DIGITAL POLICY WALLET

MONIKA M G
Register No: 1631029

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

M.Sc. (Software Systems)

OF ANNA UNIVERSITY

November 2019

DEPARTMENT OF COMPUTING
COIMBATORE INSTITUTE OF
TECHNOLOGY
(Autonomous Institution affiliated to Anna University)
COIMBATORE – 641014

TrueCover*

Monika M.G.
Coimbatore Institute of Technology,
Civil Aerodrome Post, Coimbatore - 641014

TO WHOM IT MAY CONCERN

Subject: Internship completion certificate for M.G.Monika

Dear Sir/Madam**,**

This is an internship completion certificate for Ms.M.G.MONIKA, Coimbatore institute of technology.

We state on record that Monika has successfully done an internship project in the role of **Front End Developer intern** at **TRUCVR Technologies Pvt Ltd.** The internship duration was from 20 May 2019 to 20 November 2019 and the location was at BHIVE workspace in Bangalore city.

During this period Monika worked on various areas of front end development for insurance companies/consumers and successfully met the objectives that were set at the beginning of the internship.

Best,

for **TRUCVR Technologies Pvt Ltd**

Peeyush Jain
Cofounder
peeyush@truecover.net
+91-88600 82876

COIMBATORE INSTITUTE OF TECHNOLOGY

(Autonomous Institution affiliated to Anna University)

COIMBATORE 641014

(Bonafide Certificate)

Project Work - I
Seventh Semester

DIGITAL POLICY WALLET

Bonafide record of work done by
MONIKA.M.G
(Register No: 1631029)

Submitted in partial fulfillment of the
requirements for the degree of
M.Sc. (Software Systems)
of Anna University

November 2019

_____ _____

Faculty Guide Head of the Department

Submitted for the viva-voce held on _____

_____ _____

Internal Examiner External Examiner

CONTENTS

ACKNOWLEDGEMENT

This project work would not have been possible without the help of several people and it's indeed a great pleasure for me to acknowledge the efforts and guidance of all them.

I express my sincere gratitude to **Dr.V.SELLADURAI,** Principal and Dean of Computing, Coimbatore Institute of Technology, for permitting me to undertake this project work at TruCVR technologies Private Limited, Bangalore.

I am grateful to **Dr.K.SAKTHIMALA**, Head, Department of Computing (Software Systems), Coimbatore Institute of Technology, Coimbatore, for her encouragement throughout this project.

I am indebted to my internal guide, **Dr.S.SATHYA**, Department of Computing, Coimbatore Institute of Technology, Coimbatore, for her constant support and guidance throughout the project work.

I extend my heartfelt gratitude to my manager, **Mr. PEEYUSH JAIN,** CoFounder, TruCVR Technologies Private Limited, Bangalore, for his continual support, ardent motivation, valuable suggestions, encouragement, inspiration and guidance throughout the project.

SYNOPSIS

Project entitled '**Digital Policy Wallet'** wallet has been designed where the Consumer can store all the retail and corporate policies for free and maximize insurance benefits. Smart tools compare existing policies against family's requirements and give a true picture of insurance needs. For Corporate policies the account manager details, claim administration details and benefits of the policies covered can be seen so that the consumer can easily figure out what all the consumer has been covered for and what all benefits consumer has for it.

The current system consists of applications where the consumers can view either the retail policies or the corporate policies. The disadvantage of existing system is that when an employer has a corporate policy associated with employer's company consumer cannot view both the retail and corporate policies in a single application which becomes a disadvantage.

In the proposed system the Consumer should be able to login to view or add policies. The Consumer must provide dependent's details to choose a desired topup plan for every policy. The insurance policies are arranged by type and sub type with key details easily marked out. A RiskScore measuring the coverage against consumer's and consumer's family's requirements is shown. Community board to help connect with other Consumers and get quick help.

Angular and Java are the major technologies used in development. Angular is like a JS wizard, making it actually fun and easy to work with. Angular make HTML/CSS pages "smart" and responsive. Click a button, only a section changes (no actual page refresh). It allows to create actual "web applications" instead of web pages/sites. It makes them more similar to creating programs and apps, bringing them to the modern age.

PREFACE

CHAPTER I – INTRODUCTION discusses about the organization where the project was done. It continues to give insight into the problem, that this project aims to solve and finally the environment in which the system is designed to work in.

CHAPTER II – SYSTEM ANALYSIS introduces the problem, with detailed analysis and system requirements and the specifications. The analysis model is explained using use case diagram. This chapter also explains the system in detail. The test plan for the project is explicated.

CHAPTER III – SYSTEM DESIGN explains the design with related diagrams outlines the phases of system development and elucidated the Consumer interface flow.

CHAPTER IV –TEST CASES puts forth the test cases, expected output and results for each module.

CHAPTER V – CONCLUSION puts forth the special features and suggestions for future enhancements of this project.

CHAPTER I

INTRODUCTION

This section gives a detailed description of the organization for which the system is developed along with the overview of the existing system, problems associated with it, goals and the scope of the proposed system. It also specifies the system environment used in the development of the proposed system and gives brief introduction about the various technologies used in the development of the system.

1.1 ORGANIZATION PROFILE

- Insurance is one of the most boring business on earth and Truecover was instantiated to make it simple and cool.
- Truecover provides a unique platform that enables a truly digital experience for the insurance consumers.
- While there are many players in the market focusing on distribution of insurance, the company's singular focus remains on consumer empowerment.
- The company believes that by helping consumers understand their risk coverage better the company can assist them in making informed decisions and more effective use of their insurance.
- The company's dream is to make Digital Insurance a new normal. It is building creative tools which will allow all players in the insurance ecosystem to benefit from greater customer engagement.
- TrueCover offers unique benefits to insurers, intermediaries as well as customers to help create a truly digital experience for insurance customers.

1.2 PROBLEM DEFINITION

Objective

- To help consumers understand their risk coverages better the product assists them in making informed decisions and more effective use of their insurance.
- To build an intelligent policy wallet where the Consumer can store all the retail and corporate policies for free and maximize insurance benefits.

Scope

- The Consumer must login to view or add policies and if the consumer is a first time user then dependent details must be provided to complete the profile.
- The application is really helpful because it can arrange insurance policies by type and sub type with key details easily marked out.
- The application shows a risk score measuring the coverage against Consumer's and consumer's family's requirements.
- For active participation of the Consumers, community board has been developed to help connect with other Consumers and get quick help.

Consumers

The Consumers of the system are,

- Insurer - a company or a person who sells the insurance policies, providing financial coverage in the case of unexpected, bad events covered on policy.
- Insured - a person who is the buyer of the insurance policy. An insured is the proposer of the policy who pays premium as the consideration to the insurance company

1.3 SYSTEM ENVIRONMENT

System environment gives the minimal machine and software requirements for the application to run.

Hardware Environment

- RAM - 2GB
- Hard disk - 500MB
- Processor - Intel Core i5-3210M (3rd Gen)

Software Environment

- Operating system - Windows
- Web server - Nginx
- Programming Language - Typescript, Java
- Frameworks - Angular, Spring Boot
- Database - MySQL

Deployment Environment

- Domain Provider - GoDaddy
- Hosting Provider - Amazon EC2

CHAPTER II
SYSTEM ANALYSIS

System Analysis is a process of studying the system in order to identify the goals, purpose of the system and to provide better understanding of the system's requirements. This chapter gives a brief discussion about the detailed study of the proposed system and the different functionalities involved in the system.

2.1 SYSTEM DESCRIPTION

The system is a digital policy wallet to add or view policies, which is web and mobile application. For mobile web clients, the system has a Progressive Web Apps (PWA). The system should be designed in a way to handle the heavy Consumer traffic.

EXISTING SYSTEM

The current system consists of applications where the consumers can view either the retail policies or the corporate policies.

Disadvantage of existing system:

When an employer has a corporate policy associated with employer's company consumer cannot view both the retail and corporate policies in a single application which becomes a disadvantage.

***Advantage of Digital Policy Wallet*:**

Digital policy wallet provides both retail and corporate policies in a single application.

***Digital Policy Wallet provides*:**

In retail policy the consumer can view the premium rate, sum insured and family members covered. In corporate policy the consumer can view the claim administration details, manager details and the benefits of the policy

PROPOSED SYSTEM

The Consumers can login on the platform. For a first time user, the Consumer must give their dependent details to complete their profile. The Consumer can also enroll to policies for which the window period is open. Else if consumer is a repeat Consumer, take the consumer to the consumer dashboard.

Consumer dashboard consists of corporate policies in which consumer has enrolled. It consists of the corporate policy's key details such as claim administration details, account manager details and the benefits of the policy. The Consumer can also add retail policy in the policy wallet and view its benefits and coverage details.

Consumer can upload the personal retail policy by giving details or by uploading a pdf.

The consumer can get a Risk Score for retail policies which measure the coverage against consumer's and consumer's family's requirements which includes both retail and corporate policies.

The consumer can upload KYC documents like PAN, Aadhar, Voter ID, Staff ID or cancelled cheque and MEDICAL documents like Medical bills, prescription or discharge summary for consumer and consumer's family members.

For active participation of the Consumers community board has been developed to help connect with other Consumers and get quick help.

The consumer can edit their profile where they secondary email or mobile where the consumer's can login using their secondary email or mobile. The consumer can add additional family members if any.

They can change the privacy settings where turn on or off notifications and there is an option to change password too. They can have their desired profile picture too.

Apart from these, the consumer can also share their policies to their family members.

FEATURES OF PROPOSED SYSTEM

- Consumer can authenticate to use the system.
- Consumer can enroll into corporate policies for which the window period is open.
- Consumer can add new retail policies.
- Consumer can add KYC and medical documents for consumer and consumer's family members.
- Consumer can also get help from other Consumers through community.
- Consumer can also share their policies to their family members.

2.2 USE CASE MODEL

A UML use case diagram is the primary form of system/software requirements for a new software program under developed. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. Identification of any internal or external factors that may influence the system is also possible through use case diagrams. In addition, use case diagrams provide good high level analysis from outline the system. The use case description of the digital policy wallet is given the Figure 2.1

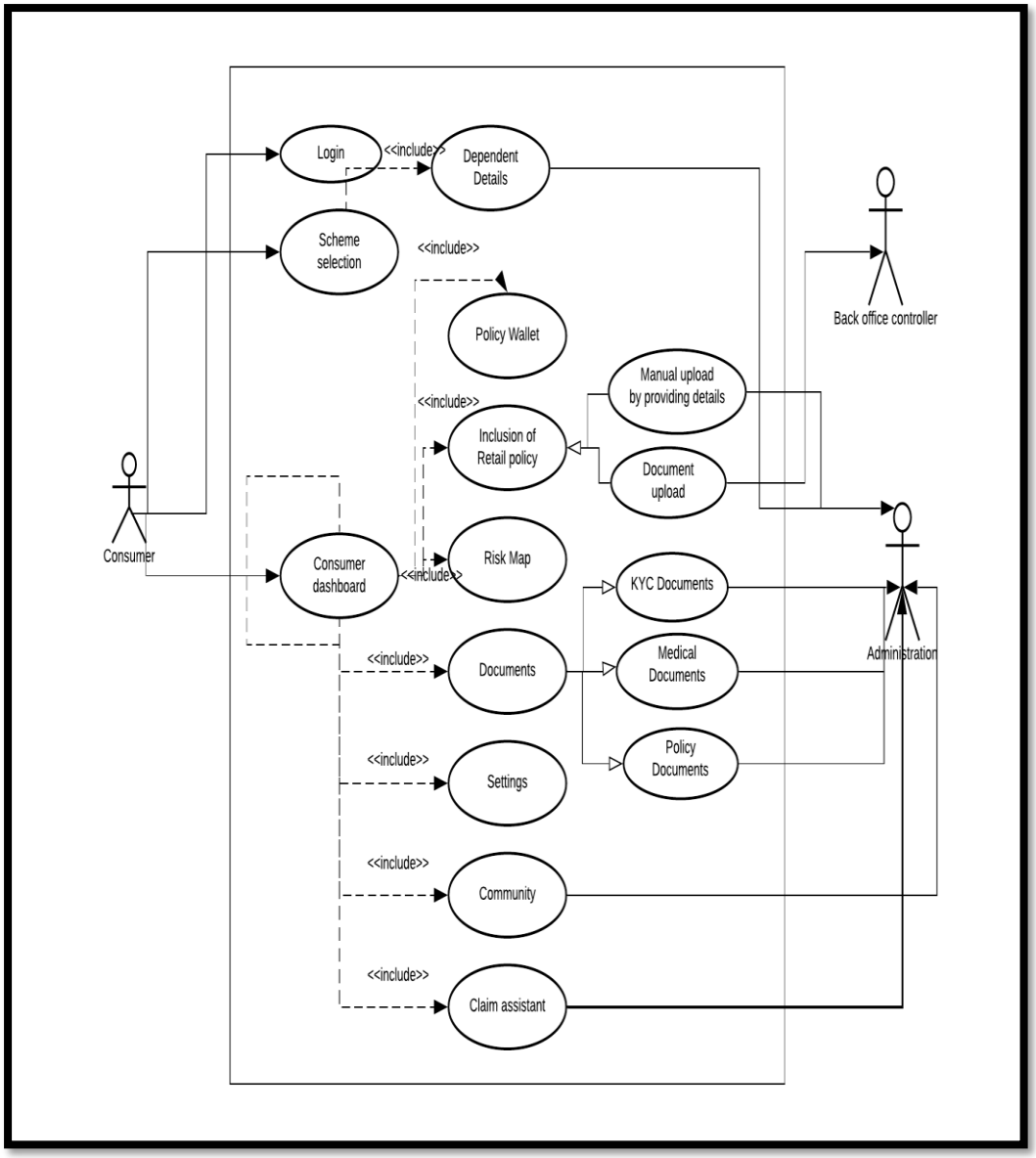


Figure 2.1 Use case model of Digital Policy Wallet

Figure 2.1 represents the use case model of digital policy wallet which consists of use cases login, scheme selection and consumer dashboard.

USE CASE DESCRIPTION

I. Login

Primary actor

The Consumer

Secondary actor

The administration for Authentication

Precondition

The Consumer must be from a valid source

Main scenario

1. In case of login, check for credentials with database.
2. New Consumer's credentials must be validated and email and phone should be verified.

Post condition

The Consumer is redirected to completion of profile page if consumer is a first time Consumer else consumer is redirected to the consumer dashboard page.

Alternative condition

If the Consumer credentials are wrong, error is thrown.

II. Scheme selection

Primary actor

The Consumer

Precondition

Consumer must be logged in and the consumer must be a first time consumer.

Main scenario

1. The consumer must provide dependent’s details.
2. May or may not provide annual income.
3. Select a desired topup plan.

Post condition

The application updates the policy enrollment schemes for the consumer.

Alternative condition

If no policy is applicable or if window period is not open for any of the policies then take the consumer to consumer-dashboard page.

III. Policy Wallet

Primary actor

The Consumer

Precondition

Consumer must be logged in.

Main scenario

1. Consumer can view the corporate policy details provided by the administration.
2. Consumer can view the personal policy details like premium and sum insured.
3. Consumer can add nominees if consumer has any term life or personal accident policy.

IV. Inclusion of Retail Policy

Primary actor

The Consumer

Precondition

Consumer must be logged in.

Main scenario

1. The consumer can add new retail policy into the policy wallet. The retail policy can either be manually uploaded or can be uploaded through a document.
2. Only valid documents like png,doc or pdf format of maximum size 10 Mb can be uploaded

Post condition

The policy gets added in the policy wallet.

Alternative condition

The policy does not get added in the policy wallet.

V. Risk Map

Primary actor

The Consumer

Precondition

Consumer must be logged in.

Main scenario

I. Consumer can view the risk score provided by the administration.

VI. Community

Primary actor

The Consumer

Secondary actor

The administration

Precondition

Consumer must be logged in.

Main scenario

For consumer interaction and as a help assistant the application provides a community board where the consumer can post questions and get answers from the administration or the other consumer

Post condition

The question gets posted on the website and other consumers can view the posted question.

Alternative condition

The question does not get posted on the website and other consumers cannot view the posted question.

VII. Documents

Primary actor

The consumer

Precondition

Consumer must be logged in.

Main scenario

1. The consumer must upload KYC and medical documents for claim processing
2. Document formats are validated

Post condition

The documents would be uploaded and saved in s3 bucket.

Alternative condition

No document uploaded.

VIII. Inclusion of Retail Policy

Primary Actor

Consumer

Precondition

Consumer must be logged in

Main scenario

1. Consumer can upload retail policy by manually providing details or by uploading a document.
2. When the consumer uploads a document digitization will be carried by the back office controller

Post Condition

Policy will be uploaded successfully and details will be displayed in the policy wallet

Alternative condition

Policy not uploaded.

2.3 SOFTWARE REQUIREMENTS SPECIFICATION

A software requirements specification is a document that describes requirements for a software product, program or set of programs. The software requirements specification represents the results of the requirements analysis and describes the requirements of the software under development. It lays out functional and nonfunctional requirements, and may include a set of use cases that describe Consumer interactions that the software must provide.

FUNCTIONAL REQUIREMENTS

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected Consumers and the type of system where the software is used. It may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

The major functional requirements of the fantasy gaming platform are as follows:

- **Login -** The Consumer should be able to login to view or add policies.
- **Scheme Selection** – The Consumer must give his dependents details to enroll his dependents to corporate policies for which the window period is open.
- **Digital Policy Wallet** – Arrange insurance policies by type and sub type with key details easily marked out.
- **Inclusion of retail policy** – Retail policies can be added in the policy wallet by providing details or by uploading pdf.
- **Insurance risk assessment** – Get a Risk Score measuring the coverage against Consumer's and Consumer's family's requirements.
- **Documents** – Consists of personal policy and retail policy documents. KYC and medical documents must be uploaded for claim assistant. Also consists of system generated ID card which consists of Insurer and TPA details.
- **Settings** – Settings consists of editing of personal details, notification and alerts can be set and password can be changed.
- **Community board** – to help connect with other Consumers and get quick help.

NON-FUNCTIONAL REQUIREMENTS

Non functional requirements describe the general characteristics of a system. Non-functional requirements may also describe aspects of the system that don't relate to it's execution. The non-functional requirements required for the application are given below.

Reliability

- The application environment should remain constant and even with a lot of changes in the package environment, sufficient fail-safes need to be kept to mitigate problems developing from software upgrades/downgrades.
- The application should logically be working forever without any problems not attributed to breaking of application.
- Easily be able to determine which piece of code/function/module is erroneous and quickly fix the same.

Usability

- This is the most important requirement scoped in the system study phase
- The application should be easy-to-use, lightweight and intuitive, without need for any training before being able to utilize the software to its full effect
- The Consumer must be able to understand the terms and conditions.

Security

- Json Web Token is used to securely transfer information between the browser and the server.
- The system should restrict Consumers to password or OTP-based Consumer-level authentication only.
- Passwords are sent in encrypted format using crypto-js algorithm while hitting the API calls.

Availability

- This application must be viewable only when a particular module can be loaded or found otherwise it must throw error.
- The API gateway must be able to manage the dependencies and should not allow the Consumer to use the application till all services are working.

Extensibility

- This requirement is a type of evolution qualities
- This application is extensible by adding any type of client devices to make use of the application

Performance

- Progressive Web Apps have the biggest impact on performance as they are responsive, progressive and work offline after initial lightweight installation.
- Angular works on the concept of single-page application (SPA) which is a web application or web site that interacts with the Consumer by dynamically rewriting the current page rather than loading entire new pages from a server.
- Angular also uses the concept of lazy loading which is a technique that allows Consumers to load JavaScript components asynchronously when a specific route is activated.
- The consumer will not lose the input while submitting the data when there is no internet connection because the concept of offline form is implemented in this application.

2.4 TEST PLAN

Test Plan is the strategy used to verify and ensure that a product or system meets its design specifications and other requirements. Test Plan is prepared by Software Testers.

TEST SCOPE

The entire application consists of multiple modules which cohesively work toward getting desired output. However, the initial concern is the test result of the following modules:

- C**onsumer Authentication** - The application must be able to login only through valid OTP or password. If a Consumer enters wrong password for 3 times the account would be locked and the consumer can recover the account after valid OTP validation and change of password
- **Scheme selection**– If a consumer is a first time Consumer then dependent details must be given so that the consumer can complete policy enrollment. Consumer can add any number of dependents. Every dependents age must be cross verified before saving it in the database.
- **Inclusion of Retail Policy** - The Consumer can add new retail policy into policy wallet. The policy start date and end date must be cross verified. The type of document that is uploaded must also be validated.

TESTING FUNCTIONS

Test functions or issue means that a testing a function or an issue to apply on software to achieve quality. All the functional features specified in the requirement analysis are to be tested. The testing functionalities are,

- System reliability
- System usability
- System security
- Data consistency
- Data representation

TESTING TECHNIQUES

The software built should be tested using functional testing. A brief description of the technique is given below.

FUNCTIONAL TESTING

Functional testing is a type of software testing whereby the system is tested against the functional requirements/specifications. Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is concerned only with the results of processing. It tests the features including the following,

- Whether result of processing the functions are accessing and picking up right data from right source.
- Whether the results meet all requirements that are specified by the Consumer according to the business rules.

CHAPTER III
SYSTEM DESIGN

System design is a set of procedures performed to convert the needs and requirements of a business or organization into a design that can be implemented on the organization's computer system. It identifies the software components, specifies relationships among components, properties of both components and relations, defines program structure and provides a blue-print for implementation. This chapter deals with the various design issues that guides the interface development of the application and gives the software design of fantasy gaming platform.

3.1 ARCHITECTURAL DESIGN

Architectural design is the process of defining a collection of structure of data, hardware and software components and their interfaces to establish the framework for the development of a computer-based system. It focuses on the decomposition of a system into different components and their interactions to satisfy functionality and performance requirements as well as non-functional requirements of the system. The software that is built for computer-based systems can exhibit one of these many architectural styles or patterns. An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture within a given context.

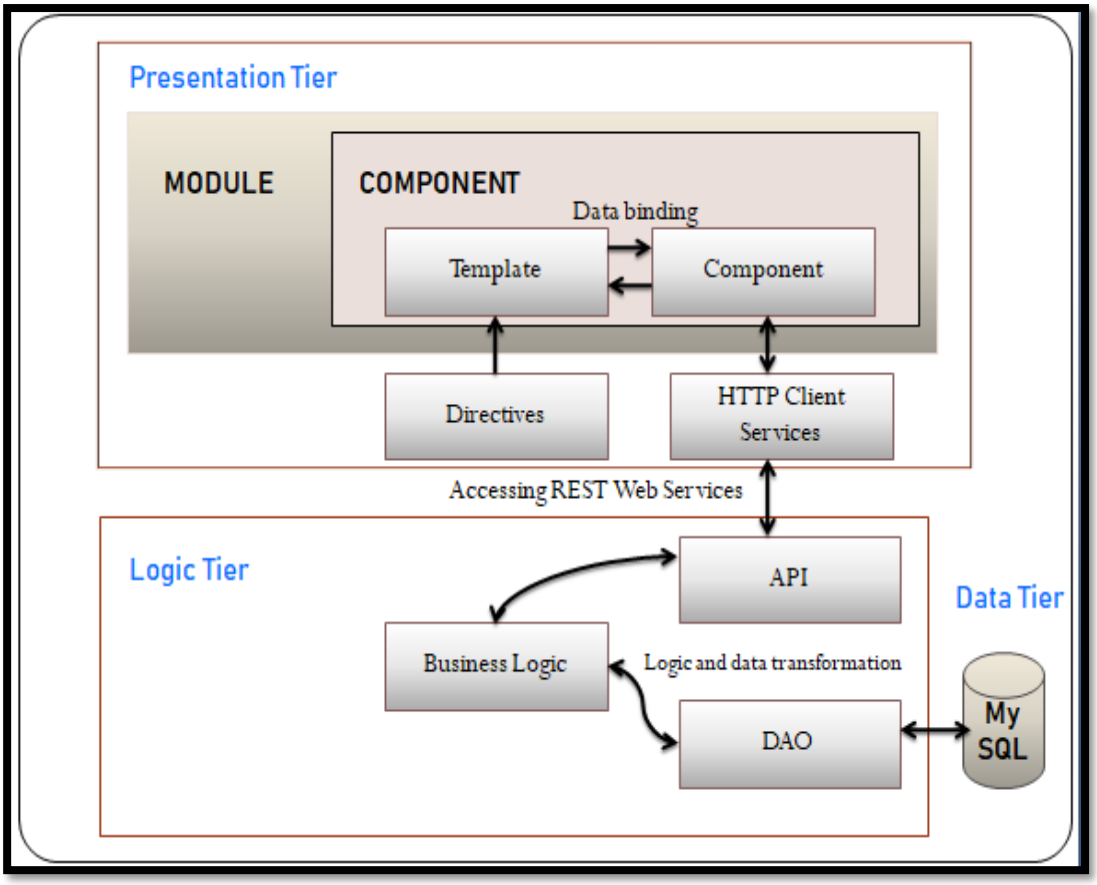


Figure 3.1 Architectural Diagram – Digital Policy Wallet

Figure 3.1 uses *3-tier* architectural model.

Presentation Tier:

The top most level of the application is the Consumer interface. The main function of the interface is to translate tasks and results to something the Consumer can understand.

Template:

Template is a User interfaces (UI), look and feel which are used to display the application's User interface (UI).

Component:

Component handles and responds to Consumer input and interactions.

Http Client Services:

HTTP client is a client that is able to send a request to and get a response from the server.

Logic Tier:

Dao:

Data Access Object. The CRUD applications, that is data to collect and process. This module represents the deepest point. This tier performs the data, it describes the entities and the access logic. Simple data access logic of inserting, modifying, deleting and displaying data, without any binding to other tiers; it's the deepest tier.

Business Logic:

Later of the identification of the data to be processed, each application has to deal with the logic of interaction between entities defined into the DAO tier. You have to combine Consumer requirements with application logic, break them up and then expose to the upper tiers simple and readable signatures. So this tier lets developers to make some processing without entering into the details of how the database is structured or what integration is underneath.

Api:

In this tier falls the logic of presentation, it represents the entry point of our application, at least from the server point of view. It deals with the management of authentication and authorization: here it is possible to determine who can perform an operation or not.

Data Tier:

Here information is stored and retrieved from a database. The information is then passed to logic tier for processing.

3.2 STRUCTURAL DESIGN

Structural design is very important to represent the working and process flow of various components of a system, database or an application. The structural design of a system is the basic step of application development. Sometimes structural design can also be used to describe the hierarchy of a system.

Module Diagram

The module diagrams are used to show the allocation of classes and objects to modules in the physical design of a system, that is module diagrams indicate the partitioning of the system architecture. Through module diagrams, it is possible to understand the general physical architecture of a system.

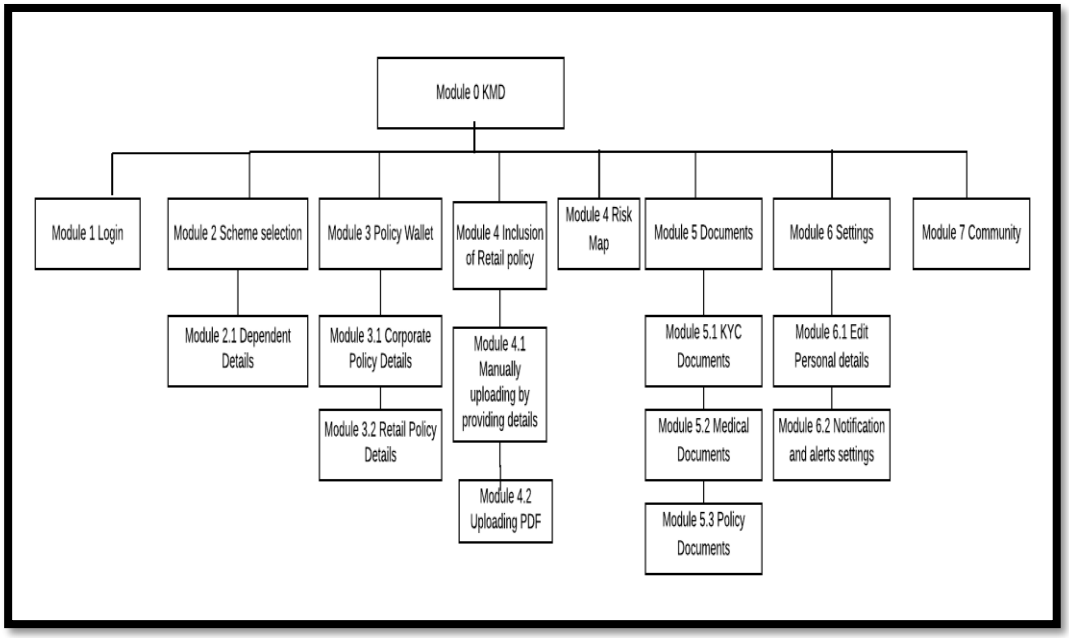


Figure 3.2 Module Diagram – Digital Policy Wallet

Figure 3.2 represents the module diagram of digital policy wallet which consists of 7 modules.

3.3 BEHAVIOURAL DESIGN

Behavioral design of a system generally represents the manner in which the system responds when a particular function is being done or when an input data is being processed at different points of time according to the system.

ACTIVITY DIAGRAM

Activity Diagrams is used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. Sequential processing and concurrent processing of activities are designed using activity diagrams. It captures the dynamic behavior of the system. An activity diagram focuses on condition of flow and the sequence in which it happens. It describes or depicts what causes a particular event. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. Both sequential processing and concurrent processing of activities can be depicted using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system. Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.

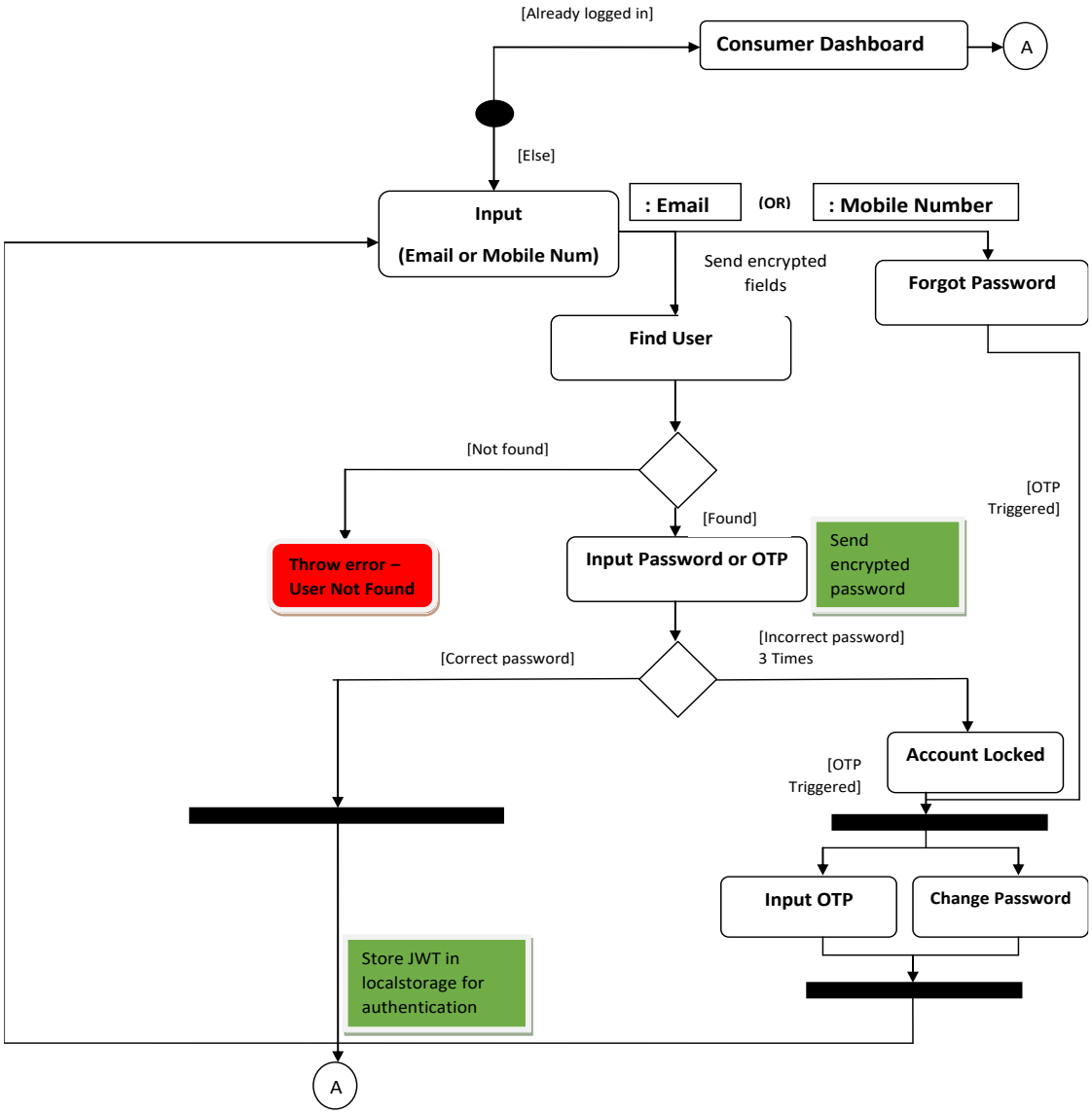


Figure 3.4.1 Activity Diagram – Login Module

Figure 3.4.1 represents the activity diagram of login module where the user can login using email or mobile number. The username and password is sent in encrypted format to the backend. On successful login the JWT is stored in localstorage for authentication. If login fails i.e., if the consumer enters the wrong password 3 times the account would be locked and an OTP will be sent to reset the password

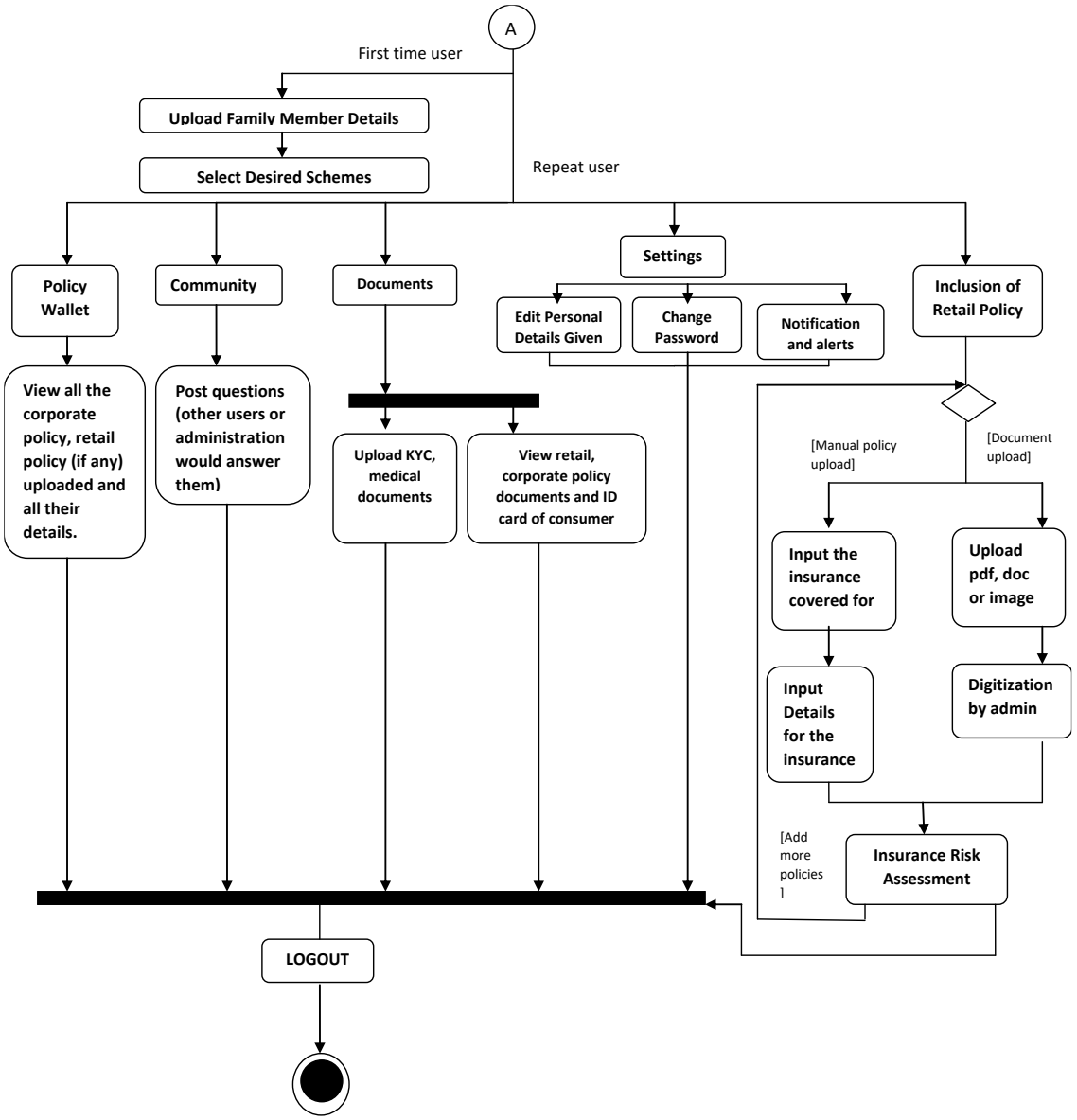


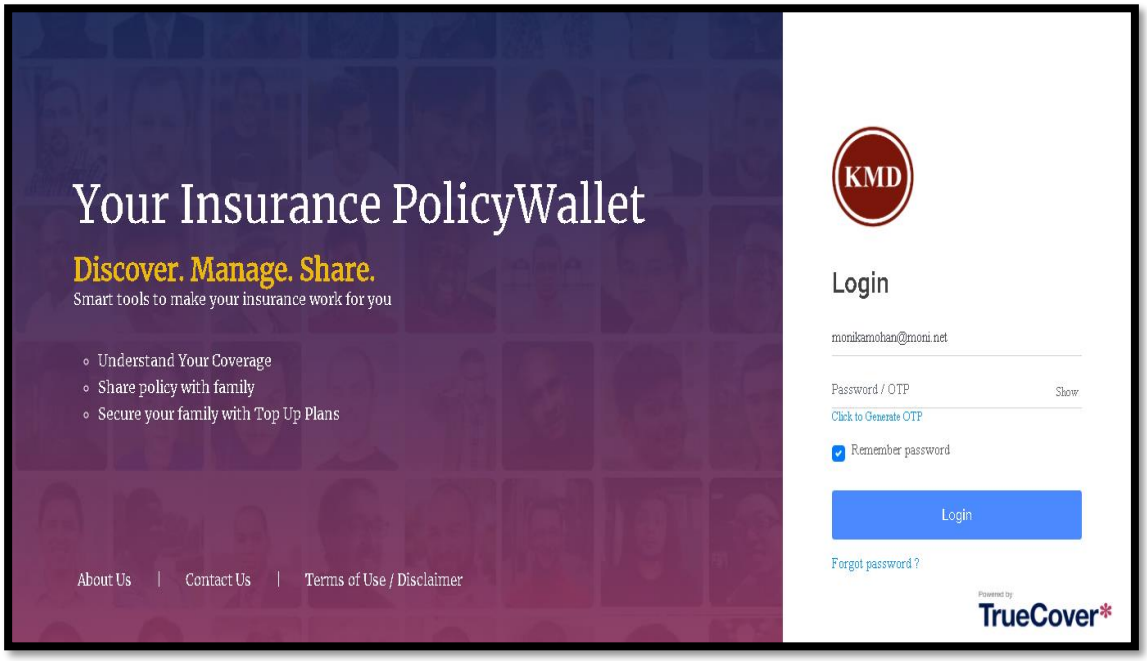
Figure 3.4.2 Activity Diagram – Consumer Module

Figure 3.4.2 represents activity diagram of consumer module where if consumer is a first time user consumer has update dependent's details and select desired topup plans (if needed) else if consumer is a repeat user consumer will be taken to the consumer dashboard which consists of policy wallet, uploading and viewing of documents, settings, community and inclusion of retail policy

3.5 CONSUMER INTERFACE DESIGN

The most important part of the system design is the Consumer interface design. The quality of the Consumer interface of a system carries the whole application forward. The Consumer interface of the proposed system has been done in HTML, CSS and Bootstrap.

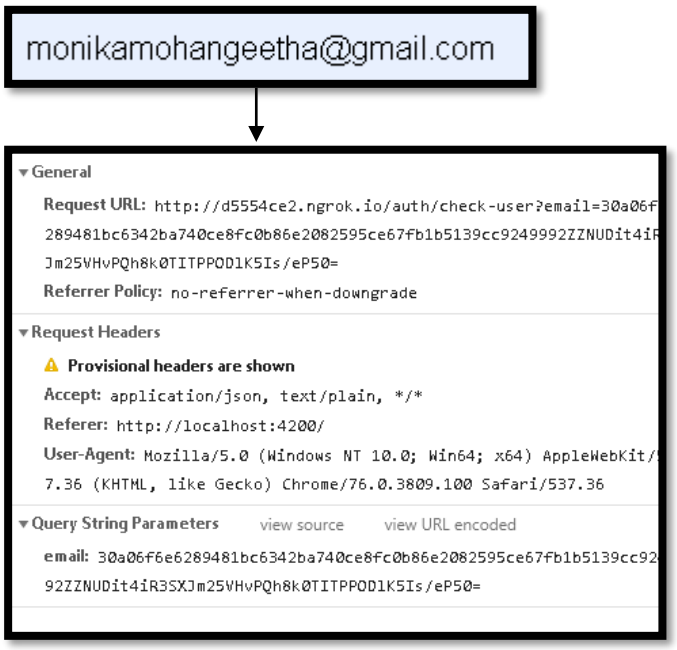
Login



Screen 3.1 Landing Screen

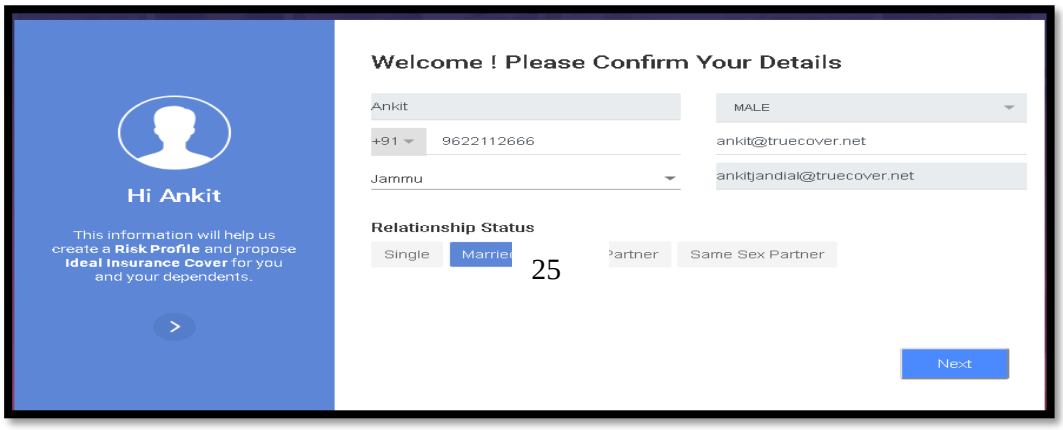
The above screen represents the first and foremost screen of the application, the landing screen which contains login.

Encryption using crypto-js algorithm:

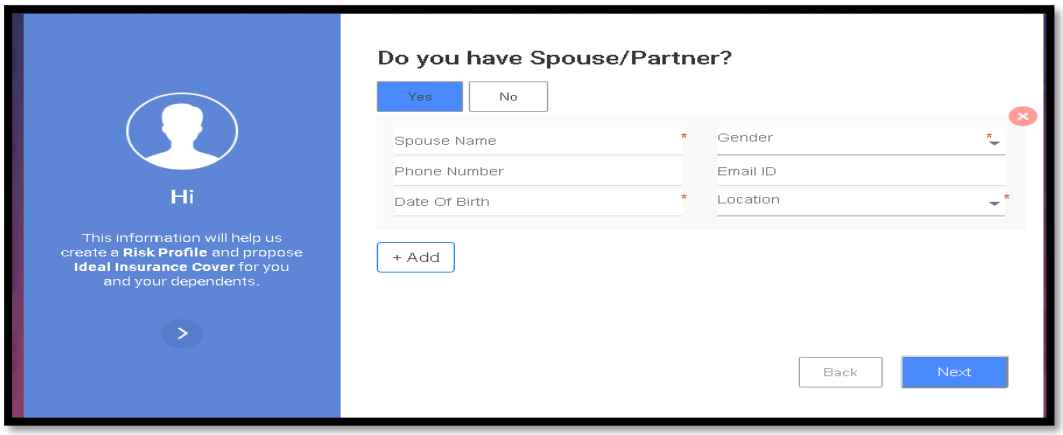


A sample input given as monikamohangeetha@gmail.com as form data is encrypted using crypto-js algorithm and the encrypted data is sent in the API calls. This encryption is used in login form and in details for scheme selection module.

Details for scheme selection



Screen 3.2.1 Personal Details



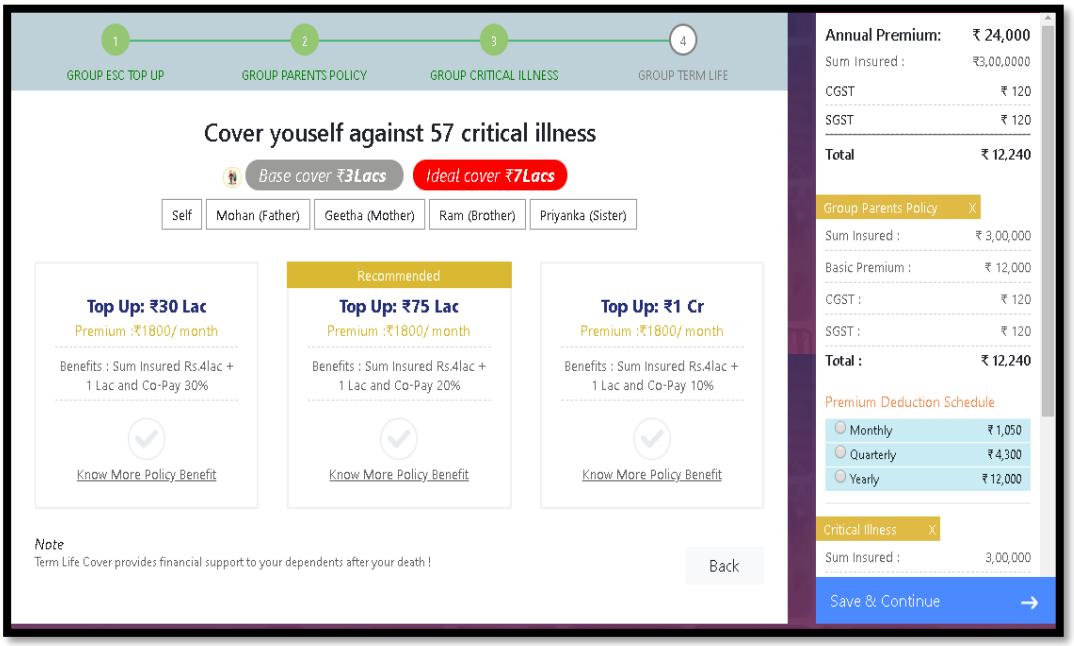
Screen 3.2.2 Details of Partner/Spouse

Screen 3.2.3 Details of Children

Screen 3.2 Details for Scheme Selection

Screen 3.2 consists of providing personal and dependent's details to enroll his dependents to corporate policies for which the window period is open.

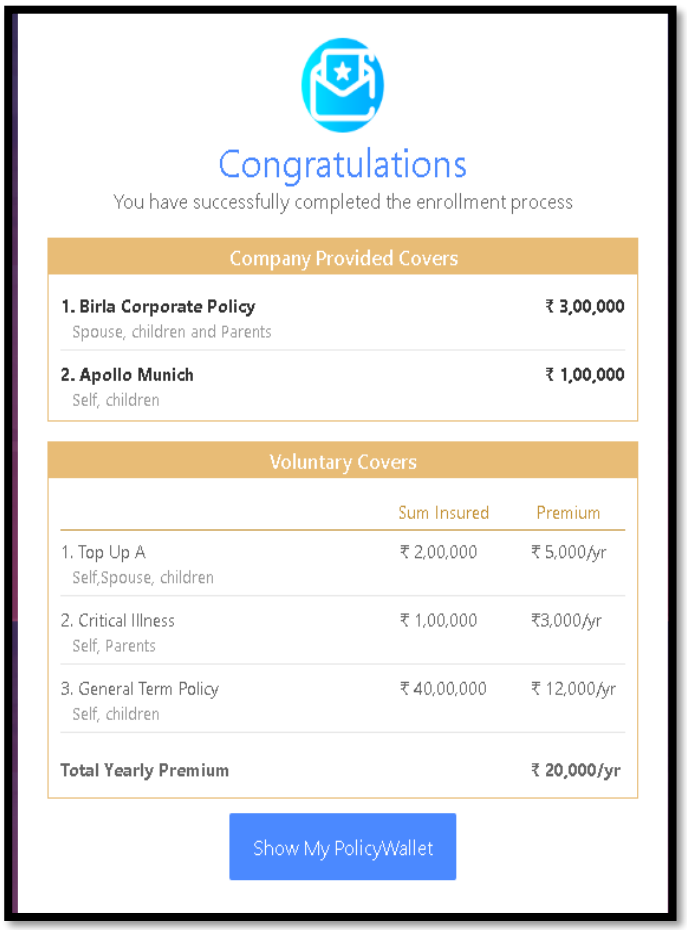
(i) Scheme Selection



Screen 3.3 Scheme Selection

Screen 3.3 shows scheme selection which consists of policy names at the top progress bar. Family members are displayed. Consumer can select the desired the family member to be included in the scheme. Consumer can add the desired topup plan from the cards displayed. The card consists of topup amount, Premium and sum insured. The recommended topup plan will also be displayed. The topup card at the right side consists of all the topup plans selected and the premium rates with SGST and CGST.

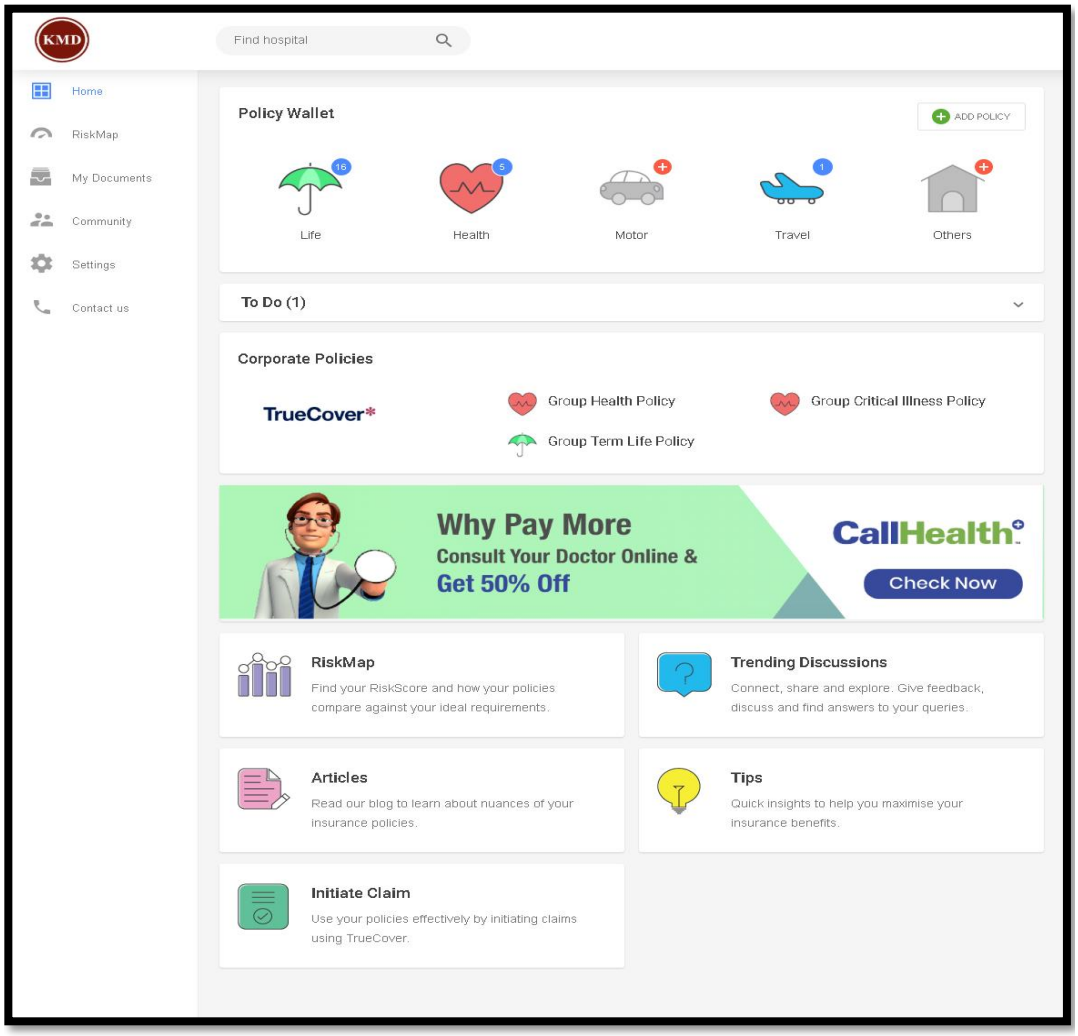
(ii) Scheme Selection



Screen 3.4 Completion Card

Screen 3.4 displays the completion card which consists of company provided covers and the voluntary covers selected by the consumer

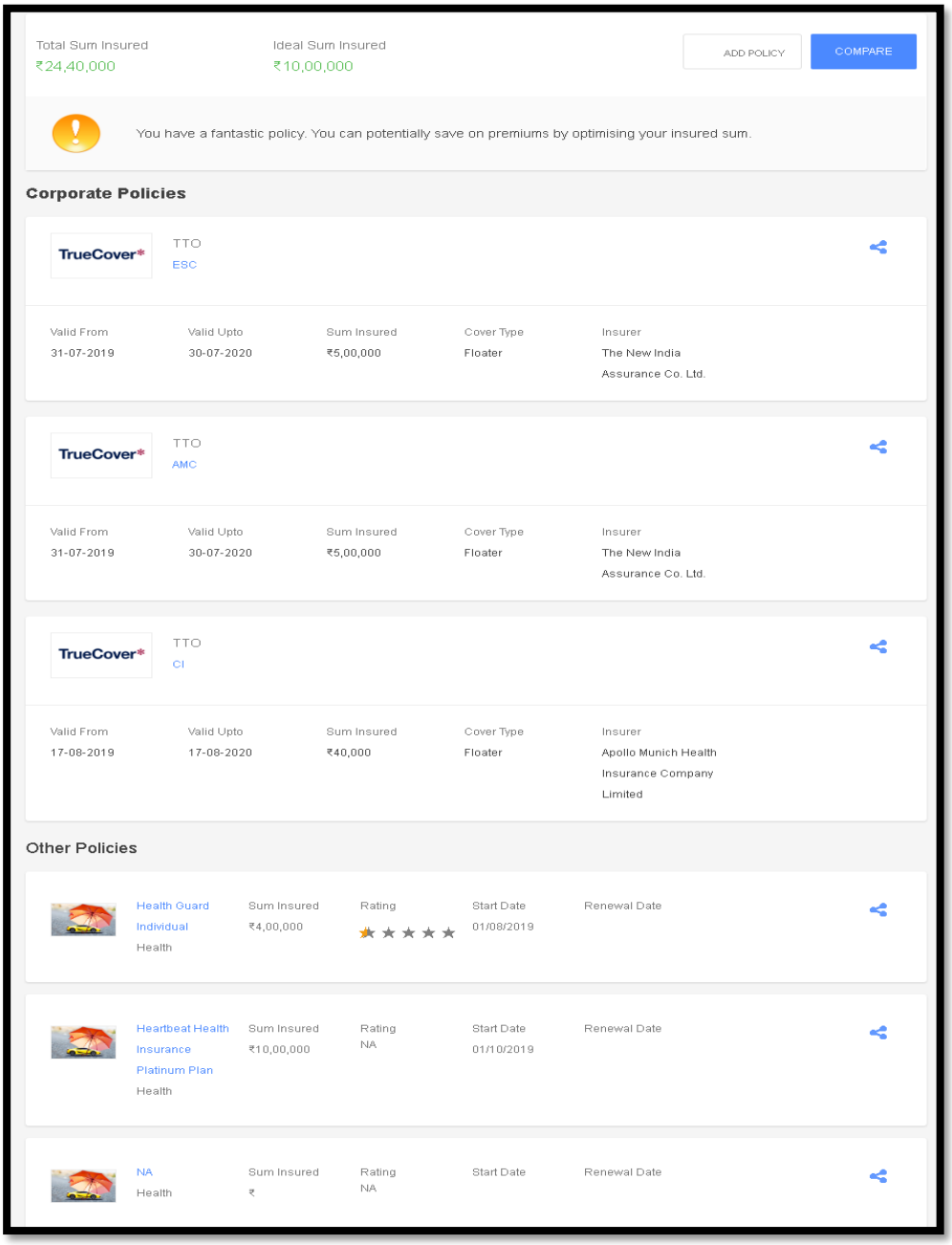
(i) Policy Wallet



Screen 3.5- Consumer Dashboard

Screen 3.5 consists of policy wallet which shows the number of policies each consumer has in every category and it also lists the corporate policies the consumer has enrolled.

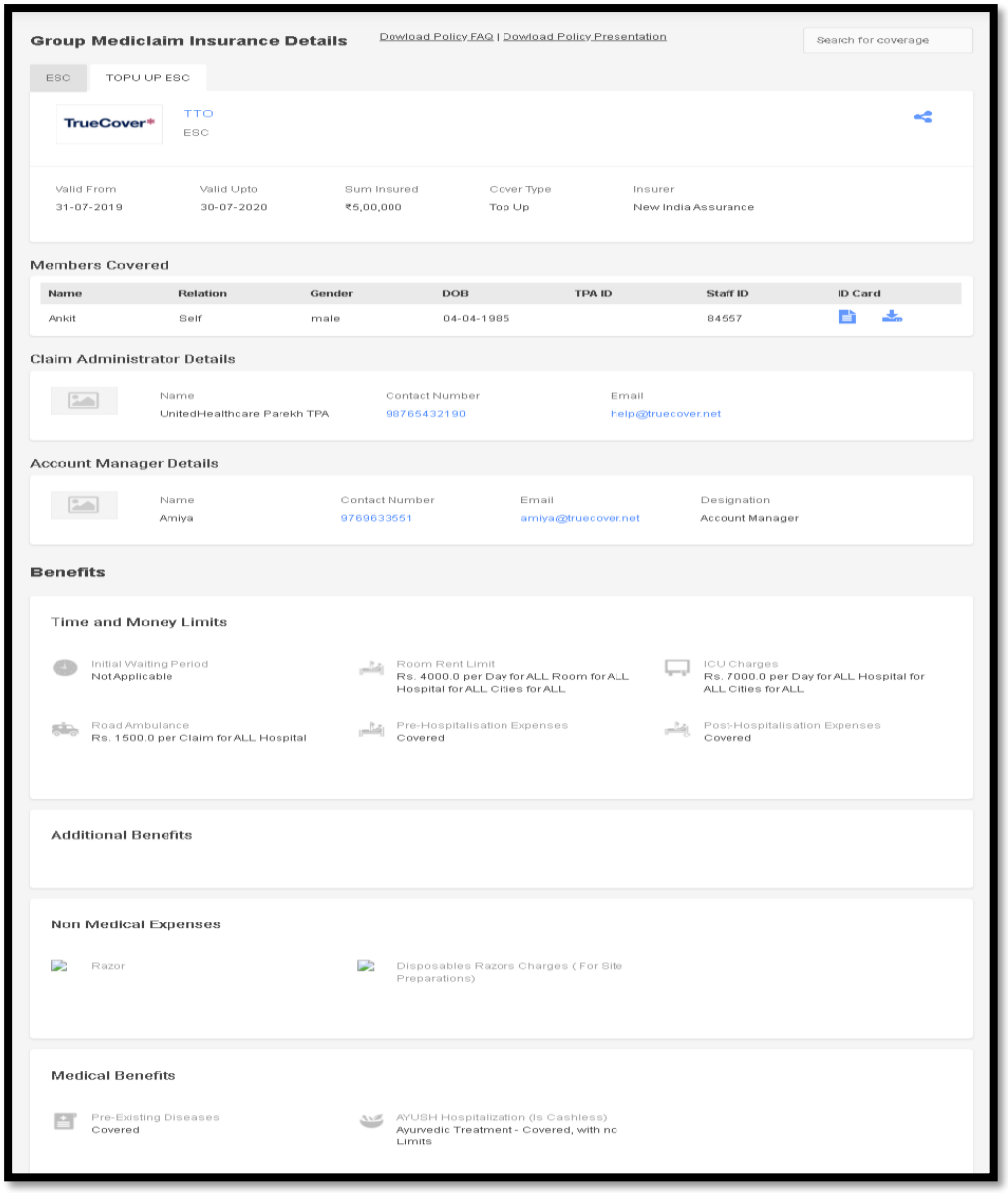
(ii) Policy Wallet



Screen 3.8 – Corporate Policies and Retail Policies Listing

Screen 3.8 appears on click of any of the categories in the policy wallet. It lists all the corporate and retail policies under the selected category.

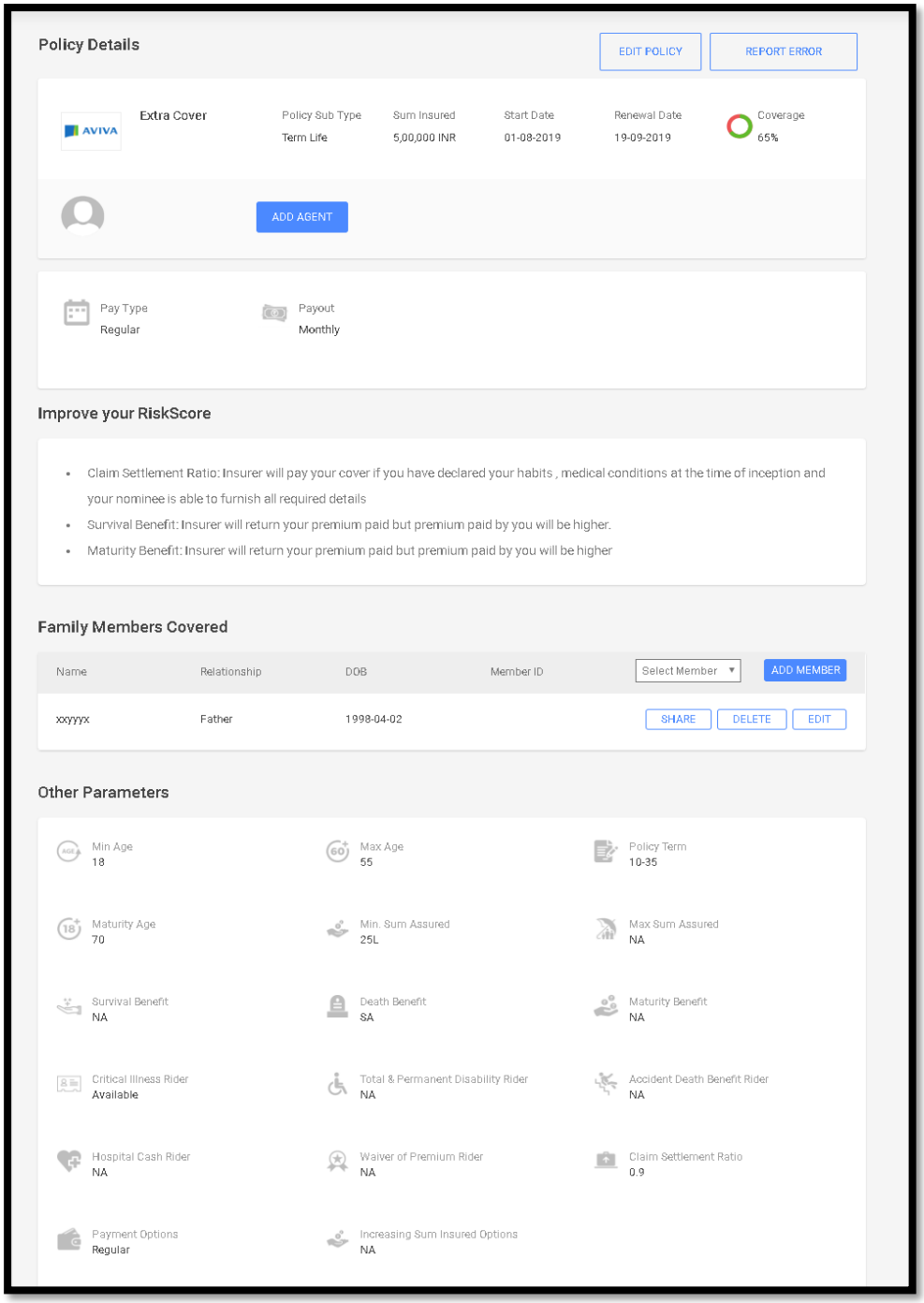
(iii) Policy Wallet



Screen 3.9 – Details of selected Corporate Policy

Screen 3.9 appears on click of any of the corporate policies listed in Screen 3.4. It displays the covers under the policy and points out the Claim Administration Details, Manager Details and the benefits of the cover.

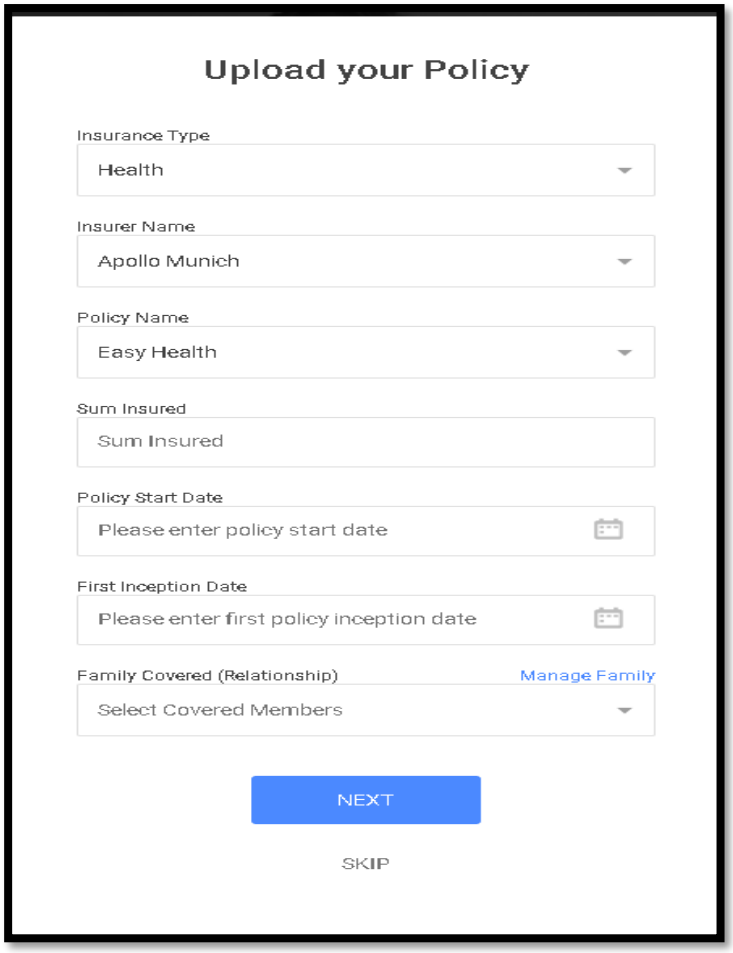
(iv) Policy Wallet



Screen 3.10 Details of selected Retail Policy

Screen 3.10 appears on click of any of the retail policies listed in Screen 3.4. It displays the premium amount, sum insured and family members covered in the policy.

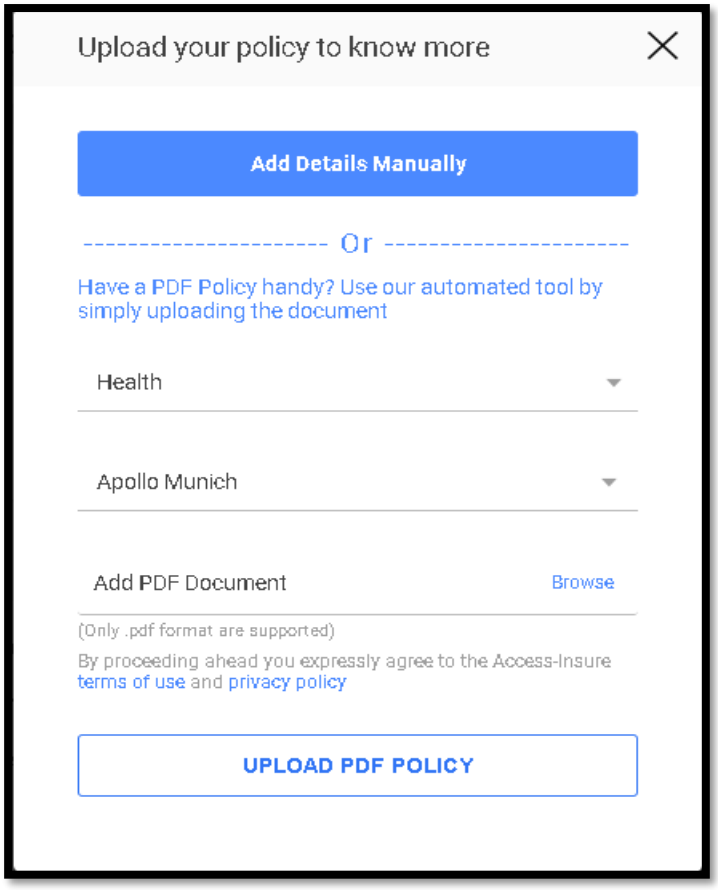
(i) Inclusion of Retail policy



Screen 3.11 Uploading Retail Policy by manually providing details

Screen 3.11 shows a screen where consumers can upload retail policy by manually providing details.

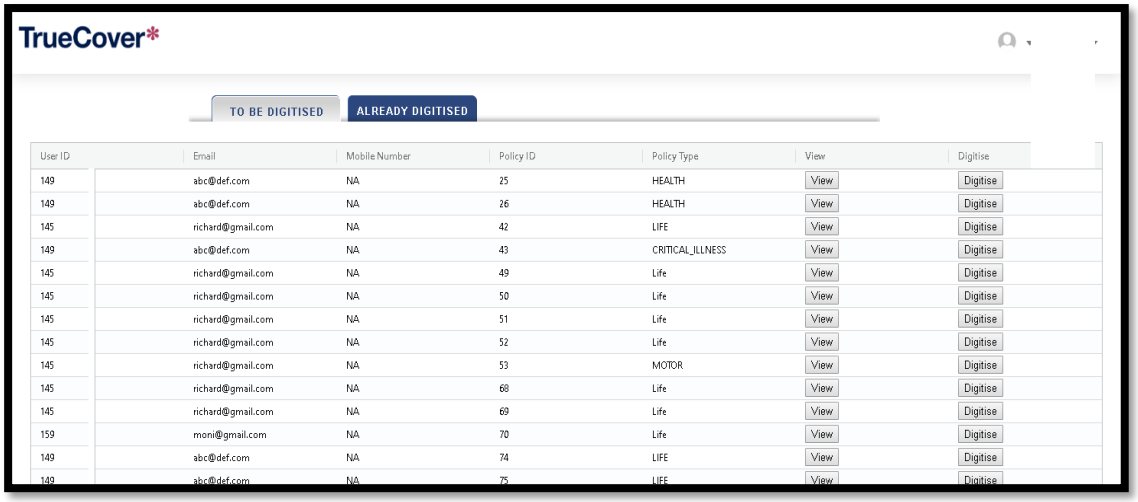
(ii) Inclusion of Retail policy



Screen 3.12 Uploading retail policies via document

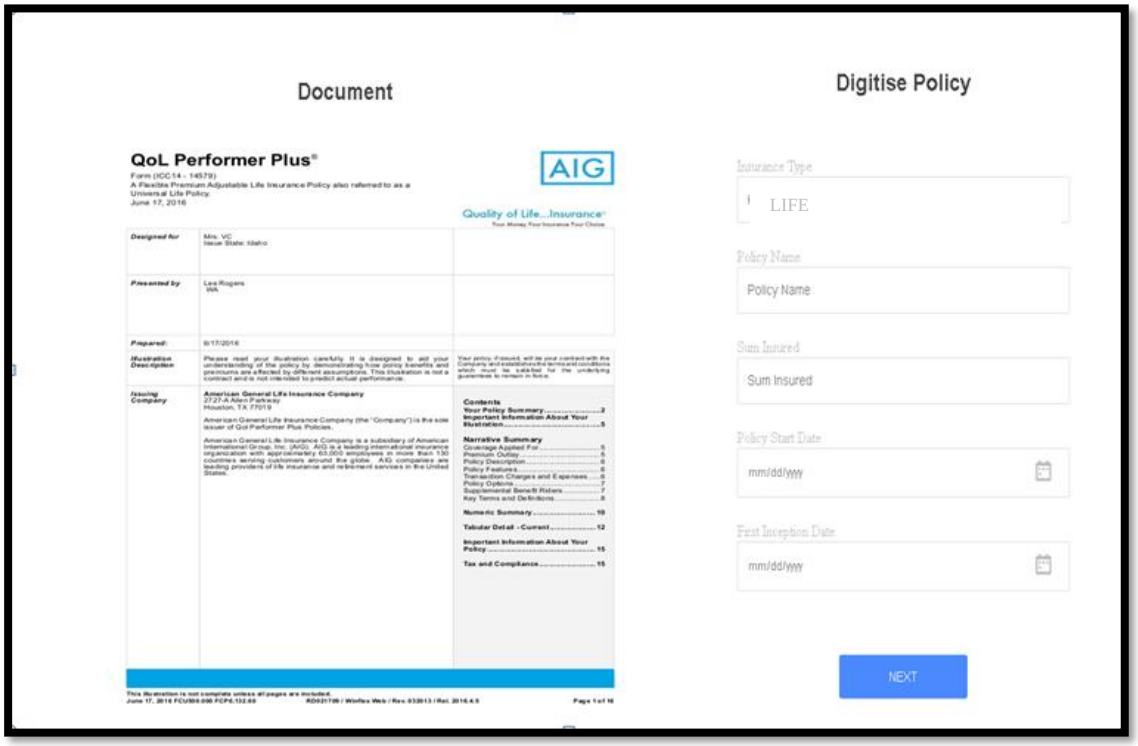
Screen 3.12 shows a screen where consumers can upload a document by pdf.

(iii) Inclusion of Retail policy



Screen 3.12.1 Digitization by Back office controller

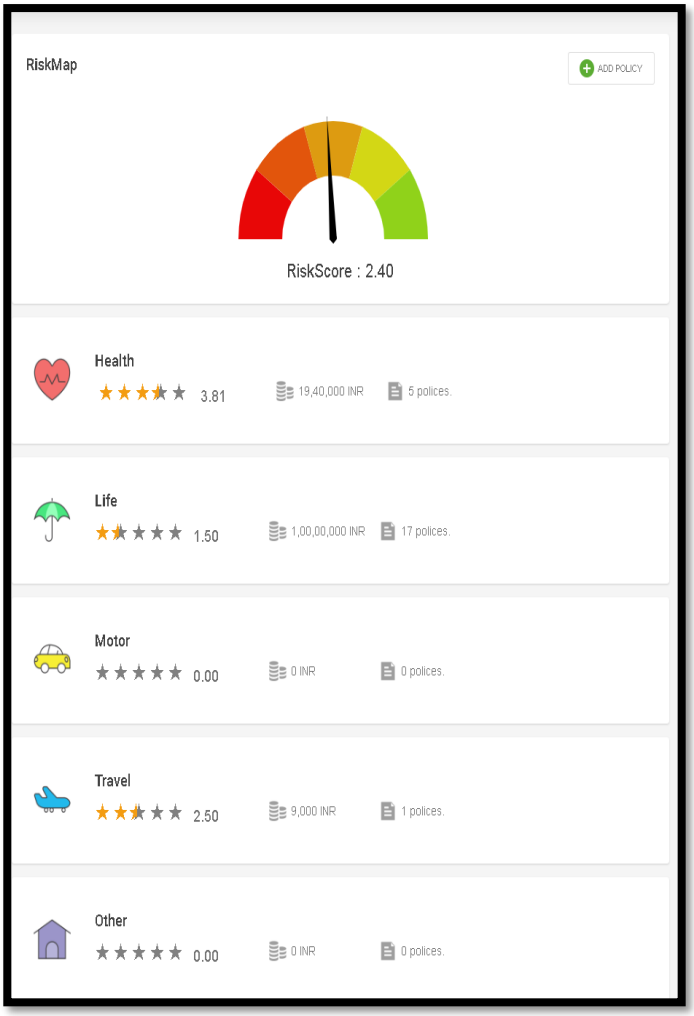
When consumer uploads a documents digitization must be carried out by where the back office controller takes out the key details like policy start date, end date etc… so that the details can be displayed in policy wallet.



Screen 3.12.2 viewing the document uploaded

Screen 3.12.2 appears on click of the digitize button shown in screen 3.12.1. The document will be shown on the left and the back office controller picks out the details needed and fills the form in right side.

Risk Map



Screen 3.13 Risk Map

Screen 3.13 displays the Risk Score measuring the coverage against consumer's and consumer's family's requirements.

ALGORITHM TO CALCULATE RISK SCORE:

Check 1:

If no policies are found return 0.0 as risk score.

Check 2:

If any policy is uploaded (because having a policy itself is a advantage) then risk score = 0.3 * 5.

Check 3:

Step 1:

Find the current Sum Insured.

Step 2:

Find the ideal Sum Insured.

Ideal sum insured is calculated based on:

1. Income
2. Age
3. Marital Status
4. Number of children
5. Parents
6. Other dependents

These parameters are listed priority wise. A multiplication factor is obtained based on the priority.

Step 3:

If Ideal sum insured is less than current sum insured i.e. the consumer is over-insured then,

Risk score += 0.4*5

Else i.e., the consumer is under-insured,

Risk score += 0.4*5*(current suminsured/ ideal suminsured)

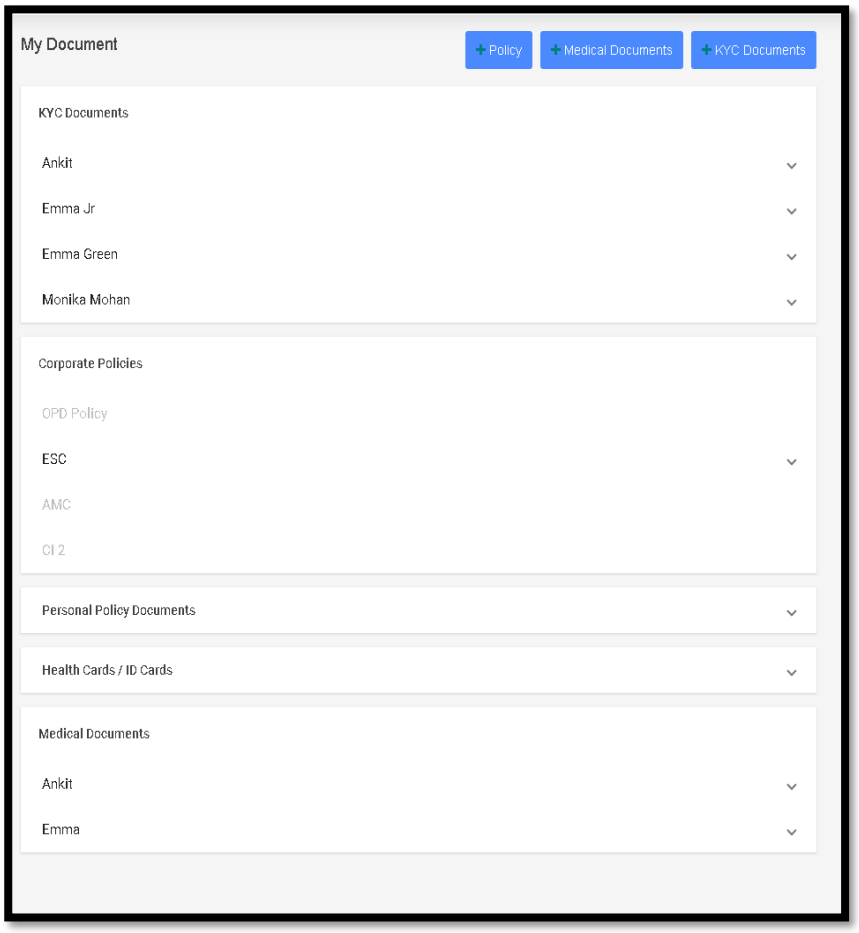
Check 4:

Risk score += 0.3*5*policyRating

Overall Risk Score Calculation:

OverallRiskScore = (0.25 * riskScoreLife) + (0.5 * riskScoreHealth) + (0.15 * riskScoreMotor) + (0.05 * riskScoreTravel) + (0.05 * riskScoreOthers)

(i) Documents

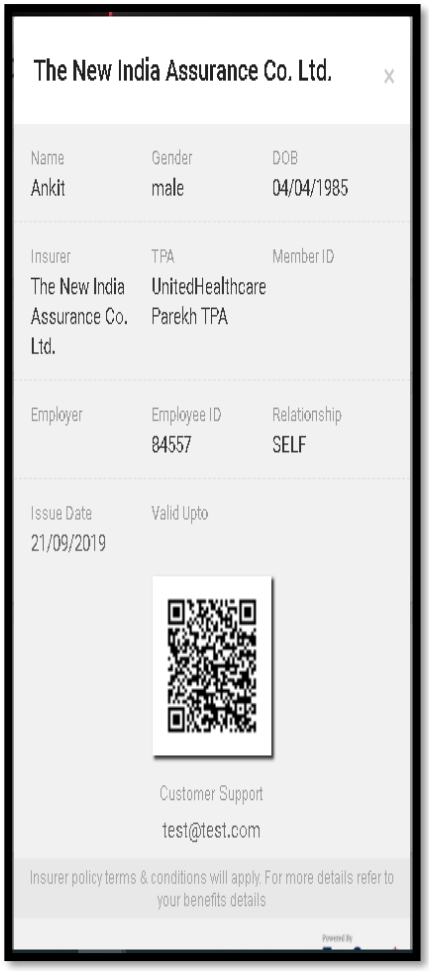


Screen 3.14 listing all the documents of consumer

Screen 3.14 consists of documents of consumer like,

- KYC Documents
    - Aadhar Card
    - PAN Card
    - Voter ID
- Corporate Policy Documents
- Retail Policy Documents
- Medical Documents
- ID card of consumer

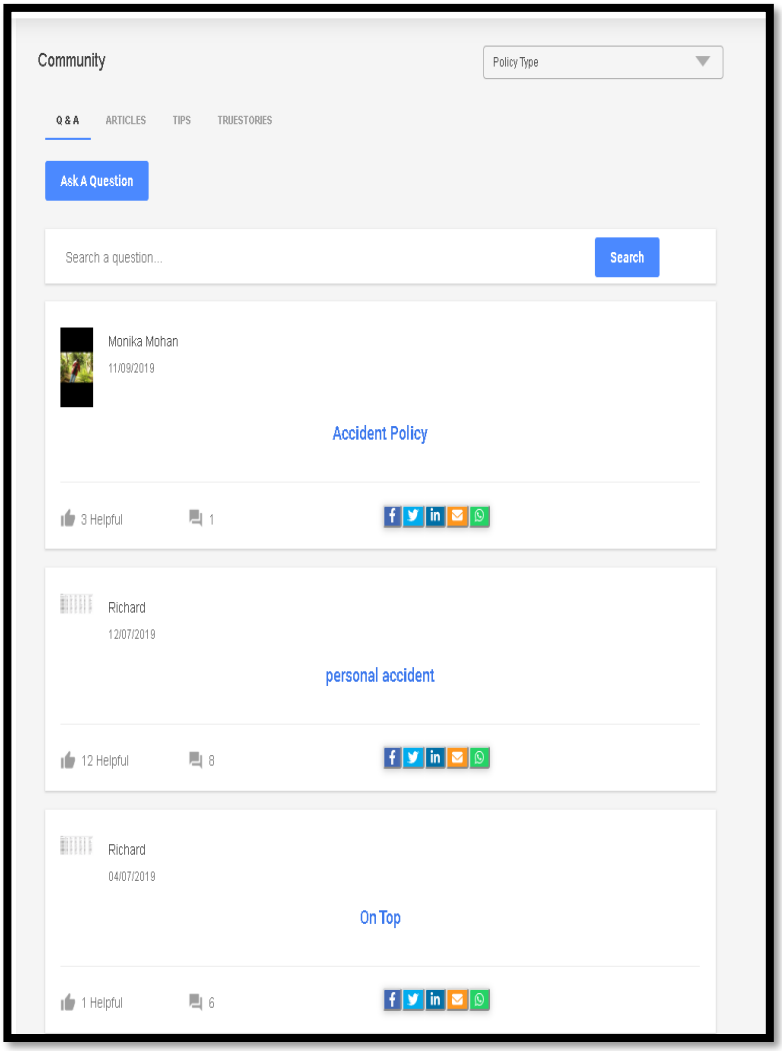
(ii) Documents



Screen 3.14.1 viewing the ID card of consumer

Screen 3.14.1 displays the ID card of consumer which consists of Insurer and TPA (Third party administrator), issue date and Employee ID

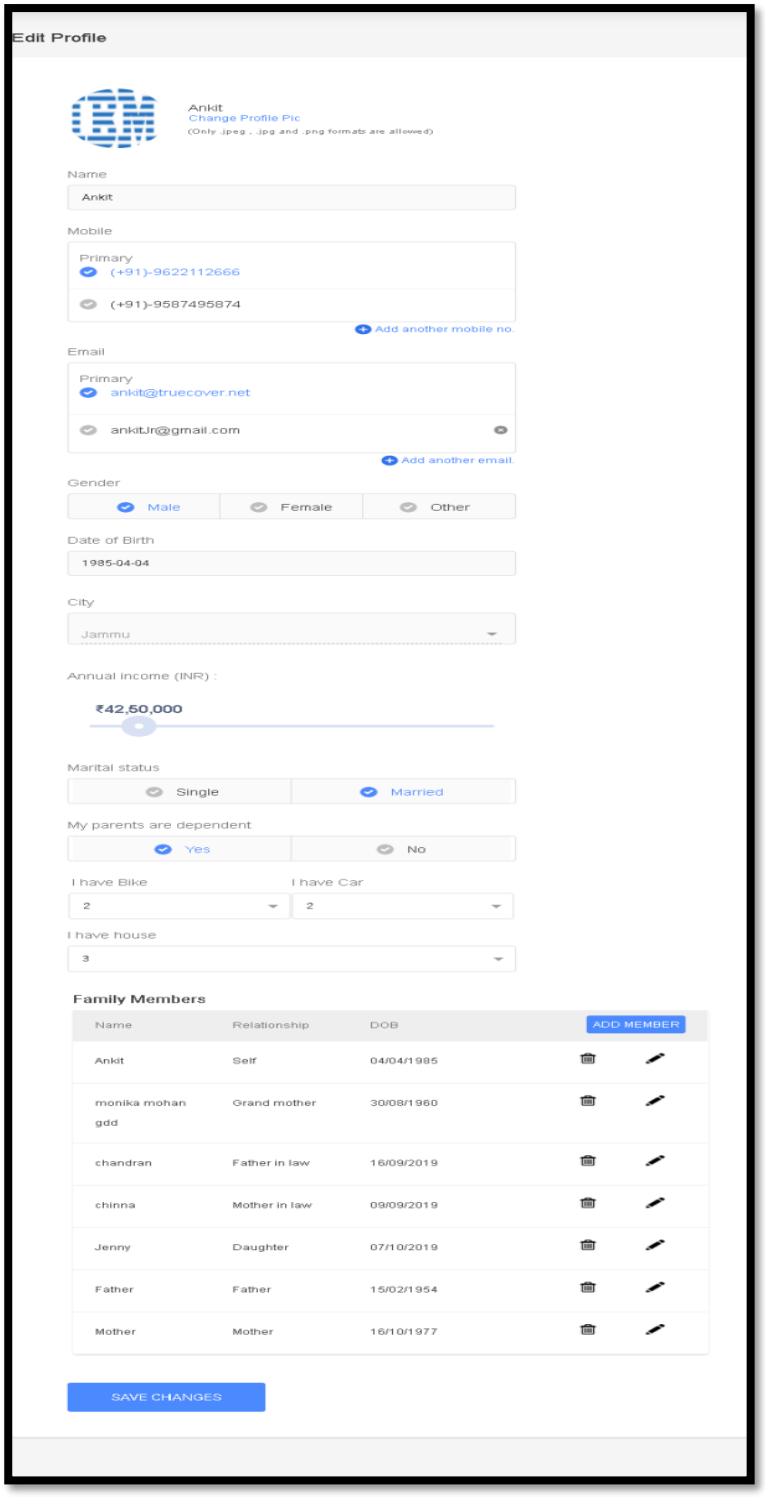
Community



Screen 3.15 Community Dashboard

Screen 3.15 shows community dashboard to help connect with other Consumers and get quick help.

Settings



Screen 3.16 Edit Profile

Screen 3.16 shows settings page where the consumer can add secondary email or mobile (the consumer can also login using secondary email or mobile), consumer can also modify his personal details and add/remove family members.

DEPLOYMENT DESIGN

Any software system must be deployed under highly favorable circumstances and environment to get optimum results. A slight variation in the implementation process may lead to errors or failure of the system. To get a better understanding of the deployment environment of the proposed system, the deployment diagram is outlined.

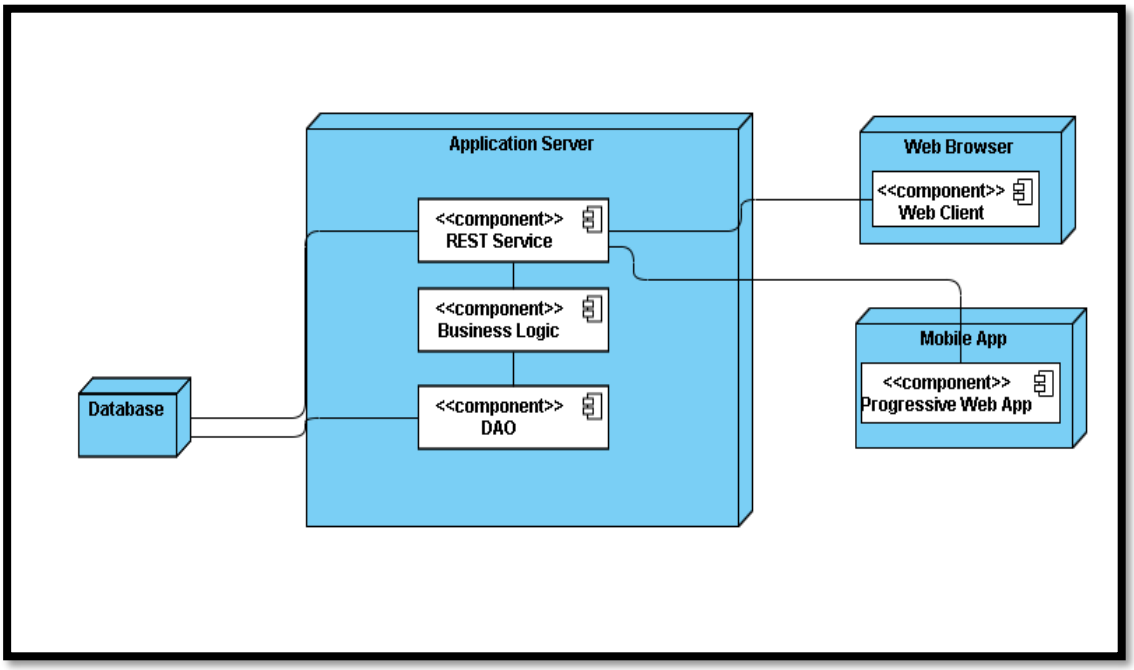


Figure 3.1 Deployment diagram

The figure 3.1 depicts the deployment diagram of the proposed system.

CHAPTER IV
TEST CASE RESULTS AND REPORTS

Testing is done on the system against the Software Requirement Specification (SRS) to ensure that the functional requirements of the system are satisfied. In addition, testing also uncovers bugs and improves the quality of the software which is included in the non-functional requirements. Testing early in the software development process ensures that any modifications later on do not affect dependent modules and produces high-quality code. The purpose of testing is the quality assurance, verification and validation, or reliability estimation. A test plan documents the strategy that is used to verify and ensure that a product or system meets its design specifications.

4.1 TEST CASES AND TEST REPORTS

The test cases are all manually written. The test cases extensively test all the classes to its fullest functionality. A test case is a detailed procedure that fully tests a feature or an aspect of a feature. Whereas the test plan describes what to test, a test case describes how to perform a particular test. The test cases are to be developed for each test listed in the test plan. Each test case contains item criteria such as Pass, Pass with exceptions and Fail

Pass

All expected results are achieved and/or all unexpected events are resolved.

Pass with exceptions

Unexpected events require alternative procedures that have to be implemented and those events are called as Exceptions.

Fail

Testing process response does not confirm the expected results.

Table 4.1 TEST CASES AND TEST REPORTS:

TestCase ID	Testcase	Given Input	Expected Output	Result
TC1	Type of username	monikamohan@gmail.com	Consumer Logged in using email	Pass
TC2	Type of username	9489165011	Consumer Logged in using mobile number	Pass
TC3	Validation in login form	948	Invalid username	Pass
TC4	Validation in login form	Mgm.com	Invalid username	Pass
TC5	Validation in change password form	truecover	Minimum length must be 8 and must contain at least one special character	Pass
TC6	Internet connectivity	Login button clicked	Your internet connection is not stable popup to be shown	Pass
TC7	Must not hit API if mandatory fields are not given	Next button clicked	Please fill all the mandatory fields	Pass

Testcase ID	Testcase	Given Input	Expected Output	Result
TC8	Parents age must be at least 15 years greater than Consumer's age	Mother's age: 11/1/2015	Please enter valid age	Pass
TC9	Policy end date must be at least one month after the policy start date	Policy Start Date: 11/5/2018 Policy End Date: 12/5/2018	Please enter valid Policy End date	Pass
TC10	Document Format check	Doc.gif	Only pdf, doc, docx, png, jpeg, jpg formats are allowed	Pass
TC11	Document size check	75Mb doc size	Maximum document size allowed is 10Mb	Pass
TC12	Deletion of primary email	Primary email: monikamohan@gmail.com Secondary email: monika@gmail.com	Primary email cannot be deleted	Pass
TC13	Session logged out	Consumer visiting the page after 3 hours	Session logged out	Pass
TC14	Family Members that are covered in policy cannot be deleted	Trying to delete spouse who is included in a policy	Spouse cannot be delete since consumer is covered in 'Selected' Policy	Pass

CHAPTER V
SYSTEM IMPLEMENTATION

System implementation is the process of converting a new or a revised system design into an operational one. It is the most crucial stage in achieving a new successful system and in giving confidence on the new system for the teams that it will work efficiently and effectively. In this phase, one can build the components either from scratch or by composition. Given the architecture document from the design phase and requirement document from the analysis phase, one can build exactly what has been requested. This chapter covers the hardware and software environment under which the proposed system will be optimally operational.

The application is developed using **Java, Angular 6 and MySQL.** All validations are done and tested effectively. The database used is MySQL where tables are created with constraints. The coding standards are strictly followed in the tool.

INSTALLATION PROCEDURE

Angular – Installation

$ npm install – To initialize the dependencies

$ ng serve –prod – To run the production build

IMPLEMENTATION OF PROGRESSIVE WEB APP:

- A Progressive Web App (or PWA) is a web application that provides a set of capabilities to give web sites an app-like experience (a site that has a banner at the bottom asking ‘Add to home screen’).
- Because of the caching strategies they can paint a picture on a user’s screen when there’s no network. This makes them more reliable.

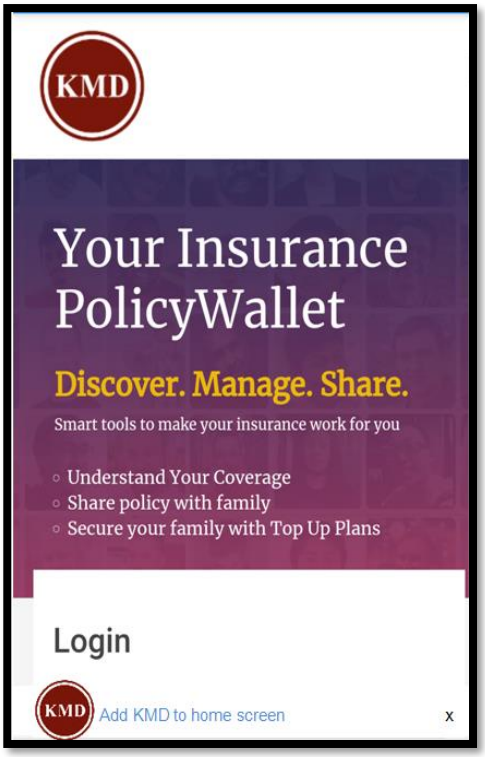


Figure 5.1 Progressive web app

Banner at the bottom consisting ‘Add to home screen’ are not real apps, but websites that behave like an installed app, also called Progressive Web App.

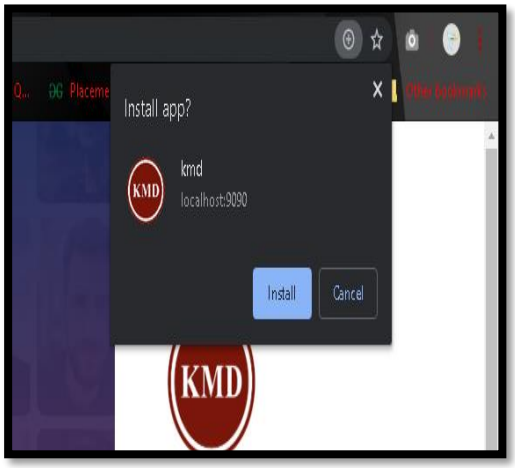


Figure 5.2 Installation of app

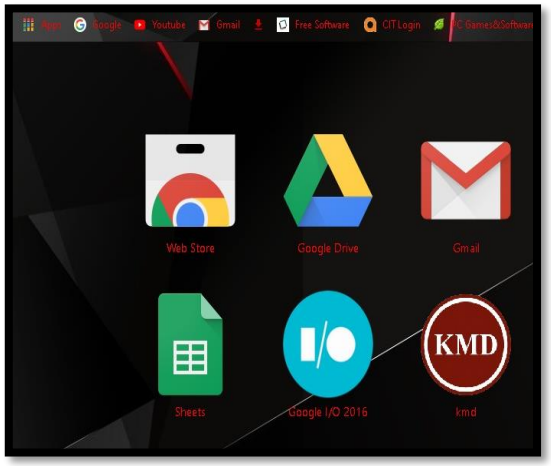


Figure 5.3 KMD App added in chrome App store

Figure 5.2 shows a prompt for installation of app and once the users install it they can view the installed app in chrome app store shown in figure 5.3

Progressive Web App Checklist

- App should have a Web App Manifest file
- App should have a working Service Worker

Web App Manifest

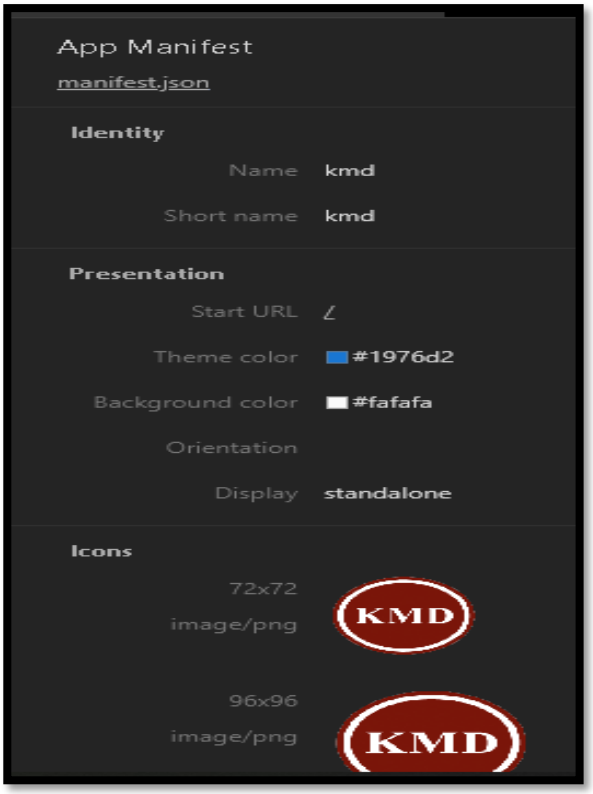


Figure 5.2 Web app manifest of Digital Policy wallet

Figure 5.2 shows a web app manifest which is a Json file with metadata about app. The content from the manifest is especially important to allow browsers like Chrome on Android phones to present an option to repeat visitors to install the web app on their home screen app on their home screen.

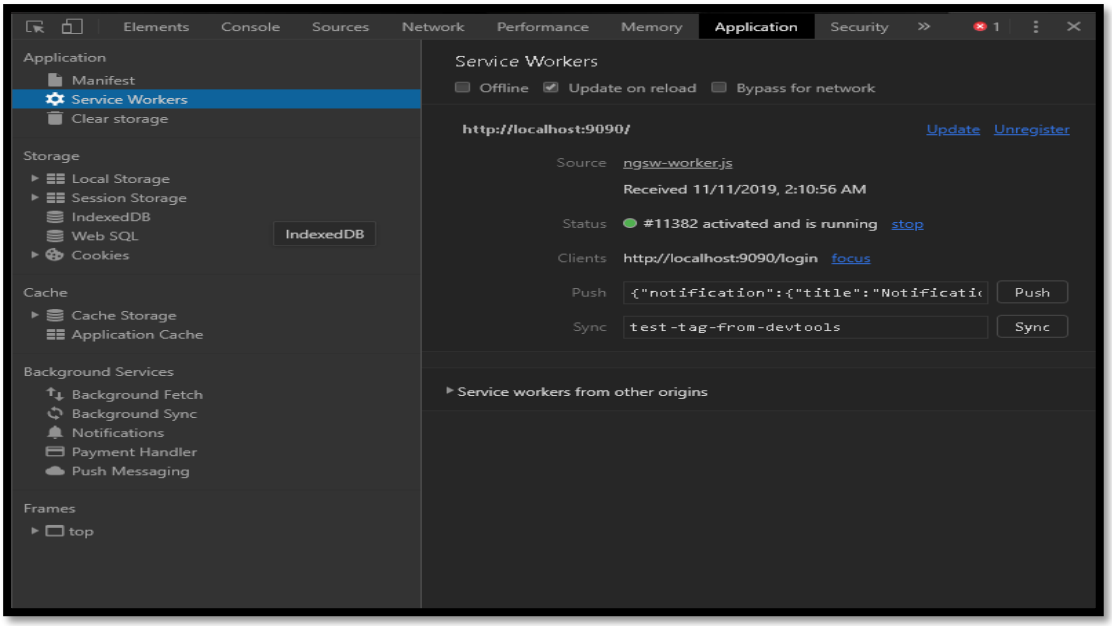


Figure 5.2 Registering a service worker

Figure 5.2 shows the registration of service worker. A service worker is a script that runs in the web browser and manages caching for an application

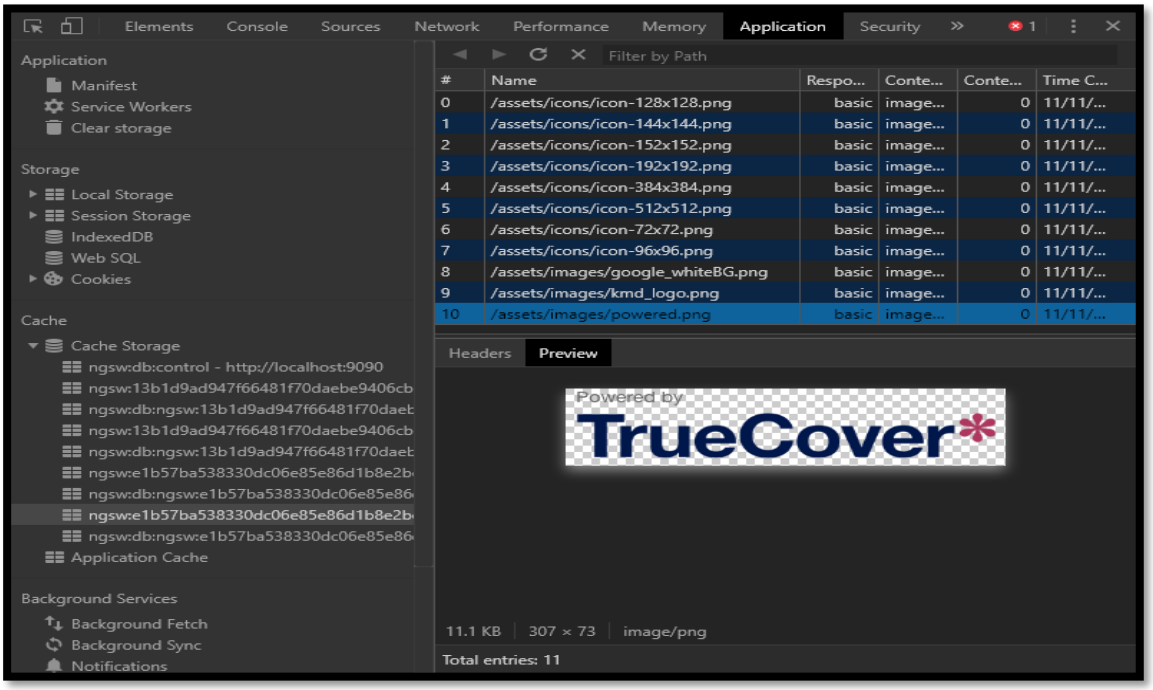


Figure 5.3 Caching by service worker

Figure 5.3 shows caching by service worker. All the css, js and images are cached and stored in cache storage during initial load.

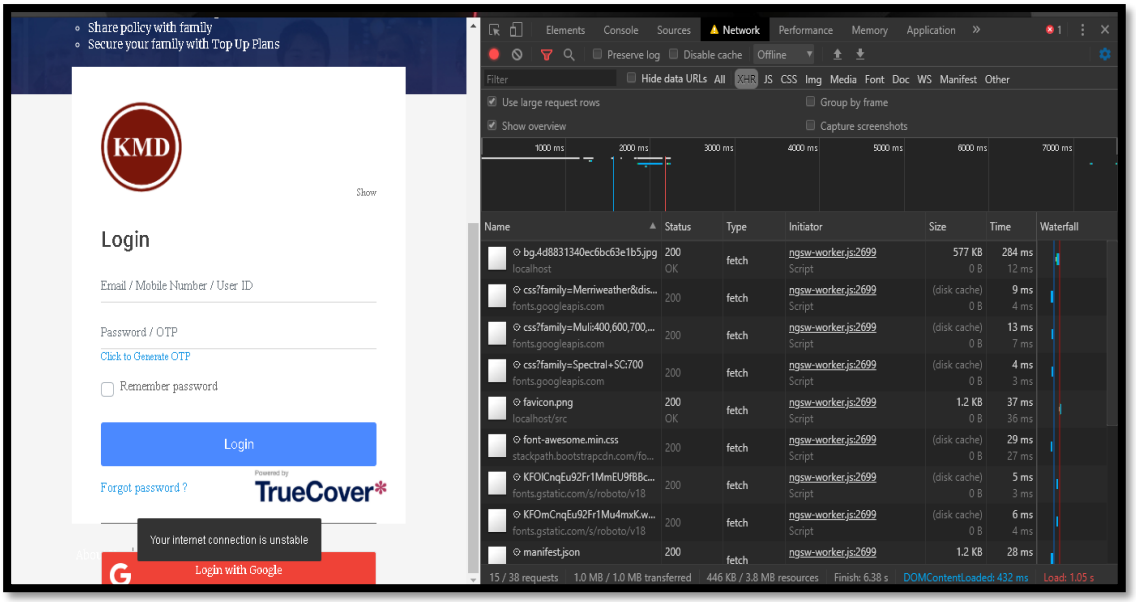


Figure 5.4 Loading the contents from cache when the user is offline

Figure 5.4 depicts that when users are offline, they see the latest fully cached version. New tabs load the latest cached code i.e. the users won't see the dinosaur showing that there is no internet connection.

IMPLEMENTATION OF OFFLINE FORM

Forms on the web don't usually play nice with bad connections. If a consumer tries to submit a form while offline, most likely the input will be lost. The form data entered by the consumer will be encrypted and stored in localstorage when the user has left offline. Now if the consumer has pressed login button while offline, a flag is stored locally and while consumer comes back online the decrypted values are sent in the API calls. The advantage is that the consumer need not fill the details again.

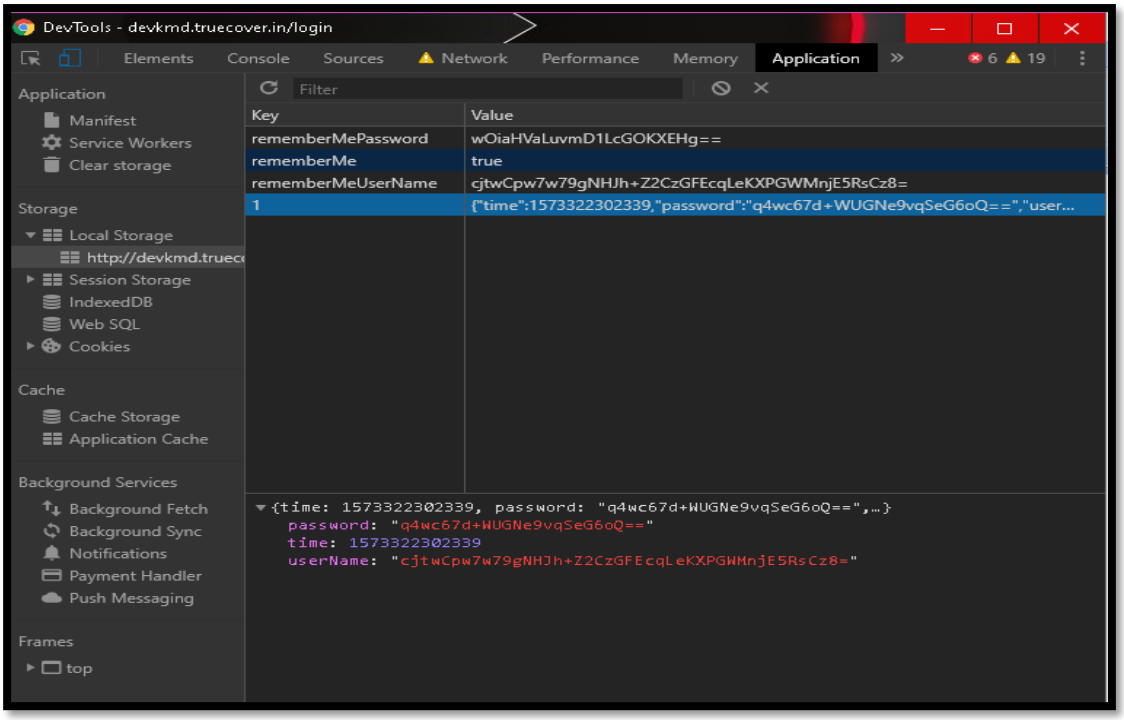


Figure 5.5 Localstorage

Figure 5.5 shows the localstorage in developer tools, which contains the encrypted username and password along with a timestamp (optional).

CHAPTER VI
CONCLUSION

Digital policy wallet helps consumers understand their risk coverage better. The product assists the consumers in making informed decisions and more effective use of their insurance. It builds an intelligent policy wallet where the Consumer can store all the retail and corporate policies for free and maximize insurance benefits.

FUTURE ENHANCEMENTS

- Truecover aims to remove the hassles and pain involved in processing claims. It also wants to eliminate the core challenge of trust in the insurance claims industry by leveraging Blockchain technology. Hence came the requirement of design and development of a version of dApp (decentralized app) taking three parties into account –
    - TPA (Third Party Administrator)
    - Hospital and
    - Insurance provider
- After the development of this new platform, signature-based certificates get attached to every document and the process gets automated via Smart Contracts, helping Consumers complete their insurance claims in approximately 5 minutes, unlike the old method which took hours and even days at times.
- “In effect, the Consumer simply scans the QR from their TrueCover dashboard and obtains the validation of the certificate as well as who issued the certificate.

BIBLIOGRAPHY

REFERENCES

Books

1. Felipe Coury , Ari Lerner and Carlos Taborda , "**ng-book: The Complete Guide to Angular**"
2. Hajian and Majid, "**Progressive Web Apps with Angular**"

Web References

1. https://www.github.com/
2. https://www.stackoverflow.com/
3. https://aws.amazon.com/
4. https://angular.io/