

THE INSURANCE MANAGER- MOBILE APPLICATION

A Paper  
Submitted to the Graduate Faculty  
of the  
North Dakota State University  
of Agriculture and Applied Science

By

Aditi Sharma

In Partial Fulfillment of the Requirements  
for the Degree of  
MASTER OF SCIENCE

Major Program:  
Software Engineering

April 2017

Fargo, North Dakota

North Dakota State University  
Graduate School

---

**Title**

The Insurance Manager – Mobile Application

---

**By**

Aditi Sharma

---

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

**MASTER OF SCIENCE**

SUPERVISORY COMMITTEE:

Dr. Kenneth Magel

---

Chair

Dr. Kendall Nygard

---

Dr. Limin Zhang

---

Approved:

04/13/2017

---

Date

Dr. Brian Slator

---

Department Chair

## **ABSTRACT**

People buy several expensive products with the insurance whether it is a Camera, Phone, Television etc. Insurance usually comes with an end date and other important information like: Product Name, Vendor Name, Price of the insurance, etc. so that a buyer can claim the insurance. For buyers who have multiple products it is hard to keep track of all of the information.

This application will help to solve this problem. Insurance Manager is a leading insurance IT solutions provider for all segments of the insurance community and insurance product management.

The purpose of this application is to bring ability to store and track Product Insurance. Example: Phone, Laptop etc. It serves the purpose of your assistant to manage all the information regarding the policies, their due dates. This application would help the users to track last date of the Insurance to use or extend the Insurance before it expires.

## **ACKNOWLEDGEMENTS**

This research would not have been possible without the constant support, motivation and guidance of Dr. Kenneth Magel. He has been an extremely invaluable resource during my entire Master's Degree program.

I would also like to thank my committee members Dr. Kendall Nygard and Dr. Limin Zhang for their time and support on my research. I am very thankful to the Computer Science Department and my International advisor for their co-operation and help.

Finally, I sincerely thank to my parents, my brother and Rahi for their unconditional support and encouragement.

## TABLE OF CONTENTS

ABSTRACT .....	iii
ACKNOWLEDGEMENTS .....	iv
LIST OF TABLES .....	vii
LIST OF FIGURES .....	viii
1. INTRODUCTION .....	1
1.1. Motivation .....	2
1.2. Aim of the Software .....	2
1.3. Literature Review .....	3
1.4. Paper Organization .....	5
2. OBJECTIVES .....	7
2.1. Requirements Elicitation .....	7
2.2. Requirements Analysis .....	8
2.3. Requirements Documentation .....	8
2.3.1. Product Perspective .....	9
2.3.1.1. User Interface .....	9
2.3.1.2. Hardware Interface .....	9
2.3.1.3. Software Interface .....	9
2.3.2. Product Function .....	10
2.3.3. User Characteristics .....	10
2.3.4. Constraints .....	11
2.3.5. Assumptions and Dependencies .....	11
2.3.6. Specific Requirements .....	12
2.3.6.1. Functional Requirements .....	12
3. IMPLEMENTATION .....	16

3.1. Defined Scope .....	16
3.1.1. Phase-1 .....	16
3.1.2. Phase-2 .....	17
3.1.3. Phase-3 .....	17
3.2. Static Decomposition and Dependency Description .....	18
3.2.1. Use Case Diagram .....	18
3.2.2. Activity Diagram .....	23
3.3. The Insurance Manager Application Interface .....	25
4. TESTING .....	31
4.1. Methodology .....	31
5. FUTURE CONSIDERATION.....	37
REFERENCES .....	38

## LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. List of the Test Cases mapped with Functional Requirements.....	32

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Admin User Use- Case.....	19
2. End User Use- Case .....	20
3. Activity Diagram .....	24
4. Screenshot of Login .....	25
5. Screenshot of Register .....	25
6. Screenshot of Home.....	26
7. Screenshot of Add new Product.....	26
8. Screenshot of User Profile .....	27
9. Screenshot of Change Password .....	27
10. Screenshot of Add Product .....	28
11. Screenshot of Insurance Details.....	29
12. Screenshot of Select Date .....	30



## 1. INTRODUCTION

Spain (2017) refers that it is quite evident that how mobile era, known as m-era has boosted the market of Mobile application to another level, where everyone wants to have everything on just single click or with a single touch. With the advent to high quality multimedia phones, everybody is so keen to have the mobile application which increases their productivity so that they can devote their time on other work.

In today's world, everything is there on your phone, so is the case of Insurance Manager, when it comes to manage insurance of your phone, your laptop and any other costly item.

Insurance Manager is the new Age application whose purpose is to provide ability to store and track Product Insurance. This would be one place where you can manage all your insurance details like Vendor name, Product Name, Expiry Date and Due Date etc.

### Objectives of the Proposed System

- To add the product insurance details
- To provide proper information briefly
- To make accuracy and efficient calculations
- To provide data security
- Flexibility of transactions can be completed in time
- To provide huge maintenance of records
- To store the receipts of the products

After searching on internet to find a meaningful and useful application where I can store my Insurance information, I felt the need for developing a new age application. After discussing and sharing this Idea with people around me I realized the Idea of building Insurance Manager

will be helpful for many people. The functionality is inspired by people's feedbacks that are currently experiencing issues while maintaining the details of their product insurances.

With the development of this application, this app would avoid the wrong interpretation and bad calculation of data. The system help the user to see their product's information like vendor name, Insurance information like policy number, due date etc. on the click of a button. The record data is maintained and backed up such a way that data is not loss.

### **1.1. Motivation**

The motivation for designing this Insurance-Manager application came because I, personally, feel it's quite challenging maintaining the Insurance of multiple items with their due dates, renewal dates etc. Further, maintaining this important data on papers is a very challenging task because you can't take all the important papers along with you all the time. Moreover, I value recent learning about the Android and Java programming languages as well as seeing how powerful and dynamic they are when it comes to web designing and applications. Also, it would be very helpful to numerous people in maintaining insurance details of their gadgets like Phone, laptop, Tablets etc. The languages used to build this application are Java and .NET because I found them to be extremely useful.

### **1.2. Aim of the Software**

This software is developed to help people out there in the world to maintain insurance details of their gadgets like Mobile, Tablet and Laptops etc. This application allows people to store important details of their gadgets or product they bought insurance for. With the advent of this application, the users will be carrying these details in their pocket. Also, there is a need of reminder like functionality which should remind you about the renewal date and provide you notifications on a timely basis so that you don't miss the renewal date and renew your insurance

on time. This Insurance Manger application is designed, primarily, for those people who maintain their insurance on registers and are looking for such kind of application which can be installed on their phones and they can carry that information all the time with them so that they can use that information as and when required. This application can be downloaded and can be installed on Android phones. This Insurance Manager application is very versatile and can be enhanced by adding more functions and modified graphics for use with commercial purposes. Also, at later stages this application could also be designed for the iPhone users and other IOS.

### **1.3. Literature Review**

According to Macedo (2009), the history of insurance denotes the development of the recent business of insurance against risks, especially regarding death, automobile accidents, and medical treatment etc. The industry provides help to eliminate risks from the individual, employer or any company to the larger community, and provides an important source of long-term finance for both the private and public sectors. The insurance industry has always been profitable and provides attractive employment opportunities for white collar workers who want to live healthy life with all the necessities of life. There was a time, when people were not aware of the term insurance. But when they understood the meaning, they welcome this with open hands nobody wants to risk their lives, which results in more number of insurance, which result in more money coming in business. With the insurance getting noticed, this industry has recently become an integral part of a Country's economy. There has always been a need of term like Insurance where people can be insured about their lives. Once the Life Insurance became popular, there was a time when people started thinking beyond life insurance and this was the time when term like Product Insurance came in picture and this has become very popular among masses. Product Insurance can be of any product. It could be even your phone, your vacuum

Cleaner, your laptop etc. This was the time, when people had immensely loved the idea behind Product Insurance.

When people started taking Insurance for their gadgets, there raised a need for any System which could manage all the insurance related information. Accordingly, people started maintaining all the records on the paper. But as the time passes by, this has also become time taking task which is highly insecure also and the chances of data lost was very high. Slowly and gradually, people have started maintaining all the information in their laptops, which was very useful. There came a time, when everybody wants the information so handy that they can even have them on their phones also and this is how idea of making Insurance Manager came into picture. With the touch of finger, one could easily get the information of their insurance policies. With the advent of Insurance Manager, the life would become very easy as any information regarding the insurance could be stored on the Phone. The proposed system would be very useful from the following perspective:

- i. One can easily store the information regarding their insurance.
- ii. A place where all the important information could be placed.
- iii. It would be capable of storing images of the product receipts.
- iv. Insurance Manager can generate notifications also regarding the due dates of the insurance policy.
- v. It would be efficient enough to store the insurance in different categories in order to make it more user-friendly
- vi. Most important, the application would be on phone, which is a very common thing in today's world and everybody carried multimedia phones which are capable of storing of large information also.

With Insurance Manager Application coming in picture, the existing system where people maintained the insurance has become old-fashioned. People could easily flaunt this small application to their friends/colleagues/family. An application should be user-friendly so that users don't get confused while navigating into the application and also it should have the more interactive UI in order to keep the interest of users in the application. The Insurance Manager would be serving all the features mentioned above in order to make the application user-friendly.

These types of applications are designed for personal or professional purposes to generate revenue by providing these applications to users looking for such options to store their information regarding insurances. The application proposed in this paper is more focused on developing a simple, yet complete, application specifically designed for everyone who wants to keep an eye on their policies while walking, travelling etc. i.e., they can view this information at any time because this application will be residing in your phones.. This application performs all the basic functions like storing product information policy date, due date, receipts etc. Other functions like notifications is a very nice feature where one can be notified of their insurance policy's due date. The final application is expected to store enough data regarding the insurance, saving bills, saving card information etc. Once registered to this application, User will register with their Email Id and create a password for their account. Once the account is created the user will be able to sign into the application and will get benefits of the app.

#### **1.4. Paper Organization**

The rest of the document is divided into three parts:

**Objectives** - The Objectives chapter lists the need for building the system. It provides use cases to help the business and technical users with their understanding. It also gives a detailed

explanation for each use case to help with design and implementation, and outlines the constraints regarding the software.

**Implementation-** The Implementation chapter contains the detailed design of the system, including the Class Diagram, Activity Diagram, and Component Diagram. This chapter also includes a detailed explanation for each component as well as the interaction of the class and its components with each other when carrying out certain tasks, besides software's mock screen shots.

## **2. OBJECTIVES**

All the steps required in the software-analysis process related to this project (product function, user characteristics, functional and non-functional requirements, constraints, assumptions, and dependencies for the Insurance Manager application) are described in the following sections.

The requirement gathering and analysis process is the most important part of any software engineering project. The requirements should be clear, concise, non-ambiguous, complete and easy to understand. Requirements analysis is a process in any software engineering project which helps to make sure that the team has analyzed the tasks and scope of the project. It helps to analyze the needs and concerns to build a new product/application.

The requirement analysis process requires having multiple sessions with the stakeholders of the projects to gather and analyze the requirements. All the documents need to be analyzed and should relate to the defined system design. Following are the steps for the requirements analysis:

### **2.1. Requirements Elicitation**

This is gathering of requirements, business analysts organize several meetings with multiple stakeholders to gather requirements. It starts with gathering high level requirements and then has multiple meetings with open ended questions for detailed discussion to get the detailed level requirements.

To gather requirements Analysts needs to define stakeholders. Stakeholders are the ones who give requirements, they are the people or an organization who have interest or benefits from the project, they can be the one who operates the system, They can be directly or indirectly involved in purchasing the system.

Stakeholders can be an organization who is responsible for the system design. The organization can be the one who is financially supporting the system build.

## **2.2. Requirements Analysis**

This step makes sure that the requirements team gathered in the previous step are Clear, Concise and Complete. Analyst needs to make sure that the requirements are non-ambiguous, if the requirements are not clear it will lead to the failure of the project.

It requires identifying different kinds of stakeholders and subject matter experts (SMEs) and taking their needs into account, to document what they need for the success of the project. In this step analyst help stakeholders to understand the circumstances of design the new system, provide them ideas about which module is the best and which ones are more cost expensive. All this information helps analysts to create a software requirement specification document. This step is to define functional and non-functional requirements, create user stories, develop multiple scenarios or identify the use cases used for the projects.

## **2.3. Requirements Documentation**

This step involves documenting the requirements in various forms, including summary lists, natural language documents, visual documents, use cases, user stories, or process specifications. A requirement specification document is categorized in different ways according to the stakeholders' need, helping to create a clear contract between development and business. The following sections include the different categories of requirements specification document that are essential for designing this application: the functional requirements, constraints, system requirements, etc.



### **2.3.1. Product Perspective**

The Insurance Manager application is an Android-based Application. It can be accessed using any Android Phone having Android Version Gingerbread and above.

#### **2.3.1.1. User Interface**

The interface type found in Insurance Manager is as follows:

**1. User Interface:** Users are able to view Product Name, Store name where from you have purchased the product, Price, Insurer, Start Date, End Date etc. Users can browse and add any number of items from any categories in the system, look for information about each product, delete the items in the system.

**2. Admin Interface:** The Admin will be able to view the User Information, User profile, The products information User has saved in their application, Admin will also be able to view the reminders and Insurance details User has saved.

#### **2.3.1.2. Hardware Interface**

The Insurance Manager application shall provide minimum hardware requirements. The following hardware configurations are required for an Android Phone using the Insurance Manager application:

- 200 MHz processor
- 32 MB of RAM
- 32 MB of storage.

#### **2.3.1.3. Software Interface**

This section lists the requirements that are needed to run the application efficiently.

- Android version 4.0 (Ice Cream Sandwich) or later
- Internet access (via WiFi or cellular data network)

### **2.3.2. Product Function**

The Insurance Manager application would have the following basic functions:

- i. Display product insurance information.
- ii. Display all the information regarding due date, start date of the product whose information needs to be submitted.
- iii. Allow the user to add information of newly bought products, gadgets etc.
- iv. Allow users to remove the information of the product's insurance.
- v. Allow the user to modify the information regarding the insurance of the product.
- vi. Allow the administrator to update information regarding the insurance of the product.

### **2.3.3. User Characteristics**

The user of the online Insurance Manager application is the person who would download and install the application on their android phone. A User can be a user of this application who has installed the app in their phone and wants to use the app to store and get benefits from the application features, User can also be the administrator (owner) of the application.

**Users (User):** The users of this Insurance Manager application are all users who would install the application on their phone. These users are anyone with any application usage experience and the know-how to use any application. The users should be able to perform the following functions using this application:

- i. View all the items stored in the system
- ii. Upload the receipts of the products, whose insurance information is to be stored on the application.
- iii. Delete items from the application whose insurance has been expired
- iv. Sign-on/login using a username and password

**Administrator (owner):** The Administrator (Admin) is the owner of the application, The administrator will be responsible for maintaining all the training documents required for the application and that is why the administrator must have the basic knowledge of computers and must know how to use internet. The administrator must have knowledge of operating the eclipse and Java Programming language to maintain the application and resolve any issues or bugs whenever required.

The Administrator will be able to:

1. Maintain the training documents
2. Update the application and provide new versions when required
3. Maintain User data in the server
4. Responsible to resolve any bugs and issues

#### **2.3.4. Constraints**

- i. **Hardware Limitations:** The minimum hardware requirement for the Android phone is 128 MB of Ram and a 10-MB internal/external memory (apart from the Operating System)
- ii. **Accessibility:** Initially, the application would be available for the Android phones.
- iii. **Others:** The application should be built using Android, Java and JavaScript inscribed in XML, and it should, initially, be accessible through the eclipse IDE and later published on a server.

#### **2.3.5. Assumptions and Dependencies**

The assumptions and dependencies are as follows:

- i. Users are accustomed to the paper-based system to store insurance information and would need to go through the Training documents to understand how the application works
- ii. The application is dependent on the availability of an Android Phone to be installed.
- iii. We assume that application users adhere to the application's minimum software and hardware requirements.

### **2.3.6. Specific Requirements**

This section contains details and specific requirements about Insurance Manager Application, This will include details of the software in the manner which will help Developers to understand and build the application. Developers use specific requirement details to create each feature and functionality based on the user's requirements. This section will also have details which will help to build detailed Test plans and Test cases. This will include details on the Graphic User Interface of the application's functionalities. This will also describe the application inputs, functions and application responses. Specific requirements can have multiple types like: Functional requirement, Non-functional requirements.

#### **2.3.6.1. Functional Requirements**

A functional requirement is a requirement which describes function or component of a software system. Function can be the behavior of the software system, a set of inputs and outputs. FR may contain calculations required to build a software system, technical details and other specific functionality of the software. Functional requirements are supported by non-functional requirements, which execute constraints on the implementation, such as performance requirements, security, or reliability. A successful software system will cover all the

requirements asked by the user.

[[www.ofnisystems.com/Validation/Functional\\_Requirements.htm](http://www.ofnisystems.com/Validation/Functional_Requirements.htm)]

**User Experience:** The main purpose of this application is to build an application which can be useful for any user who wants to store and manage their product's insurance details. So the functionality of this application must be user friendly and must be easy to use, it must also provide all the basic features and functionalities to successfully achieve the goal of the application.

- FR01: User must be able to download the application in their Android phone
- FR02: User must be able to install the application in their Android phone

**Sign-Up/Sign-In Screen:**

- FR03: User must be able to view the Sign Up/Sign-in page of the application
- FR04: User must be able to Sign up for the application to create account
- FR05: User must be able to Sign-In in the application

**Home Page:**

- FR05.1: User must be able to view the Home page of the application
- FR06: User must be able to view the list of the added Items (If any) at the home page
- FR07: User must be able to click at the added Items on the home page  
FR05.2: User must be able to view the Side bar on the Home page by clicking the Side bar button

**Side Menu Bar:**

- FR08: User must be able to view the list of the other functionalities and features of the application in the side bar:

- FR09: User must be able to view the “User profile by clicking the “User Profile” button in the side bar
- FR10: User must be able to add a new product by clicking the “Add new product” button in the side bar
- FR11: By clicking “Insured product near expire” button in the side bar, User must be able to view the list of the products whose Insurance is about to expire
- FR12: User must be able to view the ‘Setting” button in the side bar
- FR13: User must be able to logout from the application by clicking the “Logout” button in the side bar.

#### **User Profile Page:**

- FR14: User must be able to view and Edit User’s first Name
- FR15: User must be able to view and Edit User’s Last name
- FR16: User must be able to view and Edit User’s Email ID
- FR17: User must be able to reset their Password
- FR18: User must have a Save button to be able to save their profile information

#### **Add New Product:**

- FR19: User must be able to add a new Product
- FR20: User must be able to add details for the product
- FR21: User must be able to save the details for the product
- Product Details Page:
- FR22: User must be able to view Product Details on the product details page
- FR23: User must be able to view Vendor Details on the Product Details Page
- FR24: User must be able to view Insurance Details on the Product Details page

- FR25: User must be able to view and click the back button on the product details page so that user can go back to the Home page
- FR26: User must be able to view and click Edit button to Edit/modify Product details
- FR27: User must be able to select “Notification Period”

**Administrator Requirement:**

- The administrator must be able to view the entire history for the users who successfully created an account in Insurance Manager
- The administrator must be able to view the entire history of the Added Products.

### **3. IMPLEMENTATION**

This chapter includes the detailed design description used to build Insurance Manager Application. The system's design helps to create operations, functions and features for the project based on the gathered requirements, mockups, process flow diagrams, system flow diagrams and business rules. In this sections designed are based on the initial signed off requirements from the Requirement specification section. By using multiple prototypes, In this section I have created several designs for each requirement. These designs will include software features in detail including functional diagrams, activity diagrams, class diagrams, wireframes and mockups. These diagrams will describe the application's features in detail, these will help to develop the application.

#### **3.1. Defined Scope**

To make this application implementation successful I have divided this application in multiple Phases. By dividing this application implementation in phases it will also help to implement this in Agile methodology, so this application features can be delivered in chunks by using multiple Sprints. I have also written Use Cases for this application later in this paper, which is also a big part of Agile Methodology.

##### **3.1.1. Phase-1**

In the first Phase -1 the application must be able to register a user and allow him to login with the same credentials:

- User will be able to view log-in page with register new account link
- User will be able to click onto the register new account link
- User will be able to enter all the required information on the register new account link



- User will be able to successfully register a new account
- User will be able to Log-in with using the same credentials which he used to register an account

### **3.1.2. Phase-2**

In the Phase-2 the user will be able to view the home page and view the side bar with all the options, user will also be able to view his profile information

- Once the user logged in, the user will be able to view the home page
- The user will be able to view the side bar
- The user will be able to click on the 'Profile' tab in the bar
- The user will be able to view his profile with the details he used to register the account
- The user will be able to click onto the 'Home' tab to go back to the Home page

### **3.1.3. Phase-3**

In the Phase-3 the user will be able to add product, modify product or remove product.

- The user will be able to click onto the add new product link in the side bar
- The user will be able to add a new product
- The user will be able to add insurance details
- Once the user has added the product and Insurance details he will be able to save the product
- The user will be able to view the saved product and its details
- The user will be able to Edit/Modify the existing products
- The user will be able to delete the added products
- The user will be able to select Notification Period

- The list of the products user added they will be able to view the list on the Home page

### **3.2. Static Decomposition and Dependency Description**

This section contains the system use-case diagrams and description for the In-store Manager application and also explains use cases in detail.

#### **3.2.1. Use Case Diagram**

The user can see detail of each use case of the system by reviewing this use case diagram and details. This use case explains that how the actor would interact with the system and with each other. The use cases designed below explains about the different actor areas and detail description of each use-case has been provided below with the details on Pre-condition and Post-condition. The Pre-condition and Post-condition should be fulfilled once the use cases is implemented and completed in the application. (Administrator Use-Case Diagram) explains the use case for the administrator and (User Use case diagram) explains the User interaction with the system.

#### **Administrator Use-case:**

(Administrator Use-Case Diagram) explains the use case for the administrator as an actor, where an admin access the application, the admin can access Home page, Add New product and other pages including the Database.

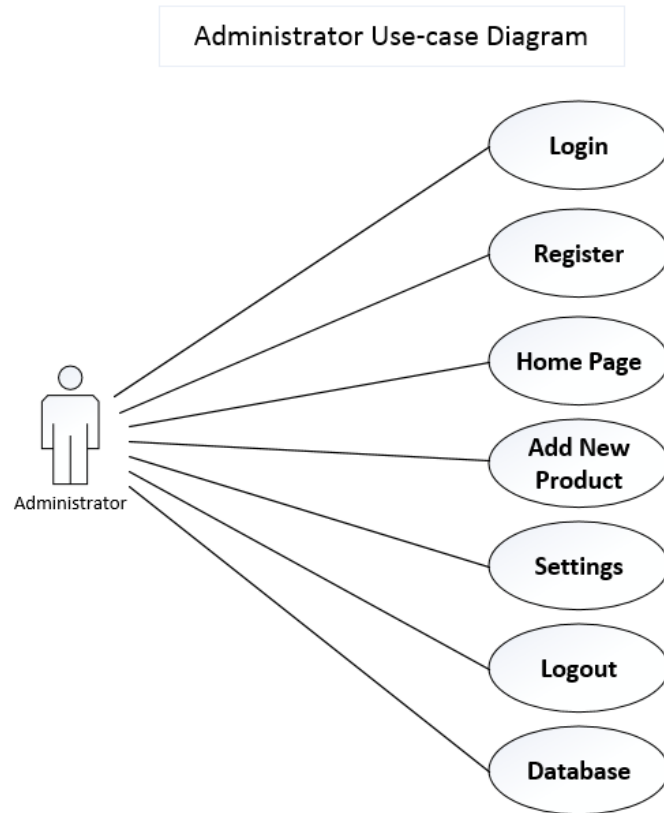


Fig. 1. Admin User Use- Case

**End User Use-case:**

It explains the use case for the End User as an actor, where a user allow login, register other features which is mentioned in the use case

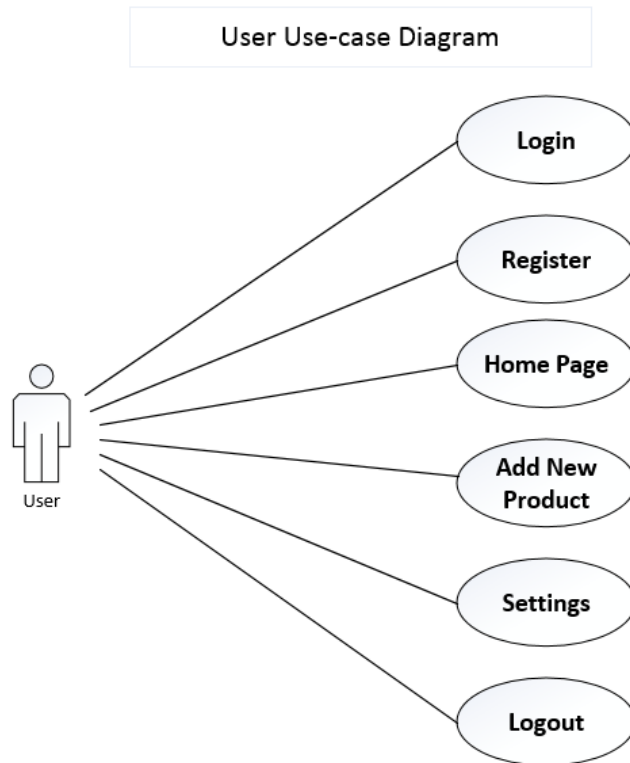


Fig. 2. End User Use- Case

**1. Use-Case Number: US-001**

**Application:** Insurance Manager Application

**Use-Case Name:** Log In

**Use-Case Description:** This use case lets the user/administrator view the Log-in page which has options to enter their credentials to be able to log-in into the app

**Primary Actor:** Admin/User

**Precondition:** Run the application

**Post-condition:** The user is able to run the application and able to view the Log-in page with the options to be able to add their credentials

**Basic Flow:**

- Run the Application

- View the Log-In Page
- Enter the credentials to log-in into the account

## 2. Use-Case Number: US-002

**Application:** Insurance Manager Application

**Use-Case Name:** Register

**Use-Case Description:** This use case lets the user/administrator to register an account in the insurance manager application

**Primary Actor:** Admin/User

**Precondition:** Run the application

**Post-condition:** The user is able to run the application and able to view the Log-in page and clicks the Register link to be able to register an account

**Basic Flow:**

- Run the Application
- View the Log-In Page
- Click the Register link
- View the page to register a new account
- Successfully registers an account

## 3. Use-Case Number: US-003

**Application:** Insurance Manager Application

**Use-Case Name:** Add New Product

**Use-Case Description:** This use case lets the user/administrator to Add a New Product in the insurance manager application

**Primary Actor:** Admin/User

**Precondition:** Run the application

**Post-condition:** The user is able to view the added Product on the Home page

**Basic Flow:**

- Run the Application
- Log-in
- Click the Add New Product
- Add all the required Product details
- Product is added in the Account and
- Product is displayed on the Home page

**4. Use-Case Number: US-004**

**Application:** Insurance Manager Application

**Use-Case Name:** Settings

**Use-Case Description:** This use case lets the user/administrator to Edit the Profile

Information in the Insurance manager application

**Primary Actor:** Admin/User

**Precondition:** Run the application

**Post-condition:** The user is able to view and edit the Profile information

**Basic Flow:**

- Run the Application
- Log-in
- Click Settings
- Edit Profile Settings
- Save Profile setting

**5. Use-Case Number: US-005**

**Application:** Insurance Manager Application

**Use-Case Name:** Add New Product

**Use-Case Description:** This use case lets the user/administrator to Logout from the insurance manager application

**Primary Actor:** Admin/User

**Precondition:** Logged-In the application

**Post-condition:** The user is able to Log Out from the application

**Basic Flow:**

- Run the Application
- Log-in
- Click the Logout button
- User is logged out from the application

### **3.2.2. Activity Diagram**

This section has the activity diagram to display the flow of the activities in the application. A detailed description is then given after the figure for each activity. Figure 3 provides the overview of the activity of Insurance manager application

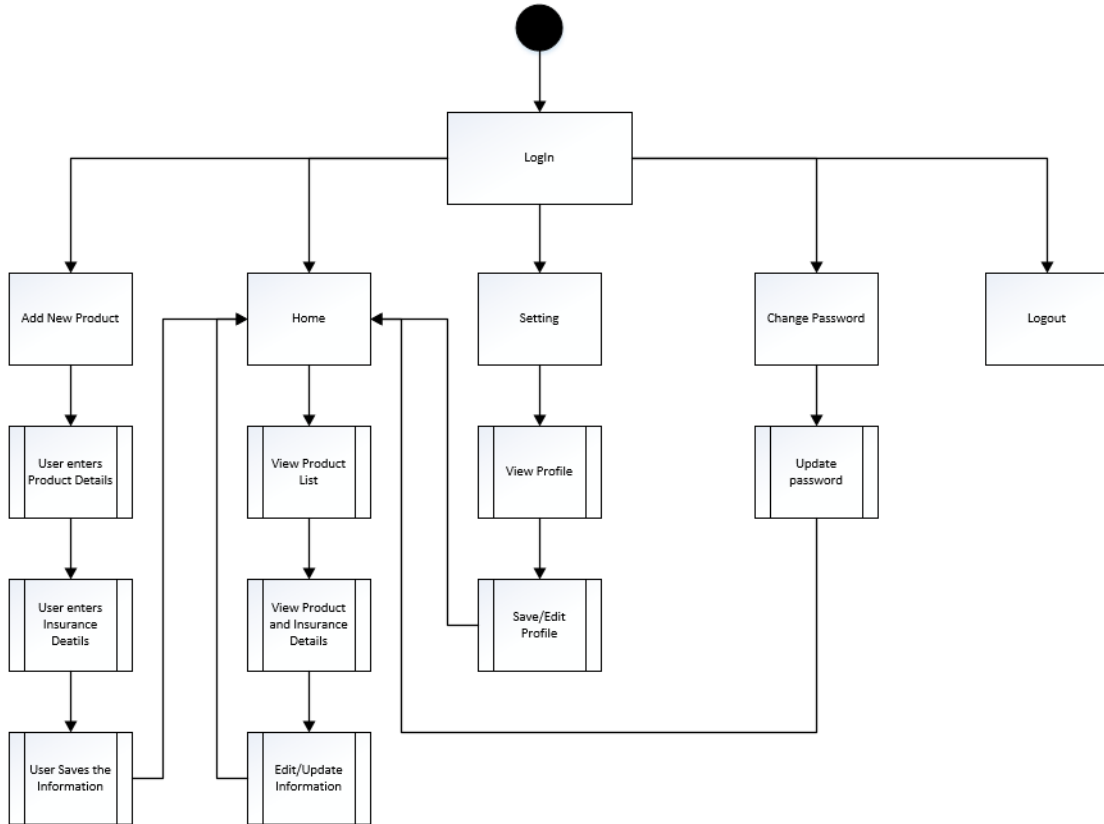
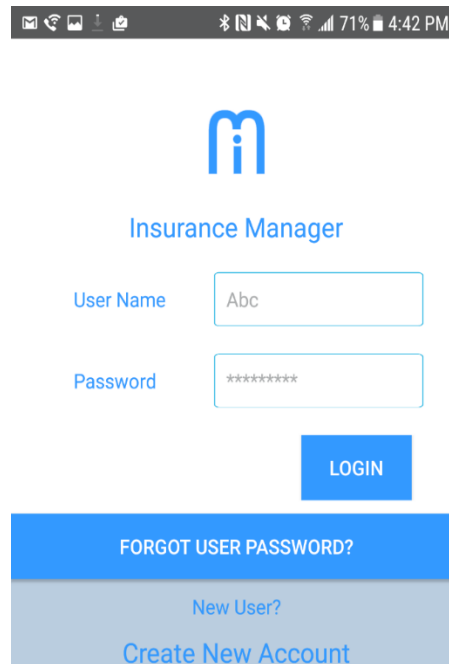


Fig. 3. Activity Diagram

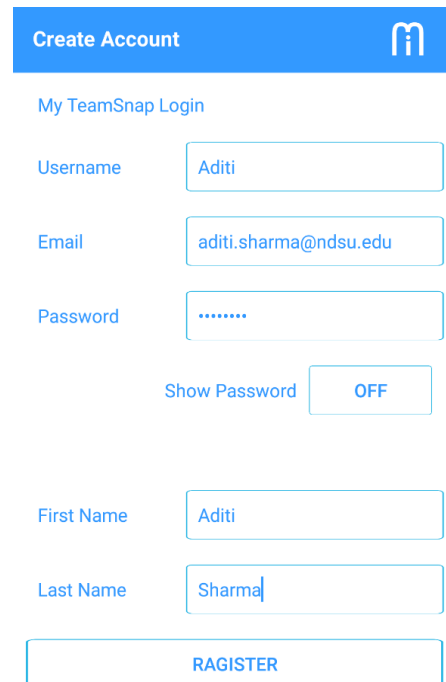


### 3.3. The Insurance Manager Application Interface



The screenshot shows the login page of the Insurance Manager application. At the top, there is a status bar with various icons and the time 4:42 PM. Below the status bar is the application logo, a stylized 'm' in blue. The title 'Insurance Manager' is displayed in blue text. The login form consists of two input fields: 'User Name' with the text 'Abc' and 'Password' with asterisks. A blue 'LOGIN' button is positioned below the password field. At the bottom, there are two buttons: a blue 'FORGOT USER PASSWORD?' button and a grey 'New User? Create New Account' button.

Fig. 4. Screenshot of Login



The screenshot shows the registration page of the Insurance Manager application. The title bar is blue with the text 'Create Account' and the application logo. Below the title bar, the text 'My TeamSnap Login' is displayed. The registration form includes four input fields: 'Username' with the text 'Aditi', 'Email' with the text 'aditi.sharma@ndsus.edu', 'Password' with asterisks, and 'First Name' with the text 'Aditi'. The 'Last Name' field contains the text 'Sharma'. A 'Show Password' toggle switch is currently set to 'OFF'. A blue 'REGISTER' button is located at the bottom of the form.

Fig. 5. Screenshot of Register

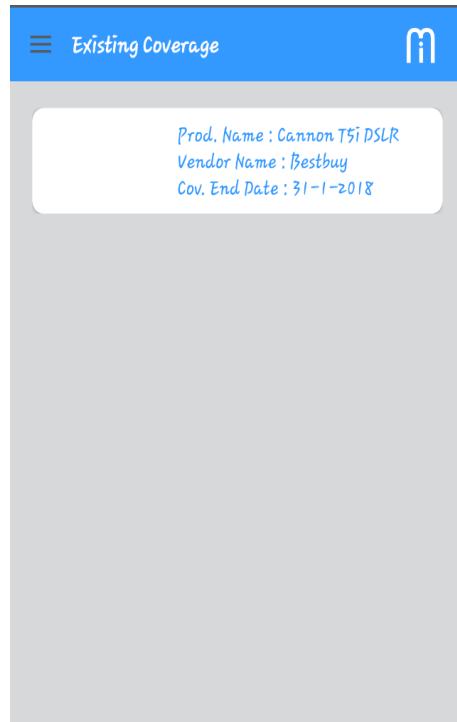


Fig. 6. Screenshot of Home

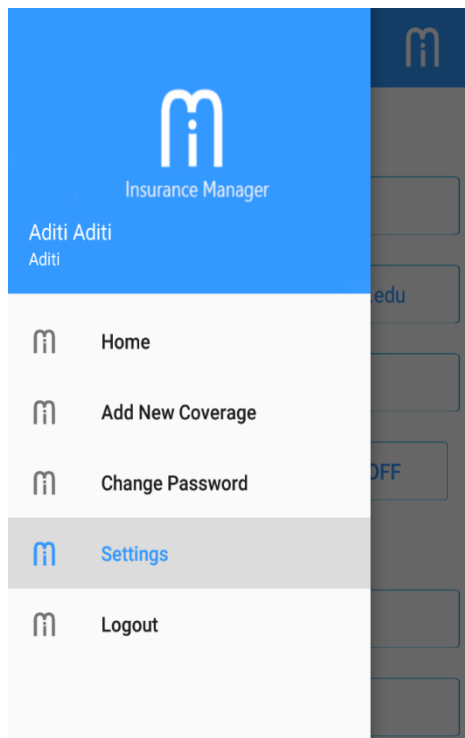



Fig. 7. Screenshot of Add new Product

The screenshot shows a 'Settings' page with a blue header containing a hamburger menu icon, the text 'Settings', and a user profile icon. Below the header, the section 'Update Account Details' is visible. It contains three input fields: 'Username' with the value 'Aditi', 'Email' with the value 'aditi.sharma@ndsu.edu', and 'Password' with masked characters '.....'. A 'Show Password' toggle switch is currently set to 'OFF'. Below these fields are two more input fields: 'First Name' with the value 'Aditi' and 'Last Name' with the value 'Sharma'. At the bottom of the form, a dark grey notification bubble contains the text '"your details has completed successfully"'. A vertical bracket on the right side of the form groups the Username, Email, and Password fields.

Fig. 8. Screenshot of User Profile

The screenshot shows a 'Change Password' page with a blue header containing a hamburger menu icon, the text 'Change Password', and a user profile icon. The main content area features a large blue user profile icon at the top, followed by the text 'Change Password'. Below this, there are three input fields: 'Old Password', 'New Password', and 'Re Enter New Password'. At the bottom of the form is a blue button with the text 'CHANGE PASSWORD'. A vertical bracket on the right side of the form groups the three password input fields.

Fig. 9. Screenshot of Change Password

☰ Add New Account 

### Product Details

Product Name

Product Type

Brand

Model Number

Serial Number

Product Price

Date of Purchase

### Vendor Details

Vendor Name

Vendor Location

Fig. 10. Screenshot of Add Product

### Insurance Details


Company	Endeavor
Policy Number	1234
Insurance Cost	\$ 235
Claim Limit	\$ 123
Notice Period	6 Months 
Coverage Start Date	DD-MM-YYYY
Coverage End Date	DD-MM-YYYY
Other Information	<div style="border: 1px solid #00aaff; border-radius: 10px; padding: 10px; min-height: 100px;">this is my information</div>

Fig. 11. Screenshot of Insurance Details

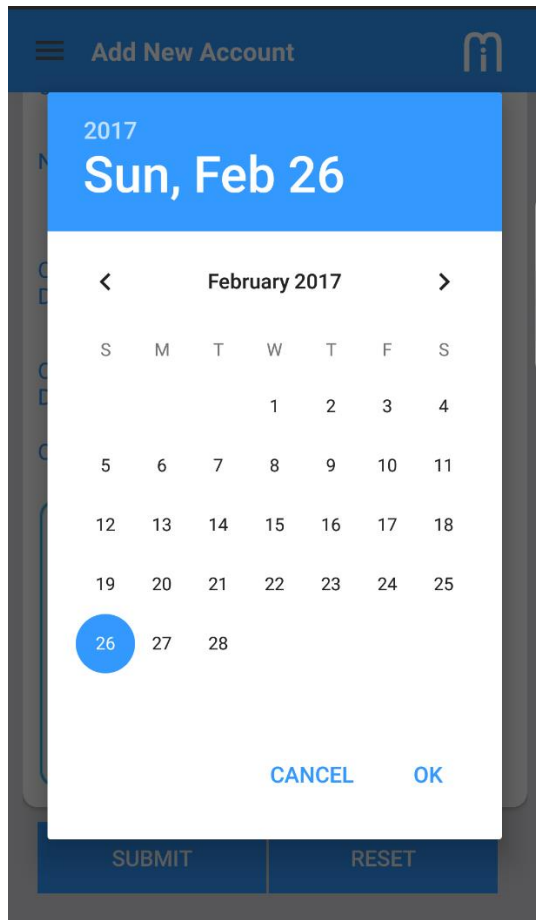


Fig. 12. Screenshot of Select Date

## **4. TESTING**

This chapter explains the methods that were used for testing, and validation the application. This will also have some samples of the test cases used to test the application, the test data is also given.

### **4.1. Methodology**

Approaching the structure of the Agile methodology, the testing of this application started approximately at the same time when the use case writing started. The test cases were started documenting based on the high level requirements and then started to be detailing out based on the use-cases. The testing was also divided into phases. The Phase-1 considered completed only when the tested was completed for phase-1 based on the phase-1 test cases. This also helped to minimize the testing that would have to be implemented at the end of the software lifecycle. However an end to end testing was also involved in each phase.

Steps followed in order to approach the methodology:

1. Create a high level functional documentation that includes high level information of all the features which are in scope for that phase.
2. Pass that document to the testers and testers would plan their high level test document.
3. Meanwhile work on the detail level requirement documentation and process flow diagrams, share all the wireframes or mock ups.
4. Share the detailed document with the testers and they would work on the detailed test case document based on the requirements.
5. Once the code is ready share all the technical information with the testers which will help them to execute their test cases documents like database tables to be used and a description of each component with the precondition with the precondition and tables that

would be affected by the component or an activity diagram with a system interaction flow.

6. Testers will log all the defects they find for the coder to fix them. The defects would be prioritized based on the criticality of the feature. The high priority defects would be fixed before the medium or the lower priority defect.
7. The end to end testing would be done at the end of the phase to make sure all the features are working as expected also all the defects have been fixed.

Table. 1. List of the Test Cases mapped with Functional Requirements

Functional Requirement Number	Functional Requirement Short Description	Test Case Number	Test Case Short Description
<b>Phase-1: Sample Test Cases</b>			
FR01	User must be able to download the application in their Android phone.	TC01	Demonstrate that the application is downloading in the user's Android phone.
FR02	User must be able to install the application in their Android phone.	TC02	Demonstrate that the application is successfully installed in the user's Android phone.
<b>Phase-2: Sample Test Cases</b>			
FR05	User must be able to Sign-In in the application.	TC04	Demonstrate that the user successfully Sign-in in the Insurance Manager Application.
<b>Phase-3: Sample Test Cases</b>			
FR10	User must be able to add a new product by clicking the "Add new product" button in the side bar.	TC05	Demonstrate that the user successfully adds a product in the Insurance Manager Application.
FR20	User must be able to add details for the product.	TC06	Demonstrate that the user successfully adds a Product's Insurance information.



The following steps would be taken by the user, the requirements that should be met for the valid execution of the test case, and the end result that required to be match with the expected results or the test case in order to pass the test.

### **Phase-1: Sample Test Cases**

#### **1. TC01: Download the application**

Demonstrate that the application is downloading in the user's Android phone.

**Input:** The user runs the Insurance manager download file of the application to download the Insurance manager application in the Android phone.

**Output:** The Insurance manager application is successfully downloaded in the user's Android phone

**Valid Range:** This must match with the Software Interfere requirements:

- Android version 4.0 (Ice Cream Sandwich) or later
- Internet access (via Wi-Fi or cellular data network)

#### **End Messages/Result:**

- I. If the Download = Successful, the application will be downloaded in the Android phone.
- II. If the Download = Unsuccessful, an error message is displayed that the application could not be downloaded.

#### **2. TC02: Install the application**

Demonstrate that the application is successfully installed in the user's Android phone.

**Input:** The user has the Insurance manager downloaded file in the Android phone and uses it to install the Insurance Manager application in the Android phone.

**Output:** The Insurance manager application is successfully installed in the user's Android phone

**Valid Range:** This must match with the Software Interfere requirements:

Android version 4.0 (Ice Cream Sandwich) or later

Internet access (via WiFi or cellular data network)

**End Messages/Result:**

- I. If the Installation = Successful, the application will be Installed in the Android phone.
- II. If the Installation = Unsuccessful, an error message is displayed that the application could not be Installed.

### **Phase-2: Sample Test Cases**

#### **3. TC03: Create an Account**

Demonstrate that the user successfully creates an account in the Insurance Manager Application

**Input:** The user attempts to create an account in the Insurance Manager application which is installed in the user's phone.

**Output:** The user has an account in the Insurance Manager application based on the information he or she entered.

**Valid Range:** User must have the Insurance manager application installed in a supporting Android phone.

**End Messages/Result:**

- I. If the Create Account = Successful, the user will have an account in the application.
- II. If the Create Account = Unsuccessful, an error message is displayed that the account has not been created

#### **4. TC04: Sign-In**

Demonstrate that the user successfully Sign-in in the Insurance Manager Application.

**Input:** The user enters the personal credentials to log in into the Insurance Manager application.

**Output:** The user has an account in the Insurance Manager application based on the information he or she entered.

**Valid Range:** User must have the Insurance manager application installed in a supporting Android phone.

**End Messages/Result:**

- I. If the Sign-In = Successful, the user will have an account in the application.
- II. If the Sign-In = Unsuccessful, an error message is displayed that the sign in was unsuccessful due to one of the following reasons:
  - a. The User name entered was incorrect.
  - b. The Password entered was incorrect.
  - c. The phone does not have network.
  - d. The user does not have account.

### **Phase-3: Sample Test Cases**

#### **5. TC05: Add New Product**

Demonstrate that the user successfully adds a product in the Insurance Manager Application.

**Input:** The user clicks Add New Product button.

**Output:** The user successfully adds a new product in the application.

**Valid Range:** User must be signed in into the application in order to add a new product.

**End Messages/Result:**

- I. If the Add New Product = Successful, the user will have a new product added into the account.

- II. If the Add New Product = Unsuccessful, the user will receive an error that the product could not be added.

#### **6. TC06: Add Product Details**

Demonstrate that the user successfully adds a product insurance details in the Insurance Manager Application.

**Input:** The user is on the Add product detail page

**Output:** The user successfully adds a product's insurance information for a product

**Valid Range:** User must be signed in into the application in order to add product details

#### **End Messages/Result:**

- I. If the Add Product Details = Successful, the user will have a product with all the insurance details user has entered.
- II. If the Add New Product = Unsuccessful, the user will receive an error that the product details could not be added.

## 5. FUTURE CONSIDERATION

The Insurance Manager application can have many future update consideration, this first version allows the user to create an account, add products and their insurance details. In the upcoming versions the application can also have following options:

- I. **Add Image:** Future work could involve option to add Images for a product so that user can capture product's receipt picture or Product's Picture, this will help user to have more information about the product, date when the user bought the product and other details.
- II. **Automatic Populate Fields:** This feature will prevent user to add Insurance information manually, when user will start typing the field will give user options to select from.
- III. **Categorized Product list:** On the home page where all the existing product is listed they can be listed as categorized. This will help user to find the product they are looking for.
- IV. **Payment Options:** Allow user to pay via application. Add different payment options, such as Visa, MasterCard, PayPal, etc.,
- V. **Increase security:** Build more secure Log-In page. Add "Re-Captcha", "Security Question" or more requirement for strong password to make this application more secure.
- VI. **History:** Provide option to add history of a particular insurance. If the user has used the insurance once then the user will have ability to store the details of the claim so that when the next time user is claiming the user will have all the history stored in the application.
- VII. **Other operating System:** Currently this application supports only Android phones, In future it can be enhanced to support I-phone, Windows, Tablets etc.

## REFERENCES

- [1] Macedo (2009). The Role of the Underwriter in Insurance, Retrieved From:  
[https://siteresources.worldbank.org/EXTFINANCIALSECTOR/Resources/282884-1242281415644/Role\\_of\\_Underwriter\\_Insurance.pdf](https://siteresources.worldbank.org/EXTFINANCIALSECTOR/Resources/282884-1242281415644/Role_of_Underwriter_Insurance.pdf)
- [2] Spain (2017). After a Decade, the Mobile Enterprise Revolution Has Only Just Begun, Retrieved From: <http://www.emcap.com/2017/03/06/decade-mobile-enterprise-revolution-just-begun/#sthash.pC2dMwhh.dpuf>
- [3] Myrberg, C. & Wiberg, N., (2015). Screen vs. paper: what is the difference for reading and learning? *Insights*. 28(2), pp.49–54. Retrieved From:<http://doi.org/10.1629/uksg.236>
- [4] Murphy (2011). More Consumers Using Tablets to Holiday Shop [STUDY]." Retrieved from: <http://mashable.com/2011/12/08/tablets-to-holiday-shop/#7Im3ykJWO8qS>
- [5] Hooson (2016). Get cheap insurance for your gadgets, Retrieved from <https://www.moneysupermarket.com/home-insurance/gadget-insurance-guide/>