can be deleted

Table 4.2. Tasks and Scenarios based on user stories for CariGari

4.4.1. Identify user story components

First of all user story components (user roles, goals and reasons) are identified from the user stories.

U/St. 1. As a user I want a GUI that helps a user to navigate through application to control the vehicle

User role: Standard user

Goal: To have a GUI

Reason: to control the vehicle

U/St. 2. As a user I want to login the system so that I can access the system

User role: Standard User

Goal: To login the system

Reason: want to access the system

U/St. 3. As a user I want to register/ create login credentials so that I can login the system

User role: Standard User

Goal: Registration

Reason: to login the system

U/St. 4. As a user I want a monthly report for the vehicle

User role: Standard User

Goal: monthly report

Reason: to keep a detailed history

U/St. 5. As a user I want the system to generate an emergency SOS in case of any detected collision so that rescue and emergency contacts can be informed

User role: Standard User

Goal: emergency SOS

Reason: to inform the emergency contacts and rescue

U/St. 6. As a user I want the system to monitor the mileage and remaining fuel in the tank of vehicle

User role: Standard User

Goal: monitoring system

Reason: to estimate the mileage and remaining fuel in the tank.

U/St. 7. As a user I want the system to lock the doors of vehicle

User role: Standard User

Goal: lock the doors of vehicle

Reason: to secure the vehicle

U/St. 8. As a user I want the system to unlock doors of vehicle

User role: Standard User

Goal: unlock the doors of vehicle

Reason: to get in or get out of the vehicle

U/St. 9. As a user I want the system to start the engine

User role: Standard User

Goal: start the engine

Reason: want to drive the vehicle

U/St. 10. As a user I want the system to stop the engine

User role: Standard User

Goal: to stop the engine

Reason: not driving the car

U/St. 11. As a user I want the device to connect with the vehicle using Bluetooth

User role: Standard User

Goal: connect the device with the vehicle using Bluetooth

Reason: to control the vehicle with device

U/St. 12. As a user I want to view the history of places visited by the vehicle

User role: Standard User

Goal: to view the history

Reason: to keep a record

U/St. 13. As a user I want to delete the history of visited places.

User role: Standard User

Goal: to delete the history

Reason: record not required any longer

4.4.2. Design test cases against components

Based on the components of user stories identified in the previous step, test cases will be generated which involves the following steps

4.4.2.1. Generating Scenarios

Based on the user story components scenarios are generated.

Scenario for U/St. 1: verify that the GUI of the application is correct and helpful in navigating the users

Scenario for U/St. 2: check the login functionality

Scenario for U/St. 3: check the registration functionality

Scenario for U/St. 4: verify the monthly report feature

Scenario for U/St. 5: verify the emergency SOS functionality

Scenario for U/St. 6: validate the fuel monitoring feature

Scenario for U/St. 7: validate the door lock functionality

Scenario for U/St. 8: validate the door unlock functionality

Scenario for U/St. 9: verify the system's ability to start the engine

Scenario for U/St. 10: verify the system's ability to stop the engine

Scenario for U/St. 11: verify that the device connects to the vehicle via Bluetooth

Scenario for U/St. 12: validate that the vehicle history can be viewed

Scenario for U/St. 13: validate that the vehicle history can be deleted

4.4.2.2. Concrete test cases

These scenarios, the pre and post-conditions will be used to generate the concrete test cases which are included in the appendix II.

4.4.3. Identification of tasks

After test case generation based on the user story components tasks will be identified from the user stories.

Tasks for U/St. 1

Task 1: make sure that the dimensions of the elements are correct

Task 2: error messages must be displayed properly

Task 3: Develop UI for different screens (Different screen sizes)

Task 4: Make sure that UI appears correctly on different screen sizes

Task 5: Design app icons for different screen sizes**.**

Task 6: Make sure that the font is readable

Task 7: text, buttons, text fields, icons and other elements must be aligned and in proper place

Tasks for U/St. 2

Task 1: Add input fields for username and password

Task 2: Make sure that the input fields allow the user to input the username and password

Task 3: Add a login button and make sure that it is working properly

Task 4: When login button is pressed it must verify the username and password

Task 5: when user enters the correct credentials login successfully

Task 6: when login credentials are invalid an error message must be generated

Task 7: a signup/ register button should be placed on the login page for unregistered users.

Tasks for U/St. 3

Task 1: unregister user should see a signup/register form

Task 2: add input fields for the required data on the signup form

Task 3: make sure that the input fields allow the user to input data

Task 4: input fields must verify the data entered by the user according to some standard pattern

Task 5: place a submit button on the registration page

Task 6: on clicking the submit button the user must be registered successfully

Task 7: Login button on the registration page to redirect the user to the login page

Tasks for U/St. 4

Task 1: add a report option in the app

Task 2: Make sure that the report contains all the required details about vehicle like fuel consumption, distance covered, and places visited.

Tasks for U/St. 5

Task 1: User must be allowed to enter the details of emergency contact

Task 2: Make sure that an emergency SOS is generated

Task 3: make sure that the SOS is sent to the specified contacts

Task 4: make sure that the collision is detected

Tasks for U/St. 6

Task 1: System must calculate the distance covered by the vehicle

Task 2: Make sure that an alert is generated when fuel is about to end

Task 3: make sure that the system can estimate the distance that can be covered in the current amount of fuel

Tasks for U/St. 7

Task 1: Place a button named lock in the application interface

Task 2: Make sure that the lock button locks the vehicle’s doors

Tasks for U/St. 8

Task 1: Place a button named unlock in the application interface

Task 2: Make sure that the unlock button unlocks the vehicle’s doors

Tasks for U/St. 9

Task 1: Place a button named start in the application interface

Task 2: Make sure that the car’s engine starts when the start button is pressed.

Tasks for U/St. 10

Task 1: Place a button named stop in the application interface

Task 2: Make sure that the car’s engine stops when the stop button is pressed.

Tasks for U/St. 11

Task 1: Make sure that the vehicle can be controlled from device using Bluetooth

Task 2: Bluetooth should be visible to each other

Tasks for U/St. 12

Task 1: Add an option to view the history of visited places

Task 2: The history must contain the places visited by the vehicle

Tasks for U/St. 13

Task 1: Add an option to delete the history of visited places

Task 2: Make sure that the history is deleted

Task 3: Display a “deleted successfully” message on successfully deleting the history.

4.4.4. Design test cases against tasks

Based on the tasks identified from the user stories concrete test cases will be generated. Generated test cases are included in the appendices.

4.4.5. Identify positive and negative user stories

Positive and negative user stories will also be identified.

4.4.6. Design the negative user stories

User stories that generally identified are considered as positive user stories that's why we have to specifically design the negative user stories as the positive ones have already been created. Test cases generated based on the negative user stories are included in appendices.

Neg. U/St. 1. As an unregistered user I want to login the system so that I can access the system

User role: Unregistered user

Goal: want to login

Reason: to access the system

Neg. U/St. 2. As a user I want to edit the monthly report of the vehicle

User role: Standard user

Goal: edit monthly report

Reason: to create some discrepancy

Neg. U/St. 3. As an unregistered user I want to login so that I can change the emergency contact details of a user

User role: Unregistered user

Goal: Login the system

Reason: To change the emergency contact details

Neg. U/St. 4. As a user I want to edit the history of places visited by the vehicle

User role: Standard user

Goal: To edit the history of visited places

Reason: to temper the history

4.5. Test case generation for Emergency Response

Sr. No.	**User Stories**	**User story components**			**Tasks based on user story**	**Scenarios based on user story**
		User Roles	**Goals**	**Reasons**		
1	As a user I want a GUI that helps a	Standard user	To have a GUI	To manage the contacts	1. Make sure that the dimensions of	Verify that the GUI of the

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
	user to navigate through application to manage the contacts and emergency responses			and generation of emergency SMS, emails and calls	the elements are correct. 2. Error messages must be displayed properly 3. Make sure that UI appears correctly on different screen sizes 4. Make sure that the font is readable 5. Text, buttons, text fields, icons, images and other elements must be aligned properly	application is correct and helpful in navigating the users to manage the contacts and emergency responses
2	As a user I want to manage contacts in the application so that these contacts can be used in case of emergency	Standard user	To manage contacts	To use them in case of emergency	1. Add "manage contacts" button/option on the application home screen 2. Make sure that the manage contacts contain add, update and delete contact options. 3. Add input fields for name, contact number and email to add a new contact 4. Add an "Add contact" button 5. Make sure that the input fields allow the user to input the name, contact number and email address	Verify that the application allows the user to manage contacts

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
					6. When “Add contact” button is pressed the contact details should be added successfully 7. Add an update button to update the contact details 8. Make sure that the contact details are updated when the update button is pressed 9. Add a delete button to delete a contact 10. Make sure that the contact details are deleted when the delete button is pressed	
3	As a user I want the application to fetch the current location of the device using the GPS coordinates so that it can be sent to the predefined contacts in case of emergency.	Standard user	To fetch the current location of the device	To send it to the predefined contacts in case of emergency	1. Add a current location option on the home screen 2. Make sure that the location can be fetched correctly using the device GPS 3. In case the GPS is not enabled it should ask the user to activate the GPS 4. Make sure to add a GUI to show the current location of the device	Verify that the application is able to fetch the current location of device

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
4	As I user I want the application to calculate the optimum path and travel time so that a scheduled emergency SMS can be sent to the contacts in case the user doesn't reach the destination	Standard user	Calculate optimum path and travel time	To generate an emergency SMS in case the user doesn't reach the destination	1. Add a scheduled emergency SMS option on the application home screen 2. Make sure that the scheduled emergency SMS option contains a button named "Distance finder" 3. On clicking the Distance Finder button user should be able to see a map with its current location 4. It must allow the user to select a destination 5. Make sure to calculate the optimum path and travel time based on the source and destination 6. It must generate an emergency SMS in case user is unable to reach	Validate that the application is able to calculate optimum time, distance and path

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
					the destination in the calculated travel time 7. It should allow user to set a duration time for a scheduled emergency SMS 8. User must be able to activate or cancel the scheduled emergency SMS	
5	As a user I want the application to send an Email and SMS to all the contacts added so that they can be informed in case of an emergency	Standard user	To send an Email and SMS to all the contacts added in database	To inform them in case of emergency	1. Add an option to send an Email and SMS on the application home screen 2. Make sure to ask the user before sending the SMS and Email 3. Make sure to send the SMS and Email if user presses the send button 4. Make sure to cancel the SMS and Email if user presses the cancel button 5. Application must display a message upon successful	Validate that the application can send the Email and SMS

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
					delivery of Email and SMS 6. Application must display a message upon failure of delivery of emergency Email and SMS 7.	
6	As a user I want the application to generate an emergency call so that the user can contact the person in case of an emergency	Standard user	To generate an emergency call	To contact the predefined person	1. Add an option for emergency calls on the application home screen 2. Make sure to ask the user before initiating the call 3. Make sure to initiate the call, in case user presses the call button 4. Make sure to cancel the call if user presses the cancel button	Verify that the application is able to make a call to the person

Table 4.3. Tasks and Scenarios based on user stories for Emergency Response

4.5.1. Identify user story components

First of all user story components (user roles, goals and reasons) are identified from the user stories.

U/St. 1. As a user I want a GUI that helps a user to navigate through application to manage the contacts and emergency responses
User role: Standard user
Goal: To have a GUI
Reason: To manage the contacts and generation of emergency SMS, emails and calls

U/St. 2. As a user I want to manage contacts in the application so that these contacts can be used in case of emergency
User role: Standard user
Goal: To manage contacts
Reason: To use them in case of emergency

U/St. 3. As a user I want the application to fetch the current location of the device using the GPS coordinates so that it can be sent to the predefined contacts in case of emergency.
User role: Standard user
Goal: To fetch the current location of the device
Reason: To send it to the predefined contacts in case of emergency

U/St. 4. As a user I want the application to calculate the optimum path and travel time so that a scheduled emergency SMS can be sent to the contacts in case the user doesn't reach the destination
User role: Standard user
Goal: Calculate optimum path and travel time
Reason: To generate an emergency SMS in case the user doesn't reach the destination

U/St. 5. As a user I want the application to send an Email and SMS to all the contacts added so that they can be informed in case of an emergency
User role: Standard user
Goal: To send an Email and SMS to all the contacts added in database
Reason: To inform them in case of emergency

U/St. 6. As a user I want the application to generate an emergency call so that the user can contact the person in case of an emergency
User role: Standard user

Goal: To generate an emergency call

Reason: To contact the predefined person

4.5.2. Design test cases against components

Based on the components of user stories identified in the previous step, test cases will be generated which involves the following steps

4.5.2.1. Generating Scenarios

Based on the user story components scenarios are generated.

Scenario for U/St. 1: Verify that the GUI of the application is correct and helpful in navigating the users to manage the contacts and emergency responses

Scenario for U/St. 2: Verify that the application allows the user to manage contacts

Scenario for U/St. 3: Verify that the application is able to fetch the current location of device

Scenario for U/St. 4: Validate that the application is able to calculate optimum time, distance and path

Scenario for U/St. 5: Validate that the application can send the Email and SMS

Scenario for U/St. 6: Verify that the application is able to make a call to the person

4.5.2.2. Concrete test cases

These scenarios, the pre and post-conditions will be used to generate the concrete test cases which are included in the appendix III.

4.5.3. Identification of tasks

After test case generation based on the user story components tasks will be identified from the user stories.

Tasks for U/St. 1

Task 1: Make sure that the dimensions of the elements are correct

Task 2: Error messages must be displayed properly

Task 3: Make sure that UI appears correctly on different screen sizes

Task 4: Make sure that the font is readable

Task 5: Text, buttons, text fields, icons, images and other elements must be aligned properly

Tasks for U/St. 2

Task 1: Add “manage contacts” button/option on the application home screen

Task 2: Make sure that the manage contacts contain add, update and delete contact options

Task 3: Add input fields for name, contact number and email to add a new contact

Task 4: Add an “Add contact” button

Task 5: Make sure that the input fields allow the user to input the name, contact number and email address

Task 6: When “Add contact” button is pressed the contact details should be added successfully

Task 7: Add an update button to update the contact details

Task 8: Make sure that the contact details are updated when the update button is pressed

Task 9: Add a delete button to delete a contact

Task 10: Make sure that the contact details are deleted when the delete button is pressed

Tasks for U/St. 3

Task 1: Add a current location option on the home screen

Task 2: Make sure that the location can be fetched correctly using the device GPS

Task 3: In case the GPS is not enabled it should ask the user to activate the GPS

Task 4: Make sure to add a GUI to show the current location of the device

Tasks for U/St. 4

Task 1: Add a scheduled emergency SMS option on the application home screen

Task 2: Make sure that the scheduled emergency SMS option contains a button named “Distance finder”

Task 3: On clicking the Distance Finder button user should be able to see a map with its current location

Task 4: It must allow the user to select a destination

Task 5: Make sure to calculate the optimum path and travel time based on the source and destination

Task 6: It must generate an emergency SMS in case user is unable to reach the destination in the calculated travel time

Task 7: It should allow user to set a duration time for a scheduled emergency SMS

Task 8: User must be able to activate or cancel the scheduled emergency SMS

Tasks for U/St. 5

Task 1: Add an option to send an Email and SMS on the application home screen

Task 2: Make sure to ask the user before sending the SMS and Email

Task 3: Make sure to send the SMS and Email if user presses the send button

Task 4: Make sure to cancel the SMS and Email if user presses the cancel button

Task 5: Application must display a message upon successful delivery of Email and SMS

Task 6: Application must display a message upon failure of delivery of emergency Email and SMS

Tasks for U/St. 6

Task 1: Add an option for emergency calls on the application home screen

Task 2: Make sure to ask the user before initiating the call

Task 3: Make sure to initiate the call, in case user presses the call button

Task 4: Make sure to cancel the call if user presses the cancel button

4.5.4. Design test cases against tasks

Based on the tasks identified from the user stories concrete test cases will be generated. Generated test cases are included in the appendix III.

4.6. Test case generation for Car rental management system

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
1	As a user I want the system to read the information for the car, so that it can be stored.	Manager	To read the information of the car	To store record of the car being rented	1. Required input fields should be available 2. System should allow the user to enter the information 3. System should be able to store the car's information in a file	Verify that the system reads the information for the car
2	As a user I want the system to display the information of	Manager	System should display the	To verify the information	1. System should display the complete car's information	Verify that the system displays

Sr. No.	User Stories	User story components			Tasks based on user story	Scenarios based on user story
		User Roles	Goals	Reasons		
	the car, so that it can be verified		information of the car			the car information
3	As a user I want the system to perform the required calculations for the car, so that profit can be calculated	Manager	System should perform all the necessary calculation	to find out the profit against the car	1. Ask the user to enter the monthly cost of fuel used 2. Ask the user to enter the monthly cost of maintenance 3. Calculate the profit based on the car's total rent and monthly cost of fuel and maintenance	verify the profit calculation
4	As a user I want the system to calculate the total cost of rent for the car	Manager	to calculate the total rent	to charge the customer accordingly	1. Make sure to ask the user to enter the number of days for which car is being rented 2. Make sure to set the per day rent for the car 3. Make sure to display the total rent of the car	verify the rent calculation

Table 4.4. Tasks and Scenarios based on user stories for Car Rental Management System

4.6.1. Identify user story components

First of all user story components (user roles, goals and reasons) are identified from the user stories.

U/St. 1. As a user I want the system to read the information for the car, so that it can be stored.

User Roles: Manager

Goals: To read the information of the car

Reasons: To store record of the car being rented

U/St. 2. As a user I want the system to display the information of the car, so that it can be verified

User Roles: Manager

Goals: System should display the information of the car

Reasons: To verify the information

U/St. 3. As a user I want the system to perform the required calculations for the car, so that profit can be calculated

User Roles: Manager

Goals: System should perform all the necessary calculations

Reasons: to find out the profit against the car

U/St. 4. As a user I want the system to calculate the total cost of rent for the car

User Roles: Manager

Goals: to calculate the total rent

Reasons: to charge the customer accordingly

4.6.2. Design test cases against components

Based on the components of user stories identified in the previous step, test cases will be generated which involves the following steps

4.6.2.1. Generating Scenarios

Based on the user story components scenarios are generated.

Scenario for U/St. 1: Verify that the system reads the information for the car

Scenario for U/St. 2: Verify that the system displays the car information

Scenario for U/St. 3: verify the profit calculation

Scenario for U/St. 4: verify the rent calculation

4.6.2.2. Concrete test cases

These scenarios, the pre and post-conditions will be used to generate the concrete test cases which are included in the appendix IV.

4.6.3. Identification of tasks

After test case generation based on the user story components tasks will be identified from the user stories.

Tasks for U/St. 1

Task 1: Required input fields should be available

Task 2: System should allow the user to enter the information

Task 3: System should be able to store the car's information in a file

Tasks for U/St. 2

Task 1: System should display the complete car's information

Tasks for U/St. 3

Task 1: Ask the user to enter the monthly cost of fuel used

Task 2: Ask the user to enter the monthly cost of maintenance

Task 3: Calculate the profit based on the car's total rent and monthly cost of fuel and maintenance

Tasks for U/St. 4

Task 1: Make sure to ask the user to enter the number of days for which car is being rented.

Task 2: Make sure to set the per day rent for the car

Task 3: Make sure to display the total rent of the car

4.6.4. Design test cases against tasks

Based on the tasks identified from the user stories concrete test cases will be generated. Generated test cases are included in the appendix IV.

4.7. Results and discussion

In section 4.4, 4.5 and 4.6 the proposed methodology for test case generation from user stories is implemented to generate the test cases for different applications. Based on our proposed methodology test cases are generated from the user stories of the applications and these test cases are covering almost all the functionalities of the applications to make sure that the applications can be fully tested. 103 test cases for CariGari, 86 test cases for Emergency Response and 28 test cases for Car rental management system are generated based on the user story components, tasks, positive and negative user stories to ensure the bug free application and to make sure that all the features are working properly. Based on the coverage metric it can be concluded that these test cases are providing maximum coverage as all the user story components are covered, all tasks are covered, positive and negative user stories are also covered in the test cases generated.

4.8. Conclusion

In this chapter the proposed methodology is evaluated and validated by generating the test cases for a mobile application using this test case generation technique. A step by step test case generation is shown which ensures the availability of the maximum test cases to test all the features of the application.

Chapter 5

CONCLUSION AND FUTURE WORK

This chapter concludes our work and provides a summary of the work done so far. It also involves the work that can be done in the future.

5.1. Conclusion

Our first objective was to propose a technique that can generate test cases from user stories. Based on the studies of test case generation techniques it is observed that a technique for test case generation from user stories is not available. All the available techniques are generating test cases based on models or specifications. Based on this gap a technique is introduced in chapter 3. In this technique user stories are used to identify the user story components and tasks. Based on these components test cases are generated. Tasks are also used to generate the test cases separately. This technique provides maximum number of test cases as it involves test cases based on the user roles, goals and reason as well as based on the tasks. This technique generates the test cases based on user stories only instead of models or specifications as discussed on chapter 2. Our second objective was to propose a coverage metric for the test coverage. Based on that objective a coverage metric has also been introduced that provides a complete coverage for the test cases as it measures the coverage based on user story components, tasks, positive and negative user stories. This technique is validated by generating test cases for an application and implementing those test cases to uncover the faults seeded in the very application.

5.2. Future work

This work can be extended in future by focusing on the negative user stories more precisely and by identifying the user story components for negative user stories. In future a standard or a formula can be derived to provide a number of test cases that can possibly be generated from a specific number of user stories based on their components and the tasks identified.

REFERENCES

[1] "1. Pirzadeh, Heidar, Andre de Santi Oliveira, and... - Google Scholar." [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=1.%09Pirzadeh%2C+Heidar%2C+Andre+de+Santi+Oliveira%2C+and+Sara+Shanian.+%22ReUse%3A+A+Recommendation+System+for+Implementing+User+Stories.%22+ICSEA+2016+%282016%29%3A+162&btnG=. [Accessed: 21-Oct-2019].

[2] M. Lindvall *et al.*, "Empirical Findings in Agile Methods," 2002.

[3] K. Schwaber, "SCRUM Development Process," in *Business Object Design and Implementation*, Springer London, 1997, pp. 117–134.

[4] R. Löffler, B. Güldali, and S. Geisen, "Towards Model-based Acceptance Testing for Scrum."

[5] D. Faragó, "Model-based Testing in Agile Software Development."

[6] M. Landhäußer and A. Genaid, "Connecting User Stories and Code for Test Development."

[7] R. Binder, *Testing object-oriented systems: models, patterns, and tools*. 2000.

[8] S. Kumar Swain, D. Prasad Mohapatra, and R. Mall, "Swain et al Test Case Generation Based on Use case and Sequence Diagram Test Case Generation Based on Use case and Sequence Diagram."

[9] S. Muchnick, *Advanced compiler design implementation*. 1997.

[10] C. Wang, F. Pastore, A. Goknil, and L. C. Briand, "Automatic Generation of System Test Cases from Use Case Specifications: an NLP-based Approach," Jul. 2019.

[11] C. Nebut, F. Fleurey, and Y. Le Traon, "Automatic Test Generation: A Use Case Driven Approach," 2006.

[12] J. Dick and A. Faivre, "Automating the generation and sequencing of test cases from model-based specifications," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1993, vol. 670 LNCS, pp. 268–284.

[13] H. Kim, S. Kang, J. Baik, and I. Ko, "Test Cases Generation from UML Activity Diagrams," 2007.

[14] P. S. ; D. Carrington, "A framework for specification-based testing," *ieeexplore.ieee.org*, vol. Volume: 22, no. Issue: 11, pp. 777–793.

[15] J. Tretmans, “Model-based testing and some steps towards test-based modelling,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 2011, vol. 6659 LNCS, pp. 297–326.

[16] A. Tanenbaum, “In defense of program testing, or correctness proofs considered harmful,” 1976.

[17] D. J. Richardson, O. O’malley, and C. Tittle, “Approaches to Specification-Based Testing. Approaches to Specication-Based Testing.”

[18] J. Goodenough, J. B. Go0denough, and S. L. Gerhart, “Toward a Theory of Test Data Selection. Eliminative Argumentation View project TOWARD A THEORY OF TEST DATA SELECTION.”

[19] J. G.-I. T. on software engineering and undefined 1983, “A mathematical framework for the investigation of testing,” *ieeexplore.ieee.org*.

[20] … J. L.-] P. S. W. on S. T. and undefined 1988, “Testing in top-down program development,” *ieeexplore.ieee.org*.

[21] T. J. Ostrand and M. J. Balcer, “THE CATEGORY-PARTITION METHOD FOR SPECIFYING AND GENERATING FUNCTIONAL TESTS.”

[22] E. J. Weyuker and T. J. Ostrand, “Theories of Program Testing and the Application of Revealing Subdomains,” 1980.

[23] D. J. Richardson Lord and A. C. Abstract, “A PARFITION ANALYSIS METHOD TO INCREASE PROGRAM RELIABILITY.”

[24] D. Richardson, L. C.-I. T. on Software, and undefined 1985, “Partition analysis: A method combining testing and verification,” *ieeexplore.ieee.org*.

[25] W. H.-I. T. on S. Engineering and undefined 1986, “A functional approach to program testing and analysis,” *ieeexplore.ieee.org*.

[26] L. Bougé, N. Choquet, L. Fribourg, M. G.-J. of S. and, and undefined 1986, “Test sets generation from algebraic specifications using logic programming,” *Elsevier*.

[27] M. Gaudel and B. . Marre, “Algebraic specifications and software testing: Theory and application,” 1988.

[28] A. Gopal and T. Budd, “Program testing by specification mutation,” 1983.

[29] J. Goguen, … J. T. C. on S. of R., and undefined 1979, “An introduction to OBJ: A language for writing and testing formal algebraic program specifications,” *IEEE CS Press*.

[30] D. Luckham, F. V. H.-I. Software, and undefined 1985, “An overview of Anna, a

specification language for Ada," *search.proquest.com*.

[31] F. V.-J. of S. and Software and undefined 1987, "A method for test data selection," *Elsevier*.

[32] M. Fowler, K. Scott, and A. Wesley, "UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language," 2000.

[33] A. Z. Javed, P. Strooper, A. Z. Javed, P. A. Strooper, and G. N. Watson, "Automated Generation of Test Cases Using Model-Driven Architecture," 2007.

[34] A. Cavarra, C. Crichton, J. Davies, … A. H.-P. of, and undefined 2002, "Using UML for automatic test generation," *academia.edu*.

[35] F. Bouquet *et al.*, "A Test Generation Solution to Automate Software Testing. A Test Generation Solution to Automate Software Testing," 2008.

[36] M. Mussa, S. Ouchani, A. Hamou-Lhadj, and W. Al Sammane, "A Survey of Model-Driven Testing Techniques Anomaly Detection using Big Data Analytic Tools View project Model-Based Security View project A Survey of Model-Driven Testing Techniques," 2009.

[37] Q. Yuan, J. Wu, C. Liu, L. Z.-2008 10th International, and undefined 2008, "A model driven approach toward business process test case generation," *ieeexplore.ieee.org*.

[38] P. Baker, Z. Dai, J. Grabowski, and I. Schieferdecker, *Model-driven testing: Using the UML testing profile*. 2007.

[39] D. Vega, I. Schieferdecker, … G. D. C. on Q. E. in S., and undefined 2008, "A TTCN-3 based Test Automation Framework for HL7-based Applications and Components."

[40] S. J. Mellor, "MDA Distilled."

[41] J. Warmer and A. Kleppe, *The object constraint language: getting your models ready for MDA*. 2003.

[42] M. Zohaib Iqbal, Z. I. Malik, A. Nadeem, Q.-U.-A. Farooq, M. Zohaib, and Z. Iqbal, "An approach for selective state machine based regression testing," 2007.

[43] J. Hartmann, M. Vieira, H. Foster, and A. Ruder, "UML-based Test Generation and Execution."

[44] T. Thanh and D. Trong, "A Systematic Procedure for Testing UML Designs," 2003.

[45] D. Leffingwell, *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. 2010.

[46] N. H. Madhavji, “Process cycle,” *Softw. Eng. J.*, vol. 6, no. 5, pp. 234–242, 1991.

[47] N. Abbas, A. M. Gravell, and G. B. Wills, “Historical roots of agile methods: Where did ‘Agile thinking’ come from?,” in *Lecture Notes in Business Information Processing*, 2008, vol. 9 LNBIP, pp. 94–103.

[48] “Iterative and Incremental Development: A Brief History,” 2003.

[49] B. Pressman, Roger & Maxim, *Software Engineering: A Practitioner’s Approach, 8th Ed.*, 8th ed. SEM; 8 edition (January 23, 2014), 2014.

[50] I. Sommerville, *Software Engineering*. 2010.

[51] L. Lindstrom and R. Jeffries, “Extreme programming and agile software development methodologies,” *Inf. Syst. Manag.*, vol. 21, no. 3, pp. 41–52, 2004.

[52] W. R.-L. Angeles and undefined 1970, “Managing the development of large software systems. proceedings of IEEE WESCON.”

[53] B. Kevin Forsberg, “The Relationship of System Engineering to the Project Cycle,” 1994.

[54] B. B.- Computer and undefined 1988, “A spiral model of software development and enhancement,” *computer.org*.

[55] J. Martin, *Rapid application development*. Macmillan Pub. Co., 1991.

[56] P. Kruchten, “The Rational Unified Process: An Introduction, 3a Edição,” 2003.

[57] D. Cohen, M. Lindvall, and P. Costa, “An Introduction to Agile Methods,” 2004.

[58] U. A. M.-O. http://www. agilemanifesto. org–2003/05/20 and undefined 2001, “Manifesto for agile software development.”

[59] J. Wiley and S. Ambler, “Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process.”

[60] S. Heng, “Impact of Unified User-Story-Based Modeling on Agile Methods: Aspects on Requirements, Design and Life Cycle Management Using Goal-Oriented Requirements Engineering in Agile Methods for Large Scale User Story Management View project.”

[61] J. Hunt, *Agile software construction*. Springer London, 2006.

[62] “Agile Software Development Smiglo View project Agile 2.0 View project Torgeir Dingsøyr SINTEF,” 2010.

[63] “Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch.” [Online]. Available: https://www.infoq.com/articles/standish-chaos-2015/. [Accessed: 30-Oct-2019].

[64] V. One, “The 10th Annual State of Agile Report.(2016),” 2016.

[65] K. Schwaber, J. S.- Nut, undefined Bolts, and O. of an A. Framework, and undefined 2011, "The Scrum Papers."

[66] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, "New Directions on Agile Methods: A Comparative Analysis."

[67] F. Tsui, O. Karam, and B. Bernal, *Essentials of software engineering*. 2013.

[68] K. Abad, W. Pérez, J. P. Carvallo, and X. Franch, "i* in Practice: Identifying Frequent Problems in its Application."

[69] K. Schwaber and M. Beedle, *Agile software development with Scrum*. 2002.

[70] K. Beck and M. Fowler, *Planning extreme programming*. 2001.

[71] M. C.-M. S. R. on May and undefined 2008, "Advantages of the 'As a user, I want' user story template."

[72] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-User Development: An Emerging Paradigm," in *End User Development*, Springer Netherlands, 2006, pp. 1–8.

[73] S. Dimitrijević, J. Jovanović, V. D.-I. and software, and undefined 2015, "A comparative study of software tools for user story management," *Elsevier*.

[74] S. A.-R. from A. Modeling and undefined 2014, "User Stories: An Agile Introduction."

[75] A. Xp and M. Cohn, "User Stories Applied For Agile Software Development," 2004.

[76] P. Kamthan, N. S.-2016 I. 17th International, and undefined 2016, "Modeling Negative User Stories is Risky Business," *ieeexplore.ieee.org*.

[77] M. Chopade, N. D.-2017 2nd International, and undefined 2017, "Agile software development: Positive and negative user stories," *ieeexplore.ieee.org*.

[78] P. Kamthan and N. Shahmir, "A Characterization of Negative User Stories," 2016.

[79] "Cohn, M. (2005). Agile Estimating and Planning. Prentice... - Google Scholar." [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Cohn%2C+M.+%282005%29.+Agile+Estimating+and+Planning.+Prentice+Hall+PTR%2C+Upper+Saddle+River%2C+NJ%2C+USA&btnG=. [Accessed: 24-Jan-2020].

[80] K. Rubin, *Essential Scrum: A practical guide to the most popular Agile process*. 2012.

[81] M. Cohn, "Succeeding with Agile: Software Development using Scrum, Addison-Wesley," 2009.

[82] S. L.-P. of T. of Object-Oriented and U. 1999, “Use case pitfalls: top 10 problems from real projects using use cases,” *ieeexplore.ieee.org*.

[83] A. Pasquini, E. D. A.-C. E. Practice, and undefined 1995, “Fault seeding for software reliability model validation,” *Elsevier*.

APPENDIX I

Test cases for Numerical Method Solver

Test case ID: 1 **Test case Name: List of methods appear successfully**		
Summary: Make sure that the list of methods appears on the landing/home page when user opens the app		
Preconditions: The application must be installed on a device.		
Step actions:	**Input values**	**Expected Results:**
1. Open the application	1. Tap on application icon	Home page is opened, and list of methods is appeared

Test case ID: 2 **Test case Name: List of methods not appear**		
Summary: User opens the application, but list of methods not appear on the home page		
Preconditions: The application must be installed on a device		
Step actions:	**Input values**	**Expected Results:**
1. Open the application	1. Tap on application icon	Home page is opened, and list of methods is not appeared

Test case ID: 3 **Test case Name: Methods can be selected from the list successfully**		
Summary: Make sure that the user can select the methods from the list appeared on the home page		
Preconditions: The application must be installed on a device and list of methods is available		
Step actions:	**Input values**	**Expected Results:**
1. Open the application 2. Select a method from the list	1. Tap on application icon 2. Tap on desire method name	Methods is selected successfully by the user

Test case ID: 4 **Test case Name: Method selection is failed**		
Summary: List of methods appeared on the home page, but user is unable to select the methods		
Preconditions: The application must be installed on a device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1. Open the application 2. Select a method from the list	1. Tap on application icon 2. Tap on desire method name	User in unable to select the method

Test case ID: 5 **Test case Name: Successfully redirected to the home page**		
Summary: Make sure that the user is redirected to the home page when the application is opened		
Preconditions: The application must be installed on a device		
Step actions:	**Input values**	**Expected Results:**
1. Open the application	1. Tap on application icon	Home page is opened

Test case ID: 6 **Test case Name: Redirection to the home page failed**		
Summary: When user opens the application home page is failed to open		
Preconditions: The application must be installed on a device		
Step actions:	**Input Values**	**Expected Results:**
1. Open the application	1. Tap on application icon	Failed to redirect on the home page

Test case ID: 7 **Test case Name: UI on different screen sizes**		
Summary: Make sure that UI of the application appears correctly on different screen sizes		
Preconditions: The application must be installed on different devices		
Step actions:	**Input Values**	**Expected Results:**
1. Open the application on different screens	1. Tap on application icon	UI appears correctly on all the screens

Test case ID: 8 **Test case Name: UI fails to load correctly on different screen sizes**		
Summary: UI not appearing correctly on different screen sizes		
Preconditions: The application must be installed on different devices		
Step actions:	**Input Values**	**Expected Results:**
1. Open the application on different screens	1. Tap on application icon	UI not appears correctly on different screen sizes

Test case ID: 9 **Test case Name: Correct list of methods**		
Summary: Make sure that the list of methods includes the required methods		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1. Open the application	1. Tap on application icon	List of methods contain all the required/defined methods

Test case ID: 10 **Test case Name: Incorrect list of methods**		
Summary: List of methods include irrelevant methods		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1. Open the application	1. Tap on application icon	List of methods contain methods that are irrelevant or may not require by the user

Test case ID: 11 **Test case Name: Able to select the methods**		
Summary: User should be able to select any of the method from the list		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	Method is selected and the method's interface is opened

Test case ID: 12 **Test case Name: Unable to select a method**		
Summary: User is unable to select the method from the list		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	User is unable to select the method

Test case ID: 13 **Test case Name: Method’s Interface appears**		
Summary: Make sure that method’s interface appears successfully against every method in the list		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	Interface against the selected method appears

Test case ID: 14 **Test case Name: Method’s interface doesn’t appear**		
Summary: Interface is not available against one or more methods in the list		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	Interface against the selected method doesn’t appear

Test case ID: 15

Test case Name: Able to provide input values

Summary:

Make sure that the user can provide input values in the interface appears against the selected method

Preconditions:

The application must be installed on device and list of methods is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Method's interface appears	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10	User can provide the input values for the method

Test case ID: 16

Test case Name: Unable to provide input values

Summary:

User is unable to provide input values for the method in the interface appeared against that method

Preconditions:

The application must be installed on device and list of methods is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Method's interface appears	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10	User is unable to provide the input values for the method selected

Test case ID: 17 **Test case Name: Method’s Interface appears**		
Summary: Make sure that method’s interface appears when the method is selected		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	Interface against the selected method appears

Test case ID: 18 **Test case Name: Method’s interface doesn’t appear**		
Summary: Method’s interface doesn’t appear on clicking the method in the list		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	Interface against the selected method doesn’t appear

Test case ID: 19 **Test case Name: UI of every method appears correctly on the device screen**		
Summary: Make sure that UI of every method appears correctly on the device		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	UI against the selected method appears correctly against the device being used

Test case ID: 20 **Test case Name: UI of the methods doesn’t appear correctly on the device screen**		
Summary: Make sure that UI of every method appears correctly on the device		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	UI against the selected method appears correctly against the device being used

Test case ID: 21 **Test case Name: Required input fields are available**		
Summary: Make sure that UI of every method contains the required input fields		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	UI against the selected method appears and contains the required input fields

Test case ID: 22 **Test case Name: Required input fields are not available**		
Summary: Input fields that are required for the methods to input the data are not available		
Preconditions: The application must be installed on device and list of methods is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	UI against the selected method appears but doesn’t contain the required input fields

Test case ID: 23 **Test case Name: User is able to input data**		
Summary: Make sure that the user can enter input values in the input fields		
Preconditions: Interface for the method is available and contain the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Interface is opened	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10	User can input values using the available input fields

Test case ID: 24 **Test case Name: User is unable to input data**		
Summary: User cannot enter input values in the input fields		
Preconditions: Interface for the method is available and contain the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Interface is opened	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10	User is unable to input values using the available input fields

Test case ID: 25 **Test case Name: Displaying an error message**		
Summary: Make sure that an error message is generated against an invalid input		
Preconditions: Interface for the method is available and contain the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Provide invalid input	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5 4. number of iterations = 0	An error message is generated

Test case ID: 26 **Test case Name: Failed to generate error message**		
Summary: Error message is not generated against an invalid input		
Preconditions: Interface for the method is available and contain the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Provide invalid input	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 0	An error message is not generated

Test case ID: 27 **Test case Name: Checking input values against the standard input format**		
Summary: Make sure that an error message is generated in case an invalid input is provided		
Preconditions: Interface for the method is available and contain the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Provide input values	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 1.5	Provided inputs are checked against the standard format and an error message is generated in case of invalid input

Test case ID: 28 **Test case Name: Failed to generate an error message upon checking input values against the standard input format**		
Summary: An error message is not generated in case of an invalid input		
Preconditions: Interface for the method is available and contain the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Provide input values	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 2.5	Provided inputs are checked against the standard format but an error message is not generated in case of invalid inputs

<table>
<tr><td colspan="3"><b>Test case ID: 29</b>
<b>Test case Name: Editing or updating the input values</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
Make sure that the user can edit or update the input values</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Interface for the method is available and contain the input fields</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Select a method
3.Provide input values
4.Re-enter the input values</td><td>1. Tap on application icon
2. Tap on desire method name
3. x = 3, y = 5,
4. number of iterations = 10</td><td>User can edit/update/re-enter the inputs values</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test case ID: 30</b>
<b>Test case Name: Unable to edit or update the input values</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
User is unable to edit or update the input values</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Interface for the method is available and contain the input fields</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Select a method
3.Provide input values
4.Re-enter the input values</td><td>1. Tap on application icon
2. Tap on desire method name
3. x = 3, y = 5,
4. number of iterations = 10</td><td>User is not allowed to edit/update/re-enter the inputs values</td></tr>
</table>

Test case ID: 31 **Test case Name: Method’s interface must include solve button**		
Summary: Make sure that the solve button is present on every method’s interface		
Preconditions: Interface for the method is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Methods interface is opened	1. Tap on application icon 2. Tap on desire method name	Solve button is available

Test case ID: 32 **Test case Name: Method’s interface doesn’t contain the solve button**		
Summary: Solve button is not available on one or more method’s interface		
Preconditions: Interface for the method is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Methods interface is opened	1. Tap on application icon 2. Tap on desire method name	Solve button is not available

<table>
<tr><td colspan="3"><b>Test case ID: 33</b>
<b>Test case Name: Solve button is working properly</b></td></tr>
<tr><td colspan="3"><b>Summary:</b>
Make sure that the solve button is present on every method’s interface and is working properly</td></tr>
<tr><td colspan="3"><b>Preconditions</b>:
Interface for the method is available and solve button is present</td></tr>
<tr><td><b>Step actions:</b></td><td><b>Input Values</b></td><td><b>Expected Results:</b></td></tr>
<tr><td>1.Open the application
2.Select a method
3.Input values
4.Click solve button</td><td>1. Tap on application icon
2. Tap on desire method name
3. x = 3, y = 5,
4. number of iterations = 10
5. Tap on the solve button</td><td>Solution for the method is displayed</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test case ID: 34</b>
<b>Test case Name: Solve button is not working properly</b></td></tr>
<tr><td colspan="3"><b>Summary:</b>
Solve button is present on every method’s interface but it is not working properly</td></tr>
<tr><td colspan="3"><b>Preconditions</b>:
Interface for the method is available and solve button is present</td></tr>
<tr><td><b>Step actions:</b></td><td><b>Input Values</b></td><td><b>Expected Results:</b></td></tr>
<tr><td>1.Open the application
2.Select a method
3.Input values
4.Click solve button</td><td>1. Tap on application icon
2. Tap on desire method name
3. x = 3, y = 5,
4. number of iterations = 10
5. Tap on the solve button</td><td>Solution for the method is not displayed</td></tr>
</table>

Test case ID: 35

Test case Name: Including solve button on method's interface

Summary:

Make sure that the solve button is present on every method's interface

Preconditions:

Interface for the method is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	Solve button is available on method's interface

Test case ID: 36

Test case Name: Solve button is not available on method's interface

Summary:

Solve button is not available on one or more method's interface

Preconditions:

Interface for the method is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method	1. Tap on application icon 2. Tap on desire method name	Solve button is not available

Test case ID: 37 **Test case Name: Redirected to the solution page on pressing the solve button**		
Summary: Make sure that the solve button redirects the user to the solution page containing the solution for the method chosen		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	User is redirected to the solution page for that specific method

Test case ID: 38 **Test case Name: User is not redirected to the solution page**		
Summary: Solve button is present on every method's interface but it is not working properly		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	Failed to redirect to the solution page

Test case ID: 39 **Test case Name: Displaying results for each method**		
Summary: On pressing the solve button result must be displayed for each method		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	Result is displayed successfully

Test case ID: 40 **Test case Name: Failed to display results for each method**		
Summary: On pressing the solve button results are not displayed for each method		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	Failed to display the results

Test case ID: 41 **Test case Name: Display results to the user based on the predefined format**		
Summary: Make sure that the result is displayed to the user in the defined form		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	Result is displayed successfully in the defined form

Test case ID: 42 **Test case Name: Results predefined format is not followed**		
Summary: Results are not displayed to the user according to the predefined format		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	Result failed to display in the defined form

Test case ID: 43 **Test case Name: Complete solution**		
Summary: Make sure that the complete result is displayed to the user		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	Complete solution is displayed to the user

Test case ID: 44 **Test case Name: Incomplete solution**		
Summary: Solution displayed to the user is incomplete		
Preconditions: Interface for the method is available and solve button is present		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button	An incomplete solution is displayed to the user

Test case ID: 45 **Test case Name: Availability of home option on the solution page**		
Summary: Make sure that the home option is available on the solution page and redirects the user to the home page		
Preconditions: Interface for solution page is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button 5.Solution page is opened	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button 6. Tap on the home option	Home option is available on the solution page and user is redirected to the home page on clicking the option

Test case ID: 46 **Test case Name: Home option on the solution page is unavailable**		
Summary: Home option is unavailable on the solution page or it doesn't redirect the user to the home page		
Preconditions: Interface for solution page is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Select a method 3.Input values 4.Click solve button 5.Solution page is opened	1. Tap on application icon 2. Tap on desire method name 3. x = 3, y = 5, 4. number of iterations = 10 5. Tap on the solve button 6. Tap on the home option	Home option is unavailable on the solution page or the user is not redirected to the home page on clicking the option

APPENDIX II

Test cases for CariGari

Test case ID: 1 **Test Case Name: GUI is working properly**		
Summary: Check the application's GUI to make sure that it is user friendly		
Preconditions: The application must have a GUI and it must be logged in		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.click the login button	1.Username: Ali 2.Password: 12345 3.Tap on login button	Home page is opened, and User Interface is working properly and is easy to understand and use

Test case ID: 2 **Test Case Name: GUI is not working properly**		
Summary: Check the application's GUI to make sure that it is user friendly		
Preconditions: The application must have a GUI and it must be logged in		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.click the login button	1.Username: Ali 2.Password: 12345 3.Tap on login button	Home page is opened, and UI is not working properly

Test case ID: 3 **Test Case Name: Valid Login**		
Summary: Check response when valid username and password is entered		
Preconditions: User is registered		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.click the login button	1.Username: Ali 2.Password: 12345 3.Tap on login button	User should be logged into the application and can access the system

Test case ID: 4 **Test Case Name: Invalid Login**		
Summary: check the response when invalid username or password is entered		
Preconditions: unregistered user or may be a wrong password		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.click the login button	1.Username Ali123 2.Password: 12345# 3.Tap on login button	User should not be logged into the system and an error message must be generated

Test case ID: 5 **Test Case Name: Register Successfully**		
Summary: Make sure to have a registration page		
Preconditions: The application should have a registration page and the required fields to enter the data		
Step actions:	**Input value**	**Expected Results:**
1.Open the application 2.Click on the register/signup button 3.Enter the required information 4.Click on the submit button	1.Tap on register button 2.Username: Usman 3.Email: usman@xyz.com 4.Password:12345 5.Confirm password:12345 6.Tap on submit button	User registered successfully and can login the system now

Test case ID: 6 **Test Case Name: Registration failed**		
Summary: Make sure to have a registration page		
Preconditions: The application should have a registration page and the required fields to enter the data		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the register/signup button 3.Enter the required information 4.Click on the submit button	1.Tap on register button 2.Username: Usman 3.Email: usman@xyz.com 4.Password:12345 5.Confirm password:12345 6.Tap on submit button	User registration failed

Test case ID: 7 **Test Case Name: Monthly report generation successfully**		
Summary: To verify the generation of monthly report for the vehicle		
Preconditions: Vehicle history is available		
Step actions:	**Input values**	**Expected Results:**
1.Open the vehicle history 2.Click on the report generation button	1.Tap on vehicle registration number 2.Tap on generate report button	Report is generated for the vehicle

Test case ID: 8 **Test Case Name: Monthly report generation failed**		
Summary: To verify the generation of monthly report for the vehicle		
Preconditions: Vehicle history is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the vehicle history 2.Click on the report generation button	1.Tap on vehicle registration number 2.Tap on generate report button	Report is not generated for the vehicle

Test case ID: 9 **Test Case Name: Emergency SOS generated successfully**		
Summary: Verification of emergency SOS generation		
Preconditions: Rescue and emergency contact details are entered		
Step actions:	**Input Values**	**Expected Results:**
1.Car collision detected	1. Collision signal 2.Contact number	Emergency SOS is generated to inform the emergency contacts

Test case ID: 10 **Test Case Name: Emergency SOS generation failed**		
Summary: Verification of emergency SOS generation		
Preconditions: Rescue and emergency contact details are entered		
Step actions:	**Input Values**	**Expected Results:**
1.Car collision detected	1. Collision signal 2.Contact number	Emergency SOS is not generated

Test case ID: 11 **Test Case Name: Monitoring the mileage and fuel consumption successfully**		
Summary: Verification of the system to make sure the mileage and fuel consumption is being monitored		
Preconditions: Device is connected to the vehicle using Bluetooth		
Step actions:	**Input Values**	**Expected Results:**
1.Enter the amount of fuel present in the tank 2.Enter the value of vehicle speedometer in km	1. Fuel: 10 litres 2. KM: 500	Mileage is calculated against the fuel consumption

Test case ID: 12 **Test Case Name: Monitoring of mileage and fuel consumption failed**		
Summary: Verification of the system to make sure the mileage and fuel consumption is being monitored		
Preconditions: Device is connected to the vehicle using Bluetooth		
Step actions:	**Input Values**	**Expected Results:**
1.Enter the amount of fuel present in the tank 2.Enter the value of vehicle speedometer in km	1. Fuel: 10 litres 2. KM: 500	Unable to monitor the fuel consumption and the mileage details

Test case ID: 13 **Test Case Name: Locking the vehicle successfully**		
Summary: Verification of the vehicle lock functionality		
Preconditions: Device is connected to the vehicle and a locking interface is available on the device		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the lock option 4.Press the Lock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the lock option 5. Tap on the lock button	Vehicle is locked

Test case ID: 14 **Test Case Name: Locking vehicle failed**		
Summary: Verification of the vehicle lock functionality		
Preconditions: Device is connected to the vehicle and a locking interface is available on the device		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the lock option 4.Press the Lock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the lock option 5. Tap on the lock button	Vehicle is not locked

Test case ID: 15

Test Case Name: Unlock the vehicle successfully

Summary:

Verification of the vehicle unlock functionality

Preconditions:

Device is connected to the vehicle and an unlocking interface is available on the device

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the unlock option 4.Press the unlock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the unlock option 5. Tap on the unlock button	Vehicle is unlocked

Test case ID: 16

Test Case Name: Unlocking of the vehicle failed

Summary:

Verification of the vehicle unlock functionality

Preconditions:

Device is connected to the vehicle and an unlocking interface is available on the device

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the unlock option 4.Press the unlock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the unlock option 5. Tap on the unlock button	Vehicle is still locked

Test case ID: 17 **Test Case Name: Starting the engine successfully**		
Summary: Verification of the system to start the engine		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the start engine interface 4.Press the start button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the start engine option 5. Tap on the start button	The engine is started, and user can drive the car

Test case ID: 18 **Test Case Name: Starting the engine failed**		
Summary: Verification of the system to start the engine		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the start engine interface 4.Press the start button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the start engine option 5. Tap on the start button	The engine is not started

Test case ID: 19 **Test Case Name: Stop the engine successfully**		
Summary: Verification of the system to stop the engine		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the stop engine interface 4.Press the stop button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the stop engine option 5. Tap on the stop button	The engine is stopped

Test case ID: 20 **Test Case Name: Stopping the engine failed**		
Summary: Verification of the system to stop the engine		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the stop engine interface 4.Press the stop button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the stop engine option 5. Tap on the stop button	The engine is not stopped

Test case ID: 21 **Test Case Name: Bluetooth Connectivity successful**		
Summary: Verification of the Bluetooth connectivity between vehicle and device		
Preconditions: Bluetooth device is installed in the vehicle and the vehicle should be in range		
Step actions:	**Input Values**	**Expected Results:**
1.Turn on the Bluetooth on the device and vehicle 2.On mobile device search for the vehicle Bluetooth 3.Connect the device's Bluetooth to the vehicle	1. Tap to on the Bluetooth 2. Tap on the name of device to connect the car with the mobile	Device is connected to the vehicle using Bluetooth

Test case ID: 22 **Test Case Name: Bluetooth Connectivity failed**		
Summary: Verification of the Bluetooth connectivity between vehicle and device		
Preconditions: Bluetooth device is installed in the vehicle and the vehicle should be in range		
Step actions:	**Input Values**	**Expected Results:**
1.Turn on the Bluetooth on the device and vehicle 2.On mobile device search for the vehicle Bluetooth 3.Connect the device's Bluetooth to the vehicle	1. Tap to on the Bluetooth 2. Tap on the name of device to connect the car with the mobile	Device is unable to connect to the vehicle using Bluetooth

Test case ID: 23 **Test Case Name: Controlling the vehicle using Bluetooth successfully**		
Summary: Verification of the vehicle control through device using Bluetooth		
Preconditions: Bluetooth device is installed in the vehicle the vehicle should be in range		
Step actions:	**Input Values**	**Expected Results:**
1.Turn on the Bluetooth on the device and vehicle 2.Connect the vehicle to the device using Bluetooth	1. Tap to on the Bluetooth 2. Tap on the name of device to connect the car with the mobile	Vehicle can be controlled using device by the user

Test case ID: 24 **Test Case Name: Controlling the vehicle using Bluetooth failed**		
Summary: Verification of the vehicle control through device using Bluetooth		
Preconditions: Bluetooth device is installed in the vehicle the vehicle should be in range		
Step actions:	**Input Values**	**Expected Results:**
1.Turn on the Bluetooth on the device and vehicle 2.Connect the vehicle to the device using Bluetooth	1. Tap to on the Bluetooth 2. Tap on the name of device to connect the car with the mobile	Vehicle can't be controlled using device

Test case ID: 25 **Test Case Name: View history of vehicle successfully**		
<u>Summary</u>: Verify that the system provides a history of the places visited by the vehicle		
<u>Preconditions</u>: Record for visited places is maintained		
<u>Step actions:</u>	**<u>Input Values</u>**	**<u>Expected Results:</u>**
1.Open the application 2.login 3.Go to the history 4.View the history	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option 5. Tap on the history button	History can be viewed by the user

Test case ID: 26 **Test Case Name: View history of vehicle failed**		
<u>Summary</u>: Verify that the system provides a history of the places visited by the vehicle		
<u>Preconditions</u>: Record for visited places is maintained		
<u>Step actions:</u>	**<u>Input Values</u>**	**<u>Expected Results:</u>**
1.Open the application 2.login 3.Go to the history 4.View the history	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option 5. Tap on the history button	History can't be viewed by the user

Test case ID: 27 **Test Case Name: Delete the history successfully**		
Summary: Verify that the history can be deleted		
Preconditions: History is available and maintained		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the history page 4.Select the option to delete the history	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option 5. Tap on the delete history button	History can be deleted by the user

Test case ID: 28 **Test Case Name: Deleting the history failed**		
Summary: Verify that the history can be deleted		
Preconditions: History is available and maintained		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the history page 4.Select the option to delete the history	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option 5. Tap on the delete history button	History can’t be deleted by the user

Test Case ID: 29 **Test Case Name: GUI dimensions are correct**		
Summary: checking the dimensions of the GUI elements like size, position, height and width		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.click the login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Dimensions of the elements should be correct

Test Case ID: 30 **Test Case Name: GUI dimensions are incorrect**		
Summary: Checking the dimensions of the GUI elements like size, position, height and width		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.click the login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Dimensions of the elements are incorrect correct

Test Case ID: 31 **Test Case Name: Error Messages displayed successfully**		
Summary: An error message must be generated in red colour whenever an error occurred		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter Invalid username or password 3.click login button	1. Tap on the application icon 2. Username: Ali123 3. Password: 12345@ 4. Tap the login button	An error message must be generated in red colour

Test Case ID: 32 **Test Case Name: Error Messages display failed**		
Summary: An error message is failed to display whenever an error is occurred		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter Invalid username or password 3.click login button	1. Tap on the application icon 2. Username: Ali123 3. Password: 12345@ 4. Tap the login button	An error message is not generated

Test Case ID: 33 **Test Case Name: Correct UI on different screens**		
Summary: Verify that the application’s GUI is working fine on the different screens having different sizes		
Preconditions: Application should be installed on different devices having different screen sizes		
Step actions:	**Input Values**	**Expected Results:**
1.install the application on different devices 2.open the application on different devices	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Application's GUI should be working fine on all the screen sizes

Test Case ID: 34 **Test Case Name: Incorrect UI on different screens**		
Summary: Application’s GUI is not working fine on the different screens having different sizes		
Preconditions: Application should be installed on different devices having different screen sizes		
Step actions:	**Input Values**	**Expected Results:**
1.install the application on different devices 2.open the application on different devices	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Application's GUI is not working fine on all the screen sizes

Test Case ID: 35 **Test Case Name: Readable font**		
Summary: verify that the font used is readable and easy to understand		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.enter username 3.enter password 4.click login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Font is readable and can be understood easily

Test Case ID: 36 **Test Case Name: Unreadable font**		
Summary: Font is unreadable and not so easy to understand		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.enter username 3.enter password 4.click login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Font is unreadable

<table>
<tr><td colspan="3"><b>Test Case ID: 37</b>
<b>Test Case Name: Elements alignment</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
verify that text, buttons, text fields, icons and other elements must be aligned and in proper place</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application must be opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.enter username
3.enter password
4.click the login button</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap the login button</td><td>All the text, buttons, icons and other elements are aligned and are properly placed</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 38</b>
<b>Test Case Name: Input fields are added</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
To verify that the input fields are available</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.open the application
2.login page is opened</td><td>1. Tap on the application icon</td><td>Fields are available to input username and password on the login page</td></tr>
</table>

Test Case ID: 39 **Test Case Name: Input fields are not added**		
Summary: Input fields are not added for the username and password		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.open the application 2.login page is opened	1. Tap on the application icon	Fields are not available to input username and password on the login page

Test Case ID: 40 **Test Case Name: Allowed to input data**		
Summary: Make sure that the input fields are working properly and allow the user to input the data		
Preconditions: Application is installed and login page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	User can enter the username and password in the input fields

<table>
<tr><td colspan="3"><b>Test Case ID: 41</b>
<b>Test Case Name: Not allowed to input data</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
Input fields do not allow the user to input the data</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and login page is opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Enter the username
3.Enter the password</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap the login button</td><td>User is unable to enter the username and password in the input fields</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 42</b>
<b>Test Case Name: Login button is working</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
Add a login button on the login page and make sure that the user is logged in when it is clicked</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is running and login page is opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Enter the username
3.Enter the password
4.Click the login button</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap the login button</td><td>user must be logged in successfully</td></tr>
</table>

Test Case ID: 43 **Test Case Name: Login button is not working properly**		
Summary: Login button is not working properly and do not enable the user to logged in the system		
Preconditions: Application is running and login page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.Click the login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	User is unable to logged in the system

Test Case ID: 44 **Test Case Name: verification of username and password**		
Summary: Make sure that the username and password must be verified on clicking the login button		
Preconditions: Login page is opened and username and password is entered.		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.Click the login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Username and password must be verified on clicking the login button and only registered user will be redirected to the home page

<table>
<tr><td colspan="3"><b>Test Case ID: 45</b>
<b>Test Case Name: verification of username and password failed</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
The username and password is not verified on clicking the login button</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Login page is opened and username and password is entered.</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Enter the username
3.Enter the password
4.Click the login button</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap the login button</td><td>Username and password is not verified on clicking the login button and only unregistered user will be redirected to the home page</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 46</b>
<b>Test Case Name: Successful login</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
User must enter the system successfully on providing the valid login credentials</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application must be opened and the user must be registered</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Enter the username
3.Enter the password
4.Click the login button</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap the login button</td><td>Username and password will be verified, and user will be successfully entered the system</td></tr>
</table>

Test Case ID: 47 **Test Case Name: Login failed**		
Summary: User is unable to login the system on providing the valid login credentials		
Preconditions: Application must be opened and the user must be registered		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.Click the login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Login credentials are valid but user is unable to logged in the system

Test Case ID: 48 **Test Case Name: login failed**		
Summary: Verify that an error message should be generated in case of invalid login credentials		
Preconditions: Application is running and the user is not registered		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the username 3.Enter the password 4.Click the login button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap the login button	Login failed and an error message is generated

Test Case ID: 49 **Test Case Name: Signup/register button**		
Summary: Make sure to add a signup or register button on the login page		
Preconditions: Login page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open application	1. Tap on the application icon	Login page appeared and signup/register button is available on the login page

Test Case ID: 50 **Test Case Name: Signup/register button not available**		
Summary: A signup or register button is not available on the login page		
Preconditions: Login page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open application	1. Tap on the application icon	Login page appeared and signup/register button is not available on the login page

Test Case ID: 51 **Test Case Name: signup form**		
Summary: User needs to be registered first, to login the system		
Preconditions: Application should be running, and the user must be unregistered		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the signup or registration button	1. Tap on the application icon 2. Tap on the register button	A registration page should be opened

Test Case ID: 52 **Test Case Name: Adding the fields on signup/register form**		
Summary: To verify that the input fields for registration are available		
Preconditions: Application is running and registration page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Click on the registration button	1. Tap on the application icon 2. Tap on the register button	Registration page is opened, and input fields are available

Test Case ID: 53 **Test Case Name: Input fields are working**		
Summary: Verify that the input fields for registration are working		
Preconditions: Registration page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Click on the signup /registration button 2.Input the data in registration form	1. Tap on the application icon 2. Username: Ali 3. Email: ali@xyz.com 4. Password: 12345 5. Confirm Password: 12345 6. Tap the register button	User can enter the data in input fields

Test Case ID: 54 **Test Case Name: verification of input data**		
Summary: Input data must be verified for registration		
Preconditions: Registration page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Click on the signup/registration button 2.Input the data in registration form	1. Tap on the application icon 2. Username: Ali 3. Email: ali@xyz.com 4. Password: 12345 5. Confirm Password: 12345 6. Tap the register button	The data entered by the user is verified and checked according to the constraints on every input field

Test Case ID: 55 **Test Case Name: Submit button**		
Summary: Verify that a submit button is placed on the registration page		
Preconditions: Registration page is opened		
Step actions:	**Input Values**	**Expected Results:**
Open the registration page	1.Tap on the application icon 2. Tap on the registration button	Submit button is available on the registration page

Test Case ID: 56 **Test Case Name: registered successfully**		
Summary: Make sure that the submit button on the registration page is working properly		
Preconditions: Registration page is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the registration page 2.Provide the required input data 3.Click the submit button	1. Tap on the application icon 2. Username: Ali 3. Email: ali@xyz.com 4. Password: 12345 5. Confirm Password: 12345 6. Tap the register button	Input details are submitted, and the user is registered successfully

Test Case ID: 57 **Test Case Name: Registration failed**		
Summary: Submit button on the registration page is not working properly		
Preconditions: Registration page is opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the submit page 2.Provide the required input data 3.Click the submit button	1. Tap on the application icon 2. Username: Ali 3. Email: ali@xyz.com 4. Password: 12345 5. Confirm Password: 12345 6. Tap the register button	Failed to register

Test Case ID: 58 **Test Case Name: Redirected to login page**		
Summary: Mae sure that the login button is present on the registration page and working properly		
Preconditions: Registration page is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the registration page 2.Click on the Login button	1. Tap on the application icon 2. Tap on the registration button 3. Tap on the login button	User is redirected to the login page

Test Case ID: 59

Test Case Name: Redirection to login page failed

Summary:

Login button is not present on the registration page or does not redirect the user to the login page

Preconditions:

Registration page is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the registration page 2.Click the registered button 3.Click on the Login button	1. Tap on the application icon 2. Tap on the registration button 3. Tap on the login button	User is not redirected to the login page

Test Case ID: 60

Test Case Name: Report generation option is available

Summary:

Verify that the application contains the report generation option

Preconditions:

History is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the history of the vehicle	1. Tap on the application icon 2. Tap on the history option	Report generation option is available on the history page.

Test Case ID: 61 **Test Case Name: Report generation option is not available**		
Summary: The application does not contain the report generation option		
Preconditions: History is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the history of the vehicle	1. Tap on the application icon 2. Tap on the history option	Report generation option is not available on the history page.

Test Case ID: 62 **Test Case Name: Complete report details**		
Summary: Verify the availability of the required information in the report generated		
Preconditions: History is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the history page 2.Click on the report generation button	1. Tap on the application icon 2. Tap on the history option 3. Tap on the report button	Report is generated containing details about fuel consumption, distance covered, and places visited

Test Case ID: 63 **Test Case Name: Incomplete report details**		
<u>**Summary**</u>: Unavailability of the required information in the report generated		
<u>**Preconditions**</u>: History is available		
<u>**Step actions:**</u>	<u>**Input Values**</u>	<u>**Expected Results:**</u>
1.Open the history page 2.Click on the report generation button	1. Tap on the application icon 2. Tap on the history option 3. Tap on the report button	The generated report does not contain all the required information

Test Case ID: 64 **Test Case Name: Emergency contact details added**		
<u>**Summary**</u>: Verification of emergency contact details		
<u>**Preconditions**</u>: An option to add emergency contact details		
<u>**Step actions:**</u>	<u>**Input Values**</u>	<u>**Expected Results:**</u>
1.Open the emergency contact option 2.Add contact details	1. Tap on the application icon 2. Tap on the add contact option 3. Name: Ali 4. Contact # 033xxxxxxxx 5. Tap on the add button	Emergency contact details are added successfully

Test Case ID: 65 **Test Case Name: Unable to add emergency contact details**		
Summary: User is unable to add emergency contact details		
Preconditions: An option to add emergency contact details		
Step actions:	**Input Values**	**Expected Results:**
1.Open the emergency contact option 2.Add contact details	1. Tap on the application icon 2. Tap on the add contact option 3. Name: Ali 4. Contact # 033xxxxxxxx 5. Tap on the add button	Failed to add emergency contact details

Test Case ID: 66 **Test Case Name: Emergency SOS generated**		
Summary: Verification of the generation of SOS		
Preconditions: Emergency contacts are added		
Step actions:	**Input Values**	**Expected Results:**
1.A collision is detected	1. Collision signal 2.Contact number	SOS is generated

Test Case ID: 67

Test Case Name: Emergency SOS failed

Summary:

Emergency SOS failed to generate

Preconditions:

Emergency contacts are added

Step actions:	**Input Values**	**Expected Results:**
1.A collision is detected	1. Collision signal 2.Contact number	SOS is not generated

Test Case ID: 68

Test Case Name: Sending SOS to specified contact

Summary:

Verify that the generated SOS is being sent to the contact specified

Preconditions:

Contact details are added

Step actions:	**Input Values**	**Expected Results:**
1.Collision is detected 2.SOS is generated	1. Collision signal 2.Contact number	SOS is sent to the specified contact

Test Case ID: 69

Test Case Name: Failed to send SOS to the specified contact

Summary:

The generated SOS is not being sent to the contact specified

Preconditions:

Contact details are added

Step actions:	**Input Values**	**Expected Results:**
1.Collision is detected 2.SOS is generated	1. Collision signal 2.Contact number	SOS is not sent to the specified contact

Test Case ID: 70

Test Case Name: Collision detection successful

Summary:

Verification of the collision detection

Preconditions:

Sensor is installed and device is connected to the vehicle using Bluetooth

Step actions:	**Input Values**	**Expected Results:**
1.Car collides 2.Sensor detects the collision	1. Collision signal 2.Contact number	Collision detected and an SOS is generated

Test Case ID: 71 **Test Case Name: Collision detection failed**		
Summary: Failed to detect the collision		
Preconditions: Sensor is installed and device is connected to the vehicle using Bluetooth		
Step actions:	**Input Values**	**Expected Results:**
1.Car collides 2.Sensor detects the collision	1. Collision signal 2.Contact number	Collision is not detected

Test Case ID: 72 **Test Case Name: Calculating the covered distance**		
Summary: Verification of the distance coverage		
Preconditions: Device is connected to the vehicle using Bluetooth		
Step actions:	**Input Values**	**Expected Results:**
1.Initial value of distance is added 2.Start the car and travel 3.Journey is completed	1. Initial Value: 35000 KM 2. Final Value: 35300 KM	Covered distance is calculated and displayed to the user

Test Case ID: 73 **Test Case Name: Failed to calculate the covered distance**		
Summary: Unable to calculate the covered distance		
Preconditions: Device is connected to the vehicle using Bluetooth		
Step actions:	**Input Values**	**Expected Results:**
1.Initial value of distance is added 2.Start the car and travel 3.Journey is completed	1. Initial Value: 35000 KM 2. Final Value: 35300 KM	System failed to calculate the covered distance

Test Case ID: 74 **Test Case Name: Fuel ending alert**		
Summary: Verification of the generation of alert when the fuel is about to end		
Preconditions: Device is connected to the vehicle using Bluetooth and fuel is in the tank		
Step actions:	**Input Values**	**Expected Results:**
1.Enter the amount of fuel present in the car tanks 2.Start the journey	1. Fuel: 10 Litres	An alert is generated when the fuel is about to end

Test Case ID: 75 **Test Case Name: Fuel ending alert failed**		
Summary: Failed to generate the alert when the fuel is about to end		
Preconditions: Device is connected to the vehicle using Bluetooth and fuel is in the tank		
Step actions:	**Input Values**	**Expected Results:**
1.Enter the amount of fuel present in the car tanks 2.Start the journey	1. Fuel: 10 Litres	System failed to generate the alert when the fuel is about to end

Test Case ID: 76 **Test Case Name: Estimation of distance**		
Summary: Verification of the distance estimation		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Enter the amount of fuel present 2.Enter the value for mileage of the car	1. Fuel: 10 Litres 2. Millage: 15KM	Estimated distance is calculated that can be covered

Test Case ID: 77 **Test Case Name: Lock button available**		
Summary: Verification of the lock button		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the vehicle lock interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the lock option	A lock button is present

Test Case ID: 78 **Test Case Name: Lock button not available**		
Summary: Lock button is not placed		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the vehicle lock interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the lock option	A lock button is not present in the app

Test Case ID: 79 **Test Case Name: Lock button works**		
Summary: Verify that the lock button is working		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the lock vehicle interface 4.Press the lock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the lock option 5. Tap on the lock button	Vehicle is locked

Test Case ID: 80 **Test Case Name: Lock button not working**		
Summary: The lock button is not working		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the lock vehicle interface 4.Press the lock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the lock option 5. Tap on the lock button	Lock button failed to lock the vehicle

Test Case ID: 81 **Test Case Name: Unlock button**		
Summary: Verification of the unlock button		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the vehicle unlock interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the unlock option	An unlock button is present

Test Case ID: 82 **Test Case Name: Unlock button not available**		
Summary: Unlock button is not available		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the vehicle lock interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the unlock option	Unlock button is not present in the app

Task 2: Make sure that the unlock button unlocks the vehicle’s doors

Test Case ID: 83 **Test Case Name: Unlock button works**		
Summary: Verify that the unlock button is working		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the unlock vehicle interface 4.Press the unlock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the unlock option 5. Tap on the unlock button	Vehicle is unlocked

Test Case ID: 84 **Test Case Name: Unlock button not working**		
Summary: The unlock button is not working		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the lock vehicle interface 4.Press the lock button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the unlock option 5. Tap on the unlock button	unlock button failed to lock the vehicle

Test Case ID: 85 **Test Case Name: Start button available**		
Summary: To verify the availability of start button		
Preconditions: Interface to start the vehicle is developed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the start engine interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the start option	Start button is present to start the engine

Test Case ID: 86 **Test Case Name: Start button is not available**		
Summary: Unavailability of start button		
Preconditions: Interface to start the vehicle is developed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the start engine interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the start option	Start button is not present to start the engine

<table>
<tr><td colspan="3"><b>Test Case ID: 87</b>
<b>Test Case Name: Engine started successfully</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
Verification of the start button functionality when it is pressed</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Device is connected to the vehicle</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.login
3.Go to the start engine interface
4.Press the start button</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap on the start option
5. Tap on the start button</td><td>The engine of the vehicle is started on pressing the start button</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 88</b>
<b>Test Case Name: Engine failed to start</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
Start button is not working properly</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Device is connected to the vehicle</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.login
3.Go to the start engine interface
4.Press the start button</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap on the start option
5. Tap on the start button</td><td>The engine of the vehicle failed to start on pressing the start button</td></tr>
</table>

Test Case ID: 89 **Test Case Name: Stop button available**		
Summary: To verify the availability of stop button		
Preconditions: Interface to start the vehicle is developed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the stop engine interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the stop option	Stop button is present to stop the engine

Test Case ID: 90 **Test Case Name: Stop button unavailable**		
Summary: To verify the unavailability of stop button		
Preconditions: Interface to start the vehicle is developed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the stop engine interface	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the stop option	Stop button is not available to stop the engine

Test Case ID: 91 **Test Case Name: Engine stopped successfully**		
Summary: Verification of the stop button functionality when it is pressed		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the stop engine interface 4.Press the stop button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the stop option 5. Tap on the stop button	The engine of the vehicle is stopped.

Test Case ID: 92 **Test Case Name: Failed to stop the engine**		
Summary: Stop button not working properly when it is pressed		
Preconditions: Device is connected to the vehicle		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the stop engine interface 4.Press the stop button	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the stop option 5. Tap on the stop button	The engine of the vehicle failed to stop.

Test Case ID: 93 **Test Case Name: Controlling the vehicle**		
Summary: Verification of the vehicle control through device using Bluetooth		
Preconditions: Bluetooth device is installed in the vehicle the vehicle should be in range		
Step actions:	**Input Values**	**Expected Results:**
1.Turn on the Bluetooth on the device and vehicle 2.Connect the vehicle to the device using Bluetooth	1. Tap to on the Bluetooth 2. Tap on the name of device to connect the car with the mobile	Vehicle can be controlled using device

Test Case ID: 94 **Test Case Name: Bluetooth visibility**		
Summary: To verify the Bluetooth visibility of vehicle to the device		
Preconditions: Bluetooth device is installed in the vehicle the vehicle should be in range		
Step actions:	**Input Values**	**Expected Results:**
1.Turn on the Bluetooth on vehicle and the device 2.Search for the vehicle’s Bluetooth	1. Tap to on the Bluetooth	Vehicle’s Bluetooth is visible to the device

Test Case ID: 95 **Test Case Name: History option**		
Summary: Verify that an option or interface is added to view the history of visited places		
Preconditions: History is available and maintained		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon 2. Username: Ali 3. Password: 12345	An option/interface to view the history of visited places is available

Test Case ID: 96 **Test Case Name: Visiting places details**		
Summary: To verify that the history contains the details for all the places visited by the vehicle		
Preconditions: Record for visited places is maintained		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.login 3.Go to the history page 4.View the history	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option	The history contains complete details about the visited places

Test Case ID: 97 **Test Case Name: Availability of delete history option**		
Summary: Verify the availability of the delete history option		
Preconditions: History is maintained and an interface for history is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Login 3.Go to the history	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option	Option to delete the history is available

Test Case ID: 98 **Test Case Name: Confirmation of deletion**		
Summary: Verify that the history is deleted when the delete history option is used		
Preconditions: History is maintained and an interface for history is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Login 3.Go to history 4.Delete the history	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option 5. Tap on the delete history button	History is deleted successfully

Test Case ID: 99 **Test Case Name: Message on deleting history**		
Summary: Verification of message generation upon deletion of history successfully		
Preconditions: History is deleted		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Login 3.Go to history 4.Delete the history 5.History is deleted	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option 5. Tap on the delete history button	A “deleted successfully” message is generated on the screen

Test Case ID: 100 **Test Case Name: Unauthorized user login**		
Summary: Unauthorized users are able to login		
Preconditions: Login page is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter invalid username and password 3.Press login button	1. Tap on the application icon 2. Username: Ali123 3. Password: 12345# 4.Tap the login button	Unauthorized user is logged in the system

<table>
<tr><td colspan="3"><b>Test Case ID: 101</b>
<b>Test Case Name: Editing the monthly report</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
User is able to edit the monthly report of the vehicle to change the record</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Monthly reports are available</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Open the report option
3.Click on the generate report button
4.Report is generated</td><td>1. Tap on the application icon
2. Username: Ali
3. Password: 12345
4. Tap on the report option</td><td>User can edit the report generated by the application to change the record or to create some discrepancy in the record</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 102</b>
<b>Test Case Name: Emergency contact details changed by the unauthorized user</b></td></tr>
<tr><td colspan="3"><b><u>Summary</u></b>:
Unauthorized user is able to change the contact details of an authorized user</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Contacts details are added, and unauthorized user is logged in</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Login the system using invalid credentials
3.Open the emergency contacts details of a user</td><td>1. Tap on the application icon
2. Username: Ali123
3. Password: 12345$$
4. Tap on the contact details</td><td>An unauthorized user is enabled to change the emergency contacts details of a registered user</td></tr>
</table>

Test Case ID: 103 **Test Case Name: Change the history details**		
Summary: User is able to change the history details of the visited places		
Preconditions: History is maintained by the application		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Login 3.Open the history details	1. Tap on the application icon 2. Username: Ali 3. Password: 12345 4. Tap on the history option	User is able to edit the history details to manipulate the record

APPENDIX III

Test cases for Emergency Response

Test Case ID: 1 **Test Case Name: GUI is working properly**		
Summary: Check the application's GUI to make sure that it is user friendly and is working properly		
Preconditions: The application must have a GUI		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Navigate through the application	1. Tap on the application icon	Home page is opened, User Interface is working properly and is easy to understand and navigate the user

Test Case ID: 2 **Test Case Name: GUI is not working properly**		
Summary: Check the application's GUI to make sure that it is user friendly and is working properly		
Preconditions: The application must have a GUI		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Navigate through the application	1. Tap on the application icon	Home page is opened, User Interface is not working properly and is difficult to understand and navigate the user

<table>
<tr><td colspan="3"><b>Test Case ID: 3</b>
<b>Test Case Name: GUI dimensions are correct</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
checking the dimensions of the GUI elements like size, position, height and width</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Navigate through the application</td><td>1. Tap on the application icon</td><td>Dimensions of the elements should be correct</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 4</b>
<b>Test Case Name: GUI dimensions are incorrect</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
checking the dimensions of the GUI elements like size, position, height and width</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Navigate through the application</td><td>1. Tap on the application icon</td><td>Dimensions of the elements are incorrect</td></tr>
</table>

Test Case ID: 5 **Test Case Name: Error Messages displayed successfully**		
Summary: An error message must be generated whenever an error occurred		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Add a contact without entering the contact details	1. Tap on the application icon 2. Name: Ali 3.Contact number: empty 4. Tap on the add button	An error message must be generated

Test Case ID: 6 **Test Case Name: Failed to display an error message**		
Summary: An error message is failed to display whenever an error occurs		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Add a contact without entering the contact details	1. Tap on the application icon 2. Name: Ali 3.Contact number: empty 4. Tap on the add button	An error message is not generated

Test Case ID: 7

Test Case Name: Correct UI on different screens

Summary:

Verify that the application's GUI is working fine on the different screens having different sizes

Preconditions:

Application should be installed on different devices having different screen sizes

Step actions:	**Input Values**	**Expected Results:**
1.Install the application on different devices 2.Open the application on different devices	1. Tap on the application icon	Application's GUI should be working fine on all the screen sizes

Test Case ID: 8

Test Case Name: Incorrect UI on different screens

Summary:

Application's GUI is not working fine on the different screens having different sizes

Preconditions:

Application should be installed on different devices having different screen sizes

Step actions:	**Input Values**	**Expected Results:**
1.Install the application on different devices 2.Open the application on different devices	1. Tap on the application icon	Application's GUI is not working fine on all the screen sizes

Test Case ID: 9 **Test Case Name: Readable font**		
Summary: verify that the font used is readable and easy to understand		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Font is readable and can be understood easily

Test Case ID: 10 **Test Case Name: Unreadable font**		
Summary: Font is unreadable and not so easy to understand		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Font is unreadable

Test Case ID: 11 **Test Case Name: Elements aligned properly**		
Summary: verify that text, buttons, text fields, icons and other elements must be aligned and in proper place		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	All the text, buttons, icons and other elements are aligned and are properly placed

Test Case ID: 12 **Test Case Name: Elements are not aligned properly**		
Summary: text, buttons, text fields, icons and other elements are not aligned properly		
Preconditions: Application must be opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	All the text, buttons, icons and other elements are not aligned properly

<table>
<tr><td colspan="3"><b>Test Case ID: 13</b>
<b>Test Case Name: Able to manage contacts</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
User is able to add, update and delete the contacts</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Manage contacts option is available on the application home screen</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Click on the manage contacts option</td><td>1. Tap on the application icon
2. Tap on the manage contacts option</td><td>User should be able to manage the contacts to use them in case of any emergency</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 14</b>
<b>Test Case Name: Unable to manage contacts</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
User is unable to manage the contacts</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Manage contacts option is available on the application home screen</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Click on the manage contacts option</td><td>1. Tap on the application icon
2. Tap on the manage contacts option</td><td>User is unable to manage the contacts or to open the manage contacts option</td></tr>
</table>

Test Case ID: 15 **Test Case Name: Adding an option to manage contacts**		
Summary: To verify that the manage contacts option is available in the application		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Manage contacts option/button is available on the application home screen

Test Case ID: 16 **Test Case Name: Manage contacts option not available**		
Summary: To verify the availability of manage contacts option in the application		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Manage contacts option not available

Test Case ID: 17

Test Case Name: Availability of add, update and delete options

Summary:

To check the availability of add, update and delete options in the manage contact

Preconditions:

Manage contacts option is available on the application home screen

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contact option	1. Tap on the application icon 2. Tap on the manage contacts option	Add, update and delete options are available

Test Case ID: 18

Test Case Name: Unavailability of add, update and delete options

Summary:

To check the unavailability of add, update and delete options in the manage contacts

Preconditions:

Manage contacts option is available on the application home screen

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contact option	1. Tap on the application icon 2. Tap on the manage contacts option	Add, update and delete options are not available

<table>
<tr><td colspan="3"><b>Test Case ID: 19</b>
<b>Test Case Name: Adding the input fields</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
To verify that the input fields are available</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Click on the manage contacts option</td><td>1. Tap on the application icon
2. Tap on the manage contacts option
3. Tap on add new contacts button</td><td>Fields are available for name, contact number and email to add a new contact</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 20</b>
<b>Test Case Name: Input fields are not added</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
input fields are not added</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and opened</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Click on the manage contact option</td><td>1. Tap on the application icon
2. Tap on the manage contacts option
3. Tap on add new contacts button</td><td>Input fields are not available</td></tr>
</table>

Test Case ID: 21 **Test Case Name: Add contact button availability**		
Summary: Make sure that the Add contact button is available		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts option	1. Tap on the application icon 2. Tap on the manage contacts option	Add contact button is available

Test Case ID: 22 **Test Case Name: Unavailability of the "Add contact button"**		
Summary: Add contact button is not available on the contact manage page		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the contact manage option	1. Tap on the application icon 2. Tap on the manage contacts option	Add contact button is not available

Test Case ID: 23 **Test Case Name: Allow input data**		
Summary: Make sure that the input fields are working properly and allow the user to input the data		
Preconditions: Application is installed, opened and input fields are added		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Enter the data in the input fields	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on add new contacts button 4. Name: Ali 5. Contact #: 033xxxxxxxx	User is able to provide the data using the input fields

Test Case ID: 24 **Test Case Name: Input data not allowed**		
Summary: User is unable to enter the data in the input fields		
Preconditions: Application is installed, opened and input fields are added		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Enter the data in the input fields	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on add new contacts button 4. Name: Ali 5. Contact #: 033xxxxxxxx	User is unable to enter the data in the input fields

Test Case ID: 25 **Test Case Name: “Add contact” button is working**		
Summary: Make sure that the contacts are added when the button is pressed		
Preconditions: Contact details are added in the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Enter the data in the input fields 4.Press the “Add contact” button	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on add new contacts button 4. Name: Ali 5. Contact #: 033xxxxxxxx 6. Tap on the add button	Contact details are added successfully

Test Case ID: 26 **Test Case Name: “Add contact” button is not working**		
Summary: Failed to add the contact when the button is pressed		
Preconditions: Contact details are added in the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Enter the data in the input fields 4.Press the “Add contact” button	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on add new contacts button 4. Name: Ali 5. Contact #: 033xxxxxxxx 6. Tap on the add button	Unable to add the contact details

Test Case ID: 27 **Test Case Name: Update button availability**		
Summary: Make sure that the update button is available		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts option	1. Tap on the application icon 2. Tap on the manage contacts option	Update button is available

Test Case ID: 28 **Test Case Name: Update button unavailability**		
Summary: Update button is not available on the manage contacts page		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts option	1. Tap on the application icon 2. Tap on the manage contacts option	Update button is not available

Test Case ID: 29 **Test Case Name: Update button is working successfully**		
Summary: Make sure that the contacts are updated when the button is pressed		
Preconditions: Contact details are updated in the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Enter the updated data in the input fields 3.Press the update button	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on update contacts button 4. Name: Umer 5. Contact #: 0331xxxxxxx 6. Tap on the update button	Contact details are updated successfully

Test Case ID: 30 **Test Case Name: Update button is not working**		
Summary: Failed to update the contact when the button is pressed		
Preconditions: Contact details are added in the input fields		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Enter the data in the input fields 4.Press the update button	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on update contacts button 4. Name: Umer 5. Contact #: 0331xxxxxxx 6. Tap on the update button	Unable to update the contact details

Test Case ID: 31 **Test Case Name: Delete button availability**		
Summary: Make sure that the delete button is available		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts option 3.Go to contact details	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on the contact details	Delete button is available to delete the contacts

Test Case ID: 32 **Test Case Name: Delete button unavailability**		
Summary: Delete button is not available on the manage contacts page		
Preconditions: Application is installed and opened		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts option 3.Go to the contact details	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on the contact details	Delete button is not available

Test Case ID: 33 **Test Case Name: Delete button is working successfully**		
Summary: Make sure that the contact details are deleted when the button is pressed		
Preconditions: Contact is selected		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Go to contact details 4.Select a contact 5.Press the delete button	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on the contact details 4. Tap on the contact 5. Tap on the delete button	Contact details are deleted successfully

Test Case ID: 34 **Test Case Name: Delete button is not working**		
Summary: Failed to delete the contact when the button is pressed		
Preconditions: Contact is selected		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the manage contacts 3.Go to contact details 4.Select a contact 5.Press the delete button	1. Tap on the application icon 2. Tap on the manage contacts option 3. Tap on the contact details 4. Tap on the contact 5. Tap on the delete button	Unable to delete the contact details

Test Case ID: 35 **Test Case Name: Fetching the current location of the device**		
Summary: The application should be able to fetch the location of the device using GPS		
Preconditions: Internet and GPS services of the device must be enabled		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the current location option	1. Tap on the application icon 2. Tap on the current location option	Device's current location is fetched successfully

Test Case ID: 36 **Test Case Name: Failed to fetch the current location of the device**		
Summary: The application is failed to fetch the location of the device using GPS		
Preconditions: Internet and GPS services of the device must be enabled		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the current location option	1. Tap on the application icon 2. Tap on the current location option	Unable to fetch the current location

Test Case ID: 37

Test Case Name: Availability of the current location option

Summary:

Current location option should be available on the home screen to fetch the current location of device

Preconditions:

Application is installed on the device and home screen is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Current location option is available on the home screen

Test Case ID: 38

Test Case Name: Unavailability of the current location option

Summary:

Current location option is not available on the home screen to fetch the current location of device

Preconditions:

Application is installed on the device and home screen is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Current location option is not available on the home screen

Test Case ID: 39

Test Case Name: Correct location

Summary:

Make sure that the current location option is able to fetch the correct location of the device

Preconditions:

Application is installed and GPS is enabled

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the current location option	1. Tap on the application icon 2. Tap on the current location option	The application is fetching the correct location of the device based on the GPS

Test Case ID: 40

Test Case Name: Incorrect location

Summary:

The current location option is unable to fetch the correct location of the device

Preconditions:

Application is installed and GPS is enabled

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the current location option	1. Tap on the application icon 2. Tap on the current location option	The application is failed to fetch the correct location of the device based on the GPS

Test Case ID: 41 **Test Case Name: Asking to enable the GPS**		
Summary: The application must ask the user to enable the GPS in case the GPS is disabled while fetching the current location		
Preconditions: Application is installed and GPS is disabled on the device		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the current location option	1. Tap on the application icon 2. Tap on the current location option	User will be asked to enable the GPS first

Test Case ID: 42 **Test Case Name: Failed to ask user to enable the GPS**		
Summary: The application failed to ask the user to enable the GPS in case the GPS is disabled while fetching the current location		
Preconditions: Application is installed and GPS is disabled on the device		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the current location option	1. Tap on the application icon 2. Tap on the current location option	User won’t be asked to enable the GPS

<table>
<tr><td colspan="3"><b>Test Case ID: 43</b>
<b>Test Case Name: GUI for location is available</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
Make sure to add a GUI for the current location to show the location</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and current location option is available</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Click on the current location option</td><td>1. Tap on the application icon
2. Tap on the current location option</td><td>GUI for the current location is available</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 44</b>
<b>Test Case Name: GUI for location is not available</b></td></tr>
<tr><td colspan="3"><b><u>Summary:</u></b>
GUI for the current location is not available</td></tr>
<tr><td colspan="3"><b><u>Preconditions</u></b>:
Application is installed and current location option is available</td></tr>
<tr><td><b><u>Step actions:</u></b></td><td><b><u>Input Values</u></b></td><td><b><u>Expected Results:</u></b></td></tr>
<tr><td>1.Open the application
2.Click on the current location option</td><td>1. Tap on the application icon
2. Tap on the current location option</td><td>GUI for the current location is not available</td></tr>
</table>

Test Case ID: 45 **Test Case Name: Calculate the optimum time, distance and path successfully**		
Summary: Make sure that the application is able to calculate the optimum time, distance and path so that an emergency message can be generated in case user won't make it in time to the destination		
Preconditions: Application is installed and emergency message option is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	Optimum time, distance and path is successfully calculated

Test Case ID: 46 **Test Case Name: Failed to calculate the optimum time, distance and path**		
Summary: The application is unable to calculate the optimum time, distance and path		
Preconditions: Application is installed and emergency message option is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	Optimum time, distance and path cannot be calculated

Test Case ID: 47 **Test Case Name: Availability of scheduled emergency SMS option**		
Summary: Scheduled emergency SMS option should be available to generate a scheduled SMS		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Scheduled emergency SMS option is available on the home screen

Test Case ID: 48 **Test Case Name: Unavailability of scheduled emergency SMS option**		
Summary: Scheduled emergency SMS option is unavailable on the home screen		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Scheduled emergency SMS option is not available on the home screen

Test Case ID: 49

Test Case Name: Availability of “distance finder”

Summary:

A button named “distance finder” should be available in scheduled emergency SMS option to calculate the distance, path and time

Preconditions:

Application is installed and scheduled emergency SMS option is available

Step actions:	**Input Values**	**Expected Results:**
1.Option the application 2.Click on the scheduled emergency SMS option	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option	Distance finder button is available

Test Case ID: 50

Test Case Name: Unavailability of “distance finder”

Summary:

“Distance finder” is not available in scheduled emergency SMS option

Preconditions:

Application is installed and scheduled emergency SMS option is available

Step actions:	**Input Values**	**Expected Results:**
1.Option the application 2.Click on the scheduled emergency SMS option	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option	Distance finder button is not available

Test Case ID: 51 **Test Case Name: Distance finder is working**		
Summary: Distance finder button should be working, and a map should be displayed to the user with its current location		
Preconditions: Distance finder is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder button	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	A map with user's current location should be displayed

Test Case ID: 52 **Test Case Name: Distance finder is not working**		
Summary: Distance finder button is not working, and unable to display the map to the user with its current location		
Preconditions: Distance finder is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder button	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	Distance finder's not working

Test Case ID: 53 **Test Case Name: Destination selection availability**		
Summary: Distance finder should also allow the user to select a destination after displaying a map upon clicking		
Preconditions: Distance finder is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder button	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	A map with the current location should be displayed and user should be asked for the destination

Test Case ID: 54 **Test Case Name: Destination selection unavailability**		
Summary: Distance finder does not allow the user to select a destination after displaying a map upon clicking		
Preconditions: Distance finder is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder button	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	Distance finder won’t ask the user about the destination

Test Case ID: 55 **Test Case Name: Calculating optimum time, path and distance successfully**		
Summary: Make sure that the distance finder can calculate the optimum path, time and distance successfully		
Preconditions: Distance finder is available along with the current location and destination		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder button 4.Enter the destination	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	Application should calculate the optimum path, time and distance successfully

Test Case ID: 56 **Test Case Name: Calculating optimum time, path and distance failed**		
Summary: Distance finder failed to calculate the optimum path, time and distance		
Preconditions: Distance finder is available along with the current location and destination		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option 3.Click on the distance finder button 4.Enter the destination	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button	Failed to calculate the optimum path, time and distance

Test Case ID: 57 **Test Case Name: SMS generation successful**		
Summary: The application should generate scheduled emergency SMS if the user can't reach the destination within the estimated time		
Preconditions: Distance finder is available along with the current location and destination		
Step actions:	**Input Values**	**Expected Results:**
1.Click on the distance finder button 2.Enter the destination 3.Start moving towards your destination	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button 4. set destination 5. Start travelling	A scheduled emergency SMS should be generated when the user can't reach the destination

Test Case ID: 58 **Test Case Name: SMS generation failed**		
Summary: The application is failed to generate scheduled emergency SMS if the user can't reach the destination within the estimated time		
Preconditions: Distance finder is available along with the current location and destination		
Step actions:	**Input Values**	**Expected Results:**
1.Click on the distance finder button 2.Enter the destination 3.Start moving towards your destination	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option 3. Tap on the distance finder button 4. set destination 5. Start travelling	A scheduled emergency SMS is not generated when the user can't reach the destination

Test Case ID: 59

Test Case Name: Set duration time for a scheduled emergency SMS

Summary:

User should be able to manually set the duration for the scheduled SMS

Preconditions:

Scheduled emergency SMS option is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option	User is able to manually set the duration of the scheduled SMS

Test Case ID: 60

Test Case Name: Failed to set duration time for a scheduled emergency SMS

Summary:

User is unable to manually set the duration for the scheduled SMS

Preconditions:

Scheduled emergency SMS option is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option	User is unable to manually set the duration of the scheduled SMS

Test Case ID: 61

Test Case Name: Activation or deactivation of the scheduled emergency SMS

Summary:

User should be able to manually activate or deactivate the scheduled emergency SMS

Preconditions:

Scheduled emergency SMS option is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option	User is able to manually activate or deactivate the scheduled emergency SMS

Test Case ID: 62

Test Case Name: Unable to activate or deactivate the scheduled emergency SMS

Summary:

User is unable to manually activate or deactivate the scheduled emergency SMS

Preconditions:

Scheduled emergency SMS option is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the scheduled emergency SMS option	1. Tap on the application icon 2. Tap on the scheduled emergency SMS option	User is unable to manually activate or deactivate the scheduled emergency SMS

Test Case ID: 63 **Test Case Name: Sending Email and SMS successfully**		
<u>Summary:</u> The application should allow the user to send an Email and SMS to all the added contacts		
<u>Preconditions</u>: At least one contact number and email address should be added in the application		
<u>Step actions:</u>	**<u>Input Values</u>**	**<u>Expected Results:</u>**
1.Open the application 2.Click on emergency SMS and Email option	1. Tap on the application icon 2. Tap on emergency SMS and Email option	Email and SMS send to all contacts added in application and message shown on screen ("Message sent successful!").

Test Case ID: 64 **Test Case Name: Sending Email and SMS failed**		
<u>Summary:</u> The application does not allow the user to send an Email and SMS to all the added contacts		
<u>Preconditions</u>: At least one contact number and email address should be added in the application		
<u>Step actions:</u>	**<u>Input Values</u>**	**<u>Expected Results:</u>**
1.Open the application 2.Click on emergency SMS and Email option	1. Tap on the application icon 2. Tap on emergency SMS and Email option	Failed to send the email and SMS

Test Case ID: 65 **Test Case Name: Availability of emergency SMS and email option**		
Summary: Emergency SMS and Email option should be available on the application home screen		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Emergency SMS and Email option is available on the home screen

Test Case ID: 66 **Test Case Name: Unavailability of emergency SMS and email option**		
Summary: Emergency SMS and Email option is not available on the application home screen		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Emergency SMS and Email option is not available on the home screen

Test Case ID: 67

Test Case Name: Asking the user before sending the SMS and Email

Summary:

Application should ask for the user confirmation before sending the emergency SMS and Email

Preconditions:

Application is installed

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency SMS and Email option	1. Tap on the application icon 2. Tap on emergency SMS and Email option	A dialog box is opened to ask the user for confirmation to send SMS and Email

Test Case ID: 68

Test Case Name: Failed to ask the user before sending the SMS and Email

Summary:

Application is not asking for the user confirmation before sending the emergency SMS and Email

Preconditions:

Application is installed

Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency SMS and Email option	1. Tap on the application icon 2. Tap on emergency SMS and Email option	User won’t be asked for the confirmation

<table>
<tr><td colspan="3"><b>Test Case ID: 69</b>
<b>Test Case Name: Sending the SMS and Email successfully</b></td></tr>
<tr><td colspan="3"><b>Summary:</b>
Application should send the emergency SMS and Email when user pressed the send button on the confirmation dialog box</td></tr>
<tr><td colspan="3"><b>Preconditions</b>:
Confirmation dialog box is available
Contact number and Email address is available</td></tr>
<tr><td><b>Step actions:</b></td><td><b>Input Values</b></td><td><b>Expected Results:</b></td></tr>
<tr><td>1.Open the application
2.Click on the emergency SMS and Email option
3.Press the send button on the confirmation dialog box</td><td>1. Tap on the application icon
2. Tap on emergency SMS and Email option
3. Tap on the send button</td><td>Emergency SMS and Email is sent successfully</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 70</b>
<b>Test Case Name: Failed to send the SMS and Email</b></td></tr>
<tr><td colspan="3"><b>Summary:</b>
Application is unable to send the emergency SMS and Email when user pressed the send button on the confirmation dialog box</td></tr>
<tr><td colspan="3"><b>Preconditions</b>:
Confirmation dialog box is available
Contact number and Email address is available</td></tr>
<tr><td><b>Step actions:</b></td><td><b>Input Values</b></td><td><b>Expected Results:</b></td></tr>
<tr><td>1Open the application
2.Click on the emergency SMS and Email option
3.Press the send button on the confirmation dialog box</td><td>1. Tap on the application icon
2. Tap on emergency SMS and Email option
3. Tap on the send button</td><td>Emergency SMS and Email not sent</td></tr>
</table>

Test Case ID: 71 **Test Case Name: Cancelling the emergency SMS and Email successfully**		
Summary: Application should cancel the emergency SMS and Email when user pressed the cancel button on the confirmation dialog box		
Preconditions: Confirmation dialog box is available Contact number and Email address is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency SMS and Email option 3.Press the cancel button on the confirmation dialog box	1. Tap on the application icon 2. Tap on emergency SMS and Email option 3. Tap on the cancel button	Emergency SMS and Email is cancelled

Test Case ID: 72 **Test Case Name: Failed to cancel the emergency SMS and Email**		
Summary: Application is unable to cancel the emergency SMS and Email when user pressed the cancel button on the confirmation dialog box		
Preconditions: Confirmation dialog box is available Contact number and Email address is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency SMS and Email option 3.Press the cancel button on the confirmation dialog box	1. Tap on the application icon 2. Tap on emergency SMS and Email option 3. Tap on the cancel button	Emergency SMS and Email not cancelled

Test Case ID: 73 **Test Case Name: Successful delivery message displayed**		
Summary: Application should display a message of success upon clicking the send button		
Preconditions: Confirmation dialog box is available Contact number and Email address is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency SMS and Email option 3.Press the send button on the confirmation dialog box	1. Tap on the application icon 2. Tap on emergency SMS and Email option 3. Tap on the send button	Successful delivery message is displayed

Test Case ID: 74 **Test Case Name: Successful delivery message not displayed**		
Summary: Application is unable to display a message of successful delivery upon clicking the send button		
Preconditions: Confirmation dialog box is available Contact number and Email address is available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency SMS and Email option 3.Press the send button on the confirmation dialog box	1. Tap on the application icon 2. Tap on emergency SMS and Email option 3. Tap on the send button	Successful delivery message is displayed

Test Case ID: 75

Test Case Name: Failure message displayed

Summary:

Application should display a message of failure upon clicking the send button if the SMS and Email is failed to deliver

Preconditions:

Confirmation dialog box is available

Contact number and Email address is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	
2.Click on the emergency SMS and Email option	2. Tap on emergency SMS and Email option	Failure message is displayed
3.Press the send button on the confirmation dialog box	3. Tap on the send button	

Test Case ID: 76

Test Case Name: Failure message not displayed

Summary:

Application is unable to display a message of failure upon clicking the send button if the SMS and Email is failed to deliver

Preconditions:

Confirmation dialog box is available

Contact number and Email address is available

Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	
2.Click on the emergency SMS and Email option	2. Tap on emergency SMS and Email option	Failure message not displayed
3.Press the send button on the confirmation dialog box	3. Tap on the send button	

Test Case ID: 77 **Test Case Name: Generating an emergency call successfully**		
Summary: To call the predefined person		
Preconditions: Emergency call option is available Contact details are added a contact has been specified to call in case of emergency		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency call option	1. Tap on the application icon 2. Tap on emergency call option	An emergency call should be generated to the specified person

Test Case ID: 78 **Test Case Name: Generating an emergency call failed**		
Summary: Fail to call the specified person		
Preconditions: Emergency call option is available Contact details are added a contact has been specified to call in case of emergency		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency call option	1. Tap on the application icon 2. Tap on emergency call option	Application is unable to call the specified person

Test Case ID: 79 **Test Case Name: Availability of emergency call option**		
Summary: Emergency call option should be available on the application home screen		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Emergency call option is available on the home screen

Test Case ID: 80 **Test Case Name: Unavailability of emergency call option**		
Summary: Emergency call option is not available on the application home screen		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	Emergency call option is not available on the home screen

Test Case ID: 81 **Test Case Name: Asking the user before initiating the call**		
Summary: Application should ask for the user confirmation before initiating the call		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency call option	1. Tap on the application icon 2. Tap on emergency call option	A dialog box is opened to ask the user for confirmation to make a call

Test Case ID: 82 **Test Case Name: Failed to ask the user before initiating the call**		
Summary: Application is not asking for the user confirmation before initiating the call		
Preconditions: Application is installed		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Click on the emergency call option	1. Tap on the application icon 2. Tap on emergency call option	User won’t be asked for the confirmation before initiating the emergency call

Test Case ID: 83

Test Case Name: Initiating the call successfully

Summary:

Application should initiate the emergency call when user pressed the call button on the confirmation dialog box

Preconditions:

Confirmation dialog box is available

Specific contact is selected

Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	
2.Click on the emergency call option	2. Tap on emergency call option	Emergency call is initiated successfully
3.Press the call button on the confirmation dialog box	3. Tap on the call button	

Test Case ID: 84

Test Case Name: Failed to initiate the call

Summary:

Application is unable to initiate the emergency call when user pressed the call button on the confirmation dialog box

Preconditions:

Confirmation dialog box is available

Specific contact is selected

Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	
2.Click on the emergency call option	2. Tap on emergency call option	Emergency call is not initiated
3.Press the call button on the confirmation dialog box	3. Tap on the call button	

Test Case ID: 85 **Test Case Name: Cancelling the emergency call successfully**		
Summary: Application should cancel the emergency call when user pressed the cancel button on the confirmation dialog box		
Preconditions: Confirmation dialog box is available Specific contact is selected		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	
2.Click on the emergency call option	2. Tap on emergency call option	Emergency call is cancelled
3.Press the cancel button on the confirmation dialog box	3. Tap on the cancel button	

Test Case ID: 86 **Test Case Name: Failed to cancel the emergency call**		
Summary: Application is unable to cancel the emergency call when user pressed the cancel button on the confirmation dialog box		
Preconditions: Confirmation dialog box is available Specific contact is selected		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Tap on the application icon	
2.Click on the emergency call option	2. Tap on emergency call option	Emergency call not cancelled
3.Press the cancel button on the confirmation dialog box	3. Tap on the cancel button	

APPENDIX IV

Test cases for Car Rental Management System

Test Case ID: 1 **Test Case Name: Reading the information successfully**		
Summary: Make sure that the system is able to read the information entered by the user		
Preconditions: Application is running and input fields are available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	System is able to read the information entered by the user

Test Case ID: 2 **Test Case Name: Failed to read the information**		
Summary: System is unable to read the information entered by the user		
Preconditions: Application is running and input fields are available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	System is unable to read the information entered by the user

Test Case ID: 3 **Test Case Name: Availability of input fields**		
Summary: Make sure the input fields are available		
Preconditions: Application is running		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1.Press enter	Input fields are available

Test Case ID: 4 **Test Case Name: Unavailability of input fields**		
Summary: Input fields are unavailable		
Preconditions: Application is running		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1. Press Enter	Input fields are not available

Test Case ID: 5 **Test Case Name: Enter the information successfully**		
Summary: User is able to enter the required information related to the car		
Preconditions: Application is running and input fields are available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	User is able to enter the required information about the car

Test Case ID: 6 **Test Case Name: Failed to enter the information**		
Summary: User is unable to enter the required information related to the car		
Preconditions: Application is running and input fields are available		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	User is unable to enter the required information about the car

Test Case ID: 7

Test Case Name: Storing the information successfully

Summary:

Make sure that all the entered information about the car is being stored in a file

Preconditions:

Application is running and information has been entered

Step actions:	**Input Values**	**Expected Results:**
1. Open the application 2. Enter the car's information	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	All the information should be stored in a file

Test Case ID: 8

Test Case Name: Failed to store the information

Summary:

Failed to store all the entered information about the car in a file

Preconditions:

Application is running and information has been entered

Step actions:	**Input Values**	**Expected Results:**
1. Open the application 2. Enter the car's information	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	Information is not being stored in the file

Test Case ID: 9 **Test Case Name: Display the information**		
Summary: The information entered by the user about the car should be displayed to the user		
Preconditions: Information is already entered		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information 3.Ask the user to display the information	1.Borrower’s name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000 6.Select the display information option	Information entered by the user is being displayed

Test Case ID: 10 **Test Case Name: Failed to display the information**		
Summary: The information entered by the user about the car is failed to be displayed to the user		
Preconditions: Information is already entered		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information 3.Ask the user to display the information	1.Borrower’s name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000 6.Select the display information option	Information entered by the user is not being displayed

Test Case ID: 11 **Test Case Name: Calculating the profit successfully**		
Summary: System should be able to calculate the profit		
Preconditions: Application is running and all the required information is provided		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information	1. Petrol cost: 10000 2. Maintenance cost: 5000	System perform the calculations based on the information to calculate the profit

Test Case ID: 12 **Test Case Name: Failed to calculate the profit**		
Summary: System is unable to calculate the profit		
Preconditions: Application is running and all the required information is provided		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information	1. Petrol cost: 10000 2. Maintenance cost: 5000	System is unable to perform the calculations

Test Case ID: 13 **Test Case Name: Ask for monthly cost of fuel**		
Summary: System should ask the user to enter the monthly cost of fuel		
Preconditions: Application is running		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information about the car 3.Displaying the information entered	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000 6.Select the display information option	User will be asked to enter the monthly cost of fuel

Test Case ID: 14 **Test Case Name: Do not ask for monthly cost of fuel**		
Summary: System is unable to ask the user to enter the monthly cost of fuel		
Preconditions: Application is running		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information about the car 3.Displaying the information entered	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000 6.Select the display information option	User will not be asked to enter the monthly cost of fuel

Test Case ID: 15 **Test Case Name: Ask for monthly cost of maintenance**		
Summary: System should ask the user to enter the monthly cost of maintenance		
Preconditions: Application is running		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information about the car 3.Displaying the information entered 4. Enter the monthly cost of fuel	1.Borrower’s name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000 6.Select the display information option 7. Fuel cost: 5000	User will be asked to enter the monthly cost of maintenance

Test Case ID: 16 **Test Case Name: Do not ask for monthly cost of maintenance**		
Summary: System is unable to ask the user to enter the monthly cost of maintenance		
Preconditions: Application is running		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information about the car 3.Displaying the information entered 4. Enter the monthly cost of fuel	1.Borrower’s name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000 6. Fuel cost: 5000	User will not be asked to enter the monthly cost of maintenance

Test Case ID: 17 Test Case Name: Calculating the profit successfully		
Summary: System should be able to calculate the profit		
Preconditions: Application is running and all the required information is provided		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information	1. Fuel cost: 5000 2. Maintenance cost: 3000	System perform the calculations based on the information to calculate the profit

Test Case ID: 18 Test Case Name: Failed to calculate the profit		
Summary: System is unable to calculate the profit		
Preconditions: Application is running and all the required information is provided		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information	1. Fuel cost: 5000 2. Maintenance cost: 3000	System is unable to perform the calculations

Test Case ID: 19 **Test Case Name: Calculate the rent**		
Summary: System should be able to calculate the rent for the car		
Preconditions: Per day rent has be set already		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information about the car 3.Enter the number of days for which the car was being rented	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	Total rent for the car has been calculated and displayed successfully

Test Case ID: 20 **Test Case Name: Failed to calculate the rent**		
Summary: System is unable to calculate the rent for the car		
Preconditions: Per day rent has be set already		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the information about the car 3.Enter the number of days for which the car was being rented	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	Total rent for the car can't be calculated

<table>
<tr><td colspan="3"><b>Test Case ID: 21</b>
<b>Test Case Name: Ask the user to enter the number of days</b></td></tr>
<tr><td colspan="3"><b>Summary:</b>
System should ask the user to enter the number of days for which the car was being rented</td></tr>
<tr><td colspan="3"><b>Preconditions</b>:
Application is running and per day rent has been set</td></tr>
<tr><td><b>Step actions:</b></td><td><b>Input Values</b></td><td><b>Expected Results:</b></td></tr>
<tr><td>1.Open the application
2.Enter the required information for the car</td><td>1.Borrower’s name: Ali</td><td>User should be asked to enter the number of days</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Test Case ID: 22</b>
<b>Test Case Name: Not asking the user to enter the number of days</b></td></tr>
<tr><td colspan="3"><b>Summary:</b>
System is unable to ask the user to enter the number of days for which the car was being rented</td></tr>
<tr><td colspan="3"><b>Preconditions</b>:
Application is running and per day rent has been set</td></tr>
<tr><td><b>Step actions:</b></td><td><b>Input Values</b></td><td><b>Expected Results:</b></td></tr>
<tr><td>1.Open the application
2.Enter the required information for the car</td><td>1.Borrower’s name: Ali</td><td>User won’t be asked to enter the number of days</td></tr>
</table>

Test Case ID: 23 **Test Case Name: Per day rent has been set**		
Summary: Per day rent for the car should have been set already		
Preconditions: Per day rent is decided		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the required information 3.Enter the number of days	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	Total rent should be calculated based on the per day rent already being set

Test Case ID: 24 **Test Case Name: Per day rent has not been set**		
Summary: Per day rent for the car is not being set already		
Preconditions: Per day rent has been decided		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the required information 3.Enter the number of days	1.Borrower's name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	Failed to calculate the total rent because of unavailability of the per day rent

Test Case ID: 25 **Test Case Name: Display the total rent**		
Summary: System should be able to display the total rent for the car		
Preconditions: Rent has been calculated		
Step actions:	**Input Values**	**Expected Results:**
1.Open the application 2.Enter the required information 3.Enter the number of days 4.Total rent has been calculated	1.Borrower’s name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	Total rent should be displayed to the user

Test Case ID: 26 **Test Case Name: Unable to display the total rent**		
Summary: System should be able to display the total rent for the car		
Preconditions: Rent has been calculated		
Step actions:	**Input Values**	**Expected Results:**
1. Open the application 2. Enter the required information 3. Enter the number of days 4. Total rent has been calculated	1.Borrower’s name: Ali 2.Rented days: 2 3.Car # Wf838 4.Car company: Suzuki 5.Car cost: 500000	System is unable to display the total rent to the user