Addis Ababa Science and Technology University

College of Electrical & Mechanical Engineering

Department of Software Engineering

Graduation Project Documentation

Title: Customer Loyalty Program

For Hotels

Group Members

No.	Name	ID
1.	Abel Shiferaw	ETS 0028/09
2.	Alehegn Melaku	ETS 0100/09
3.	Wendm Werku	ETS 0978/09
4.	Yosef Chane	ETS 1054/09

Advisor: Mr. Abdi Mulatu

Signature________

Jan 15, 2021

ADDIS ABABA, ETHIOPIA

Acknowledgment

First and foremost, our greet thanks go to Almighty God, for his help us in every aspect of our life. We would like to acknowledge the support of all our families and friends who support and encouraged us with continuous reminders to put more effort, a determination is constantly motivating us and for their unconditional encouragement, support, and understanding which had a great value in every effort of our study.

Next, we would be glad to extend our deepest gratitude and appreciation to our advisor Mr. Abdi Mulatu for his unreserved effort in providing all the necessary guidance supervision at all steps of the work. His constructive comments and encouragements have contributed a lot to the successful completion of the study.

In addition to this, we would like to express our thanks to the management staff of **Capital hotel, Ras hotel, and Ananya** hotel for their willingness to allow us to share the requirements that could help us for the study.

Table of Content

List of Tables

List Of Figures

List of Abbreviations

CLP- Customer Loyalty Program

SMS- Short Message Service

QR – Quick Response

HTML- Hypertext Markup Language

CSS – Cascading Style Sheet

NPM – Node Package Manager

API – Application Program Interface

SDK – Software Development Kit

UML – Unified Modeling Language

CLS – Customer Loyalty System

ID – Identification Number

HTTP – HyperText Transfer Protocol

DBMS – Database Management System

FAQ – Frequently Asked Question

IOS – iPhone Operating System

URL– Universal Resource Locator

Admin - Administrator

Abstract

A customer loyalty program is an entire process that focuses on the interface between the organization and its customers. The objectives of the customer loyalty program are to enhance profitability, income, and customer satisfaction. In the current competitive business scenario, it costs a lot less to keep existing customers than to earn new ones.

One of the best ways to keep customers coming back for more is by establishing an effective loyalty or rewards program. The use of loyalty programs as a powerful tool of relationship marketing is becoming popular to encourage customer loyalty.

The main goal of our study is to obtain a deep understanding of the impact of customer loyalty programs on customer retention and attracting new customers with minimum cost in most Ethiopian hotels.

Chapter 1

Introduction

1.1. Background of the Project

Loyalty and rewards programs have become commonplace in today’s society. Both traditional brick and mortar stores and eCommerce sites are using them to boost retention and drive down acquisition costs.

In this topic, we have discussed and reviewed different kinds of literature to explain our project’s background and as a result, we have gotten the following histories which define how loyalty programs started and evolved.

The very early beginnings (the 1700s)

Some believe that the early roots of customer loyalty programs started with “premium marketing” in the late 18th century. At this time, American retailers began to give customers copper tokens with purchases that could be later redeemed for products on future purchases [1].

Transition to other forms of loyalty (the 1800s)

Retailers quickly learned that using coins as a customer retention tool was a costly way of promoting loyalty. In the late 19th century (around 1891), the transition to less expensive tools began when stamps replaced coins in “loyalty programs” [1].

Green Shield stamps became one of the first retail loyalty programs, and could arguably be considered a predecessor of what popular programs like Air Miles and Aero plan do today. Green Shield awarded stamps for purchases at select retailers that could be later redeemed for catalog products.

Modern loyalty programs (the 1900s)

Loyalty programs would continue to be led by stamp collection programs into the early 1900s. Beyond this point, individual brands and retailers began to introduce their ways of engaging customers, with "box tops" as one of the original brand-specific programs [1].

Boxtops were coupons that were printed directly onto product packages that could be later redeemed for premiums or rewards. Betty Crocker introduced their box top program in 1929, laying the framework for loyalty programs as we know them today [1].

The late 1900s saw the birth of one of the most well-known loyalty programs ever created: Frequent Fliers. Often regarded as the first full-scale loyalty program of the modern era, American Airlines' launched its Frequent Flier program in 1981. The program revolutionized customer loyalty and now boasts over 50 million members in its revamped Advantage program [1].

Card-based retail loyalty programs also gained popularity in the 1990s. Retailers were looking to implement in-store loyalty that was easier to monitor than the collection of stamps or branded currency, and these types of programs are still popular today! Just look in your wallet identification number (ID) be willing to bet there's at least one loyalty card! [1].

Loyalty Programs Today

With the rise of e-commerce and digital payment infrastructure, it is now possible for anyone to start a reward/loyalty program for their store or site. A loyalty program is no longer reserved for the power players in the retail space many of the most successful programs are online and on mobile [1].

Loyalty programs have evolved throughout history. While they once existed simply to reward customers for spending a certain amount at a store, today they are rewarding customers for a variety of desired actions! From points for social media shares that extend the marketing scope of a retailer to customer referrals that drive new customers and grow revenue, the options for customer rewards are almost endless.

Throughout history and in a modern context, loyalty programs have always been about moving customers along their customer journey. The customer journey describes a series of steps the average customer moves through in their interaction with your store, and your loyalty program should be designed to offer rewards and incentives that move them along.

Ecommerce loyalty programs have eliminated the need for physical cards and have made program adoption much simpler. So, is this the final chapter in the history of loyalty programs? Unlikely! Customer loyalty will continue to evolve and become more powerful and dynamic in years to come [1].

1.2. Statement of the problem

1.2.1 Existing System

There are a lot of customer retention mechanisms that are implemented in the world. Customer loyalty program is one of them, and a lot of loyalty program exists and implemented in an efficient and organized way in various place mostly outside Ethiopia. In Ethiopia, most organizations especially, hotels have no formal system or program which gives reward or benefit for their customers to attract new customer, retain the existing customer and engaging customers who are going away from their service.

Instead of having a system used to retain and attract customers, most hotels use television, radio, and/or broacher for advertising their services, events, and special days. And most hotels have no method to retain their frequently visiting customers apart from giving handshakes and hot greetings.

1.2.2 Major problems of an existing system

As we have mentioned in section 1.2.1, in Ethiopia there is no formal customer loyalty program (CLP) in which organizations use. So, it is very difficult for the organization to retain their customers for a long time, and also the cost of advertising their services, events, and holiday programs is very high. It is also difficult to understand the behavior of their customer for serving them as per their behavior. These problems are deeply described in section 1.2.2.1.

Here are the problems that our project solutions in the business relationship between customers and hotels.

1.2.2.1 Problems from the hotel’s side

Now a day’s many hotels are competing with one another to attract new customers and to retain the existing customers for a long time to increase their income. But the main issues are the cost of advertisement to attract new customers and the method to retain their customers. Some of the problems from the hotels perspective are discussed below:

Hotels don’t predict usable and satisfactory services

To ensure the satisfaction of a customer, organizations need a deep understanding of users’ data, but most Ethiopian hotels don’t even know or do it manually with simple observations. This creates an unbalanced product and customer ratio. Generally, no system analyzes the customer’s data and generates results about what is needed more and what is not needed.

Hotels don’t give recognition to their loyal customers

Even though no system identifies who is loyal customers or not, some customers may use the same hotel over and over again but the best thing they can get doesn’t pass from handshakes from waiters or a manager. This is one way to treat customers but not much satisfactory for both of them because there might be other hotels that will do better than them. There must be a system to recognize these loyal customers and give appreciation so they can recommend the hotel and stay satisfied in their time with the hotel.

Hotels don’t keep track of their customers

Hotels don’t keep track of their customers. This makes their business vulnerable to lose because it may take time to realize there is a fluctuation of customers or there is a problem with services. If there are no loyal customers, they may not know exactly what the customers need in that particular area.

Hotels don’t have simple ways to announce new services or events

Hotels don’t have any specific system to announce their new services or upcoming events. Hotels announce these kinds of things using radio, television or Banners which is a waste of money especially for the small hotels or another type of company, for the hotels that don’t use this kind of technique, it may take time for customers to discover new services or they will miss events. This makes a customer or others who didn’t know the hotel before will not be happy.

1.2.2.2 Problems from the customers’ side

Customers that do not participate in the loyalty program may lack recognition from the organization he/she frequently served. Besides, customers may not know new services, events, and other special gifts provided by the hotel at specific times.

This might not be a problem, but these make customers not feel good about what they buy. Every average-class person will think to spend less and gain more. By implementing a Loyalty Program, a customer gets benefits from the company in terms of incentives, rewards, and other things that are used as rewards. The main things that the customer want from the hotel to stay for a long time apart from getting satisfactory services are:

- Recognition
- Special Attention and
- Benefits

1.2.3. Proposed system

We have proposed a web-based and mobile application which solves the problem of both hotels and customers. For hotels, it allows them to advertise their services at less cost through the system and the system also allows them to understand the behavior of their frequent customer and this, in turn, helps them to retain the customers for a long time by giving rewards and discounts for their customer based on their behavior. And also, this system allows the customers to get benefits through rewards and discounts from the services they have used frequently. Generally, the system we proposed creates a win-win situation for both hotels and customers.

The mobile application is used for customers. It allows them to see the list of menus about the services and to give feedback for the service they use. On the other hand, the web-based application is used by both the hotel administrator (admin) and the customer. It gives the same access to customers. But it allows the admins to post services, advertise their services, and to generate reports and summaries.

1.2.4. Advantage of the proposed system

The proposed system has the following main advantages:

- **Customer retention** - The primary motive behind a loyalty program is to retain customers by rewarding them for their repeat purchase behavior.
- **Relevant customer data and consumer trends -** Consumer data gets recorded in the hotel's database as soon as a customer registers for a loyalty program. Hotels use this data for rewarding and offering their loyal customers.
- **Attracting new customers** – This system helps hotel owners to attract new customers.
- **Better customer communication-** A loyalty program offers a direct line to customers, making communication much easier. Aside from announcing new products/services, promoting sales, and giving and receiving feedback to ensure customer satisfaction.

1.3. Motivation

Every business depends on the existence of the customer. But it is difficult to attract new customers and hold them for a long time. The organizations must concentrate on preventing “customers-exit" since the cost of attracting new customers is higher than the cost of retaining existing customers.

The key to customer retention techniques is customer satisfaction and loyalty which largely depend on the service quality offered by the hotels. We have seen the problem of attracting new customers and retaining the existing customers for a long time with satisfactory services. In Ethiopia, many organizations lose their customers and they become out of the business competition because of these problems. Then, we are motivated to develop CLP because this kind of customer rewarding system is not widely used in Ethiopia. And it provides a win-win solution for business owners and customers. The system can be implemented in any sector however out focus on hotels.

1.4. Project Objectives and Goals

Loyalty programs are deliberate marketing efforts that influence customers’ buying behavior. In short, these programs aim to influence users to buy more from products listed under the loyalty program.

worku

1.4.1. General Objective

The general objective of this project is to build a customer loyalty system for hotels.

1.4.2. Specific Objective

The specific objectives of this project are:

- The deal about the current system of customer handling in hotels to identify the problems via different methodologies.
- Design, implement, test, deploy, and train users of the system.
- Provide web-based and mobile applications for hotels.
- Attract new customers by offering points or discounts for sign-ups.
- Minimize the time, effort, and money spent on hotels to advertise their work.
- Help hotels to have a consistent number of loyal customers.
- Assist hotels to have a general understanding of the customers' flow and have a summary and prediction of the customers’ needs.
- Better customer retention. The primary motive behind the customer loyalty program is to retain customers by rewarding them for their repeat purchase behavior in the hotel.

1.4.3. Goals

The goals of the proposed system are: -

- To bring the customers and hotels closer to each other so both of them can feel satisfied with what they do.
- To bring the technology in between the customers and the hotel so they can always be updated and go with the time.
- To make hotels more preferable by reducing the cost of advertisement to attract new customers and retaining their customers for a long time.
- To expand this system to other small businesses like supermarkets and gymnasiums so the customers can always have multiple choices of services and products. The system is easily modifiable for these kinds of purposes.

1.5. Scope and Limitations of the System

1.5.1. Scope

CLP is a web-based and mobile application which primarily used for hotels and customers to have a close relationship. To achieve the above goals and objectives the CLP must have the following functionalities and procedures.

Customer Acquisition (Get new customer)

This is the first process of new customer attraction to the hotel. We first start with product listing with offers for new customers that are not loyal.

a) Client web application and mobile application
   i. Clients can see menus or listings of menus of services and they can see the benefits of joining the loyalty program.
   ii. Clients can give feedback for the service they are used with satisfaction scores.
   iii. Receive a thank you short message service (SMS) with the name of the hotel when they signup and they can get points.

b) Administrator
   i. Allows to prepare offer form (name of the product or service to be offered, discount or free, number of customers to be served and date the offer to be finished)
   ii. Receive and see feedback list

Market Automation (Auto Customer engagement)

This is a method to win the customers back when they are not enrolling in the program but they were registered already.

a) Client web application and mobile application.

   i. Display offer list on the offer menu according to the offer type.

   ii. The customer will give feedback until the deadline is reached.

b) Administrator

   i. Prepare offer giving form but this offer type is for customers after the first-time visit. The offer will be free or the discount may be based on data mining or knowledge extraction.

   ii. By enrolling in the auto engage, it sends the offer stated above to the customer through their phone number, app, or automatically send the offer without engaging when the date is reached.

   iii. It finally measures the rate of success of the hotel gates by the number of receiving customers and the number of customers who successfully accepted the offer.

   iv. Receive if any feedback is given from customers that are not willing to accept the offer.

Loyalty System

This is a method used to give discount or offer for loyal customers

a) Client web application and mobile application

   i. Allows the customers to sign up to enroll in the loyalty program. Then, sends confirmation code through their phone number and fill the code to sign in and get a unique bar or quick response (QR) code.

   ii. See the complete status of points and rewards.

   iii. Allows to share the app or referral for the customer acquisition process and earn extra points.

   iv. See the free or discount service after they have been rewarded.

   v. Gives feedback for the services they get because their feedback is mandatory and also is the way of getting another point.

   vi. View their level.

   vii. View the summery report on their spending.

b) Administrator

   i. Allows to see the status of each customer like the points they get (from signup, purchase, feedback), how many points they have left to get the reward, see the round of reward they get for an advance reward.

ii. Generate membership rules specific, product rules, events, and referral rules.

iii. Segmentation and leveling of loyal customers, transfer customers from a lower to a higher level of loyalty.

iv. Assign the point limit and reward of customers. Here, the main thing the system uses the concept of data mining or knowledge extraction that helps to see patterns of the services the customer uses and automatically recommend reward of service or discount.

v. Allows seeing the status of getting a point instantly after a transaction on the front table or cashier with the help of a tablet or other digital display.

vi. Send a short message or notification using the app when you earn a point, the remaining point to get the award is getting smaller or when the customer is awarded.

vii. Predict mostly usable products and services.

viii. Feedback view and transactional report generation using bar, chart and gives evaluation.

1.5.2. Limitation

Due to the current technological status of our country, we will not be able to implement an online payment system. If there is any chance of having an online payment system in the country, it will not be that much of a problem to integrate the system with payment, The system doesn’t include a hotel management system, and also if there is a failure of the network, the customers who want to use services when they are far from the hotel can’t communicate with the hotel.

1.6. Methodology

We have used different methods and tools throughout the development life cycle of the project to verify the required requirements and validate the proposed needs of customers and system owners. We are going to use the iterative software process model since it is a very flexible development method. Some of the techniques and tools we have used in our project are listed below.

1.6.1. Data Collection

The requirement elicitation and requirement analysis process was conducted via Interview, Observation, and Questioner.

- **Interview**: We have interviewed 3 hotels with different levels. These hotels are Capital (in Addis Ababa), Ras hotel (in Adama), and Ananya hotel (in Adama). As we interviewed these hotels have a different business model to get how they are currently working on their customers, and also we interview some customers about how they are served and what benefit will get from the hotel they are using. In the interview, we have got a deep understanding of the current system.
- **Observation**: it helps us to get more information about the current system of how they work.
- **Questioner:** we prepared some sort of questions to ask about the system they have. Some of these questions include different open and closed forms of questions.

The interview and the questioner are indicated in appendix I.

1.6.2. System Design and Analysis Tools

To develop a usable, achievable, and efficient system, analysis and design are the common and main parts of the development life cycle of a project. In our system, we have included analysis and design tools and techniques with object-oriented analysis and design approaches.

The tools that we are going to use are:

- **Microsoft Office** – is an application used to prepare requirement specification document, we have used Microsoft word 2016.
- **Enterprise Architecture, Edrawmax**, – is a software to create models, viewpoints, views, visualizations for stakeholders. To prepare a simplified unified modeling language (UML) design of the proposed system.
- **Gantt** –is an online project management solution that utilizes a Gantt chart approach to help users become more efficient and productive in managing their projects. We have used this GanttPro to produce a Gantt chart for the schedule or timeline.
- **Azure**– is a prototyping, documentation, and specification software tool for web, mobile,0 and desktop applications and is used to develop the designed user interface or prototype.

1.6.3. System Development Tool

We have planned to use different development tools, programming languages, and styling frameworks in the system development process.

- For the **front end**, it's planned to use **Vuejs**. Vuejs is a progressive framework for building user interfaces. Unlike other monolithic frameworks, it is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only and is easy to pick up and integrate with other libraries or existing projects. On the other hand, it also perfectly capable of powering sophisticated single-page-applications when used in combination with modern tooling and supporting libraries. It also supports the ES6 features "write less do more", like importing and exporting modules to build a usable and maintainable system component [10]. And **HTML** is included in the front end for rendering the pages to the browser.
- For **styling,** it is planned to use **Vuetify** and cascade sheet style (**CSS**). Vuetify is a Vuejs library that is used to build effective, user friendly, and responsive web applications [11].
- On the **backend** side, we will use **NodeJS.** Nodejs is a JavaScript run-time built on chrome's v8java script engine. It uses an event-driven, non-blocking (asynchronous) I/O model that makes it lightweight and efficient. Node.js package ecosystem, npm (node package manager), is the largest ecosystem of open-source libraries in the world [6].
- In the **back end, expresses** is selected. it is a highly-extensible, open-source NodeJS framework that enables us to [7]:
    - Create dynamic end-to-end rest APIs.
    - Access data from major non-relational databases, like Mongo DB.
    - Incorporate model relationships and access controls for complex APIs.
    - Separate components for file storage, third-party login, and authentication.
- On the **database** side, we will use a non-relational database which is **MongoDB** [8].

  It enables us to:
    - Stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.
    - The document model maps to the objects in our application code, making data easy to work with.
    - Ad-hoc queries, indexing, and real-time aggregation provide powerful ways to access and analyze our data.

- Flutter: is an open-source mobile application development software development kit (SDK). It is used to develop applications for android and iPhone operating system (IOS), as well as being the primary method of creating applications for Google fuchsia [9].
- Editors: visual studio code, android studio.
- Environment: windows, android, and IOS operating systems are the platforms where our system is to be running.

1.7. Significance of the Project

Customers are the most important factor in every service industry. If the customer is satisfied with the service you provide to them, then definitely they will not only retain but also they will help you to increase your business by word of mouth publicity. Then, it becomes walking talking advertisements for hotel owners and service providers. Thus, it indirectly helps to lower the cost of attracting new customers. Satisfied customers tend to buy more and to generate positive word-of-mouth recommendations.

Then the CLP will increase the relationship between the customer and the hotel owner by minimizing the gap. And this program allows the hotel to have many customers and to be more profitable.

1.8. Organization of the Document

In this chapter, we provided an overview of the system. In the second chapter, we have discussed the requirements of the system stating the functional and non-function requirements. The third chapter mainly discusses the UML that shows the structural, behavioral, and interaction diagrams in detail. From these UML diagrams, we have mentioned use case diagrams, activity diagrams, sequence diagrams, state diagrams, and class diagrams. In the fourth chapter, the system design is described briefly. The main points we have touched on in this chapter are system decomposition, the architecture of the system, component diagram, deployment diagram, and database diagram which is Entity-Relationship diagram, and also, we have discussed the interface design. The next chapters are mainly concerned with the implementation, testing, and deployment of the system respectively.

Chapter 2

System Requirement Specification

2.1. Background Overview

CLP is a web-based application dedicated to satisfying both hotels and their customers by more tighten the interaction between them.

In Ethiopia, almost no system helps hotels helps to attract new customers and retain their customers for a long time by giving offers and special advantages.

2.2. Functional Requirements

Functional requirements are requirements that are used to build the main functionality of a system in which it interacts directly with both customers and system owners. Our project includes many functional requirements [14]. These requirements are collected, organized, and ordered through different requirement gathering methods and finally, we have used the elicitation process to select the best one, more appropriate, and these functionalities are used as a backbone for the system to apply it without any doubt. We used a temporary ID for each functional requirement for better identification with one another.

Registration

Super admin registration

- First of all, Super admin is registered to manage the whole system.

Branch admin registration

- The system allows branch admin to be registered by the super admin if the hotel has a branch

Customer registration

- The system allows customers to be registered online by the mobile application or the website.
- Basic information will be first name, last name, sex, birth date, age, phone number, username, password, and recovery answer.

Login

The system lets the super admin to login with his/her user name and password

- The system gives the super admin the highest level. So, he/she can control the others by registering or deleting, but not adding information about the service. offer and events.
- The system lets to view the report about the events, offers, and others
- The system doesn't allow to see and change the password and username of the branch admins or the lower authority

The system lets branch admins to login with their user name and password given by the system.

- The system gives branch admins the responsibility of adding information about the service, events, offers
- The system doesn't allow to see and change the password and username of the customers

The system lets customers login with their user name and password given by the system

- It allows accessing released information by the hotel

Add information

Add new branch

- The system allows adding a new branch by the super admin using the information of name, city, region, telephone.

Add services

- The system allows the branch Admin to add new services to the system, including its price and additional information.

Add events

- The system allows the branch Admin to add upcoming events; it may screen some customers to have special access to the event.

Add offers

- The system allows the branch Admin to add offers the hotel gives when some point is reached.

Add levels

- The system allows the branch Admin to add levels that the customers reach when spending some amount of points in the hotel.

Add earning point rules

- The system allows the branch Admin to add the rules that govern how the point is earned in the hotel.

Add reward campaign

- The system allows the branch Admin to set a campaign to get new customers.

Scanning

Customer scan barCode or bar code

- The system allows every loyal customer to scan their barcode QR code when they purchase.

Update Information

Update super-admin profile information

- The system can update the super Admin profile and information.

Update branch admin profile information

- The system can update the branch Admin profile and information.

Update customer information

- The system can update the customer profile and information.

Update earning point rules

- The system lets the branch Admins update existing rules that govern how the point is earned in the hotel.

Update services

- The system lets branch Admins update existing services to the system, including their price and additional information.

Update offers

- The system lets branch Admins update offers the hotel gives when some point is reached.

Update levels

- The system lets branch Admins update levels that the customers reach when spending some amount of point in the hotel.

Delete Information

Delete branch admin

- The system allows the super Admin to delete the branch admin; if necessary.

Delete customers

- The system allows the branch Admins to delete customers; if the customer is no longer involved in the system.

Delete services

- The system allows the branch Admins to services; if services are no longer given in the hotel.

Delete offers

- The system allows the branch Admins to delete services.

Delete events

- The system allows the branch Admins to delete events that have passed or no longer available.

Delete earning point rules

- The system allows the branch Admins to delete point earning rules.

Delete levels

- The system allows the branch Admins to delete levels governing types of customers.

View Information

View spent points

- The system lets the customers view their spent points and unused points.

View customer information

- The system lets customers view their profile and information.

View level list

- The system lets the customers view their levels and other levels they can reach when some point is reached.

Earning point rule

- The system lets the customers' view point earning rules and do things to earn the points.

Earning point

- The system lets the customers view when they earn points.

Reward campaign list

- The system lets the customers view the reward campaign list and involve in the campaign.

Feedback or comments

- The system allows branch Admins to see the comments and feedback given to the comments.

View notification from the system

- The system gives notification about some content that is to be managed.

View services

- The system lets the customers view services given by the hotel before getting into the hotel.

View offers

- The system lets the customer's view offers given by the hotel when some point is reached.

View events

- The system lets the customers view upcoming events that are held in the hotel with their specific times and branch.

View admins profile information

- The system lets super Admin to view the profiles and information of the branch Admins.

View customer profile information

- The system lets the branch Admins view the profiles and information of customers.

Viewpoints (active points, used points, expired points)

- The system allows customers to see active points, used points or expired points he/she has.

View rewards

- The system allows customers to view the rewards they get from different offers, events….

View transactions

- The system allows the branch admins to view the transactions that took place in the hotel.

Data manipulation

Searching information

- The system can search for specific activities.

Sorting information

- The system can sort the list of information such as transactions, offer lists….

Generate Reports

Generate a report about the active loyal customer

- The system generates reports about active loyal customers and gives a summary.

Generate a report about customers referred by another customer

- The system generates reports about the customers that involve in the system from referrals of other loyal customers.

Generate a report about customer feedback with their time

- The system generates reports about the customers' feedback about the hotel with their respective times of feedback.

Generate a report about a customer transaction

- The system generates reports about the customers' transactions in a day or week.

Generate a report about rewards, point spent and also the date

- The system generates reports about the total points spent in the system with their respective time.

Generate a report of customer distribution like Male, female

- The system generates reports about customers' gender distribution to know in detail about the type of customers.

Share

Share services

- The system lets customers share the services provided by the hotel and earn points.

Share offers

- The system lets customers share the offers provided by the hotel and earn points.

Share events

- The system lets customers share the events provided by the hotel and earn points.

Share referral code

- The system lets customers share their referral codes as links to let people follow and see the system.

Alert Notification

- The system gives alert notification when customer sign up for the first time like welcome, adding a point
- The system gives alert notification after customer make transaction like the point earned and the item used
- The system gives alert notification when an event, offers, and rewards are available based on recommendation
- The system gives an alert notification when the user leaves or wants to delete an account
- The system gives an alert notification when a referred customer is signed up

Logout

User logout

- The system allows any user to log out when necessary.

Forgotten Account

Forgotten account

- If any user will lose his or her password, then, the system allows resetting their passwords.

2.3. Non-Functional Requirements

Nonfunctional requirements are the properties that your product must-have. Think of these properties as the characteristics or qualities that make the product attractive, usable, fast, or reliable, and they generally support all users in that they describe the business standards and the business environment, as well as the overall user's experience (user attributes). They do not alter the product's functionality. That is, the functional requirements remain the same no matter what properties you attach to them. Our project includes some of the quality attributes which make the system functional indirectly.

- **Usability** – It allows customers and system owners to use this system easily and efficiently since it supports user-friendly interfaces and customized forms with different running environments. CLP also supports real-time data receiving and providing to the stakeholders without any redundancy of information. It will be accessible on any smartphone, computer, or tablet, and others.

- **Reliability** – The system will have crash handling mechanisms to recover from any failure during the transaction process in a few seconds. This mechanism is supported by the database itself and recovers from failure by nodejs libraries, morgan, and pool execution.
- **Performance** – Our system has a short response time for given access, accuracy, efficiency, and high throughput (high rate of work), since the back end will be built with node js framework that is a non-blocking and asynchronous, no one is waiting for others process to complete when accessing in the same time that is many users access the system in an instant.
- **Availability** - the system is accessible anytime, anywhere via computer, mobile devices, and tablets with an internet connection.
- **Maintainability** – It is easy to maintain this proposed system during correcting defects or their cause, repair or replace faulty or worn-out components without having to replace still working parts, and prevent unexpected working conditions.
- **Security**– We use a secure and non-relational database to protect customer information and customer points from any harm forced by unauthorized access. Any user of this system will have a username and password stored in the database. Both the username and password are encrypted data, no one can use them without authentication.
- **Sustainability**– The capacity of the software to endure**.** In other words, it means that the software will continue to be available in the future, on new platforms, meeting new needs**.**

2.4. Feasibility Study

The feasibility study is the procedure to predict the outcome of an investigation examination or assessment of a planned scheme along with possible gain. We have tried to check whether our new proposed project is feasible or not feasible in different conditions and situations. Some of the evaluated visibility conditions are listed below.

- **Product feasibility** – The final product of our new project is a web-based and mobile application used to engage or retain customers, give them some offers, communicate with customers directly to increase benefits and satisfaction, and store customer information's to determine who is loyal.

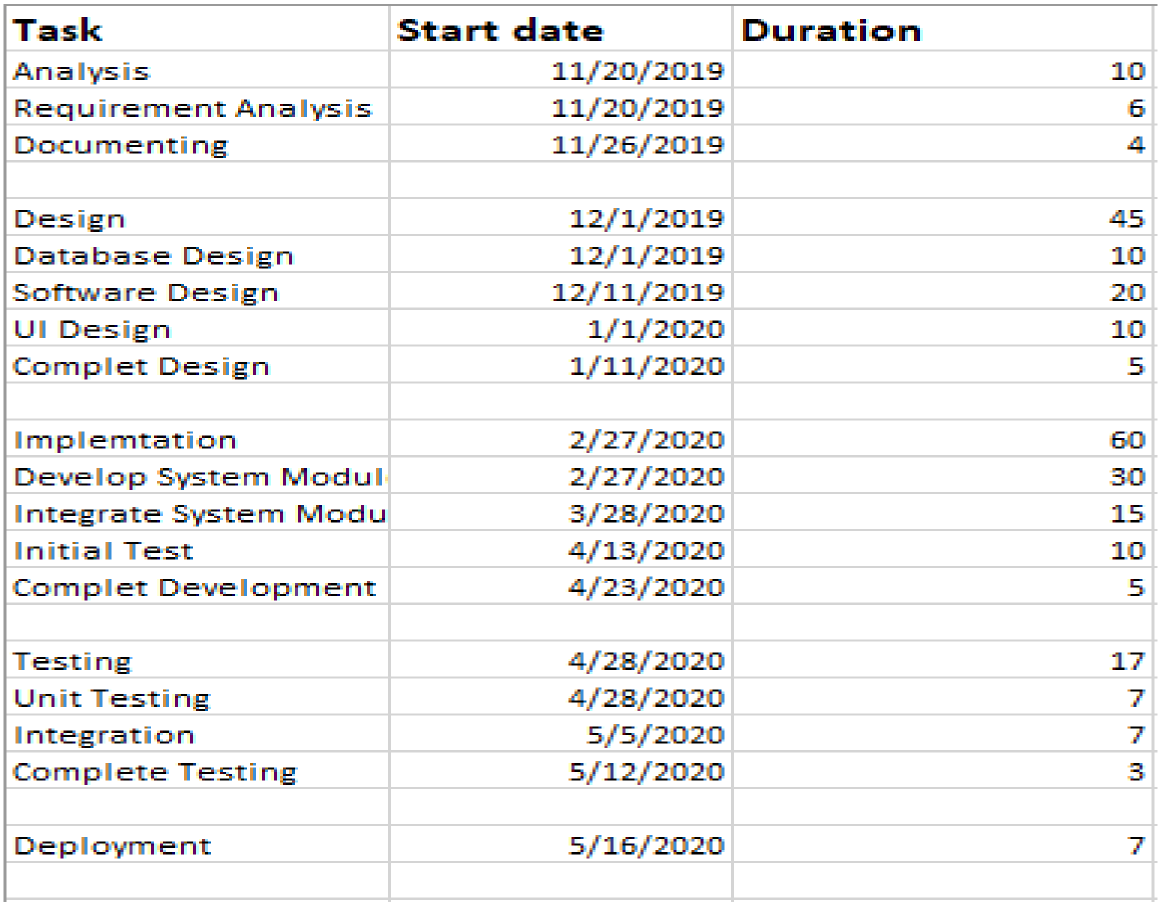
- **Economic Feasibility** – This project is economically feasible since the cost of customer attraction and retention is high without loyal programs, but the development cost of this project is lesser than the cost of manually attracting customers. Generally, this new project is cost-effective and efficient.
- **Technical Feasibility** -- Technical feasibility is mainly associated with the technologically evaluates the project. The main technologies and tools included with the customer loyalty program are:

  - **Microsoft office** – used to write and edit the project's documentation.
  - **AzureRP** – used to draw all the design models needed for the system.
  - **Vuejs** – used to implement the front-end for our web application.
  - **Nodejs** – used to implement the back-end for our web application.
  - **Flutter** – used to develop a mobile application
  - **MongoDB** – the non-relational database for our system.
  - **Expressjs** - used to create dynamic end-to-end rest APIs.

  Each of the technologies is freely available and the technical skills required are manageable. Time limitations of product development and the ease of implementing using these technologies are synchronized.

- **Legal Feasibility** – CLP is legally feasible because this system does not conflict with national or international legal requirements. Any personal information is stored securely. This system is focused on the loyal programs, not on the management issues of the Hotel.
- **Operational Feasibility** – We have studied the behavioral feasibility of our project. It will fulfill all the functional and business requirements to satisfy user needs. These actions forecast all possible schemes to recognize and resolve troubles. This system also supports user-friendly interfaces with minimum input forms and any help to avoid confusion.
- **Schedule Feasibility** – We have planned to complete this system in 7 months with different phases and with iterative requirement analysis. If it is necessary through the development cycle. Currently, we are going with the right schedule to finish all phases with the specified time interval. So, we are sure our proposed project is schedule feasible. The schedule of our project is shown in table 1.1 and figure 1.1.

Table 2 1: Time Schedule



Task	Start date	Duration
Analysis	11/20/2019	10
Requirement Analysis	11/20/2019	6
Documenting	11/26/2019	4
Design	12/1/2019	45
Database Design	12/1/2019	10
Software Design	12/11/2019	20
UI Design	1/1/2020	10
Complet Design	1/11/2020	5
Implemtation	2/27/2020	60
Develop System Modul	2/27/2020	30
Integrate System Modu	3/28/2020	15
Initial Test	4/13/2020	10
Complet Development	4/23/2020	5
Testing	4/28/2020	17
Unit Testing	4/28/2020	7
Integration	5/5/2020	7
Complete Testing	5/12/2020	3
Deployment	5/16/2020	7

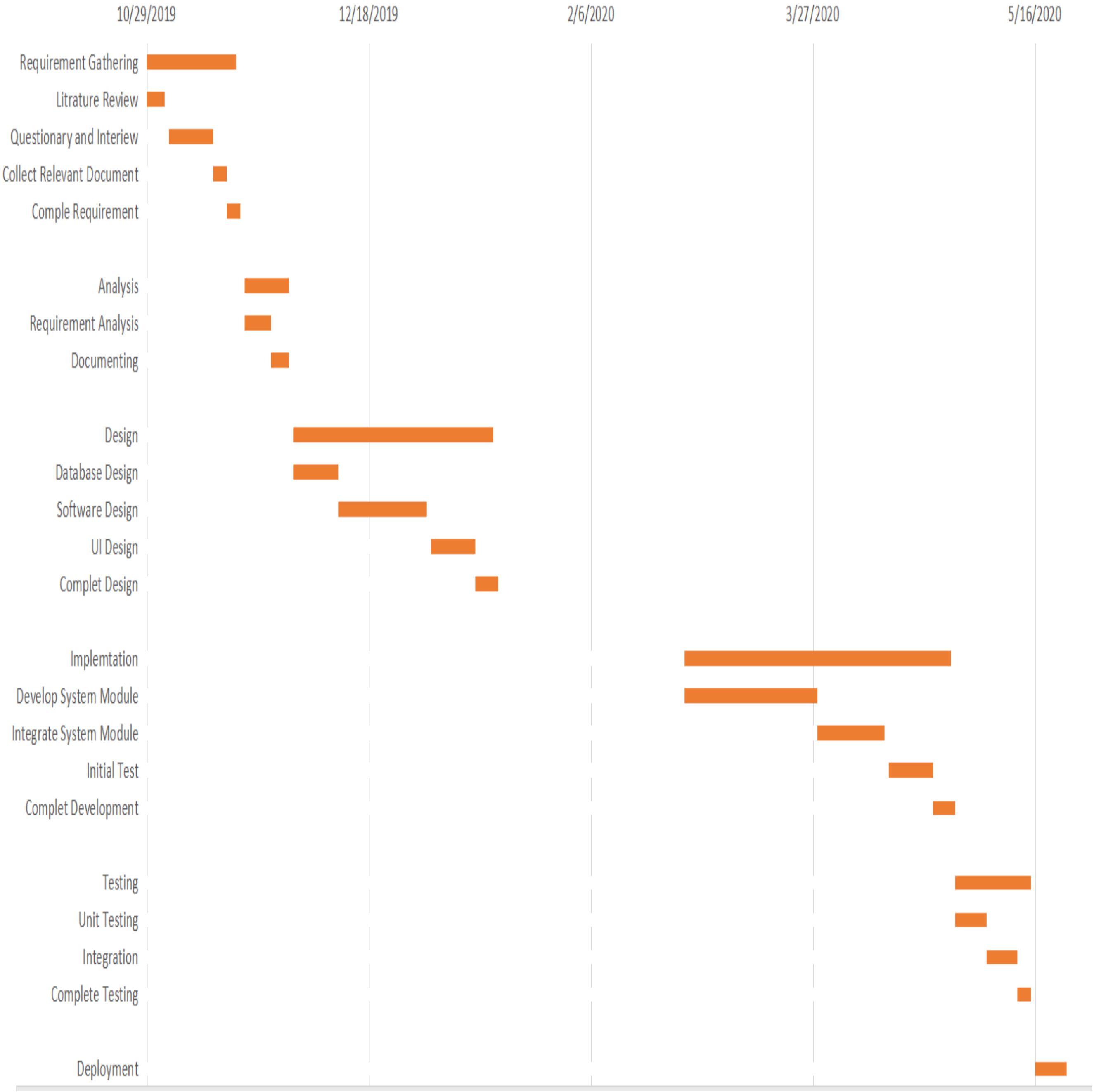


Figure 2 1: Time Schedule in Gantt Format

Chapter 3

System Analysis and Modeling

3.1. Overview

System analysis and modeling is the process of studying an activity to define its goals, purposes, discover operations and procedures for accomplishing them most efficiently. It is also a process of collecting, interpreting facts, identifying the problems, and decomposition of a system into its components [5].

UML is a standardized modeling language enabling developers to specify, visualize, construct, and document artifacts of a software system. UML is an important aspect involved in object-oriented software development. It uses graphic notation to create visual models of software systems [5]. We use UML to develop this system to explain the functionality of the system due to the flexibility, portability, and its worldwide standard (acceptance to modeling processes of a system).

3.2. Scenario-Based Modeling

While technically there is no right way to go through the stages of requirements modeling, it typically begins with **scenario-based modeling**, and that is because it identifies the possible use cases for the system and produces the use case diagram, to which all the other stages of requirements modeling [5].

3.2.1. Use Case Identification

A use case is a list of actions or event steps typically defining the interactions between an actor and a system to achieve a specific goal [5]. Based on the collected and extracted requirements we have included the following functions or use cases.

- Signup
- Login
- Manage branches
- Manage branch admin
- Manage service
- Manage events

- Manage campaign
- Update information
- View information
- Search information
- Sort information
- Share posts
- Refer friends
- Redeem rewards
- Give feedback
- Report generation
- Feedback response
- Update personal information
- Delete account
- Logout
- Get help
- Update mobile application
- Synch data from another system

3.2.2. Actor Identification

An actor is a user or other system which interacts with the system to achieve a goal. This interaction is done through the identified use case. In our system, we have identified the following actors.

- **Super administrator (admin) –** Super admin is any person who works in the hotel and appointed to control the whole activities of the system, activities like adding customers, registering branch admins, confirm customers when they leave the system, announcing events, view feedback or comment with the response, etc.
- **Branch admin** – the responsibility of branch admin is to register customers, post information the hotel provides, give the response to customer feedback, and others.
- **Customer –**is any person who has an account to get and additional services from the hotel. They consume services like offers, discounts, events, rewards, etc.
- **Visitor –** any person who has no account in the system and visits the hotel services.
- **System –** the system which is used to sync data from the other system. In this case, it

is used to fetch customer data from the hotel management software when customers purchase.

3.2.3. Use Case Diagrams

A use case diagram is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved [5]. We have classified the use case diagram based on actor type. Figure-2.1 up to figure-2.5 shows the use case diagram for the respected actors.

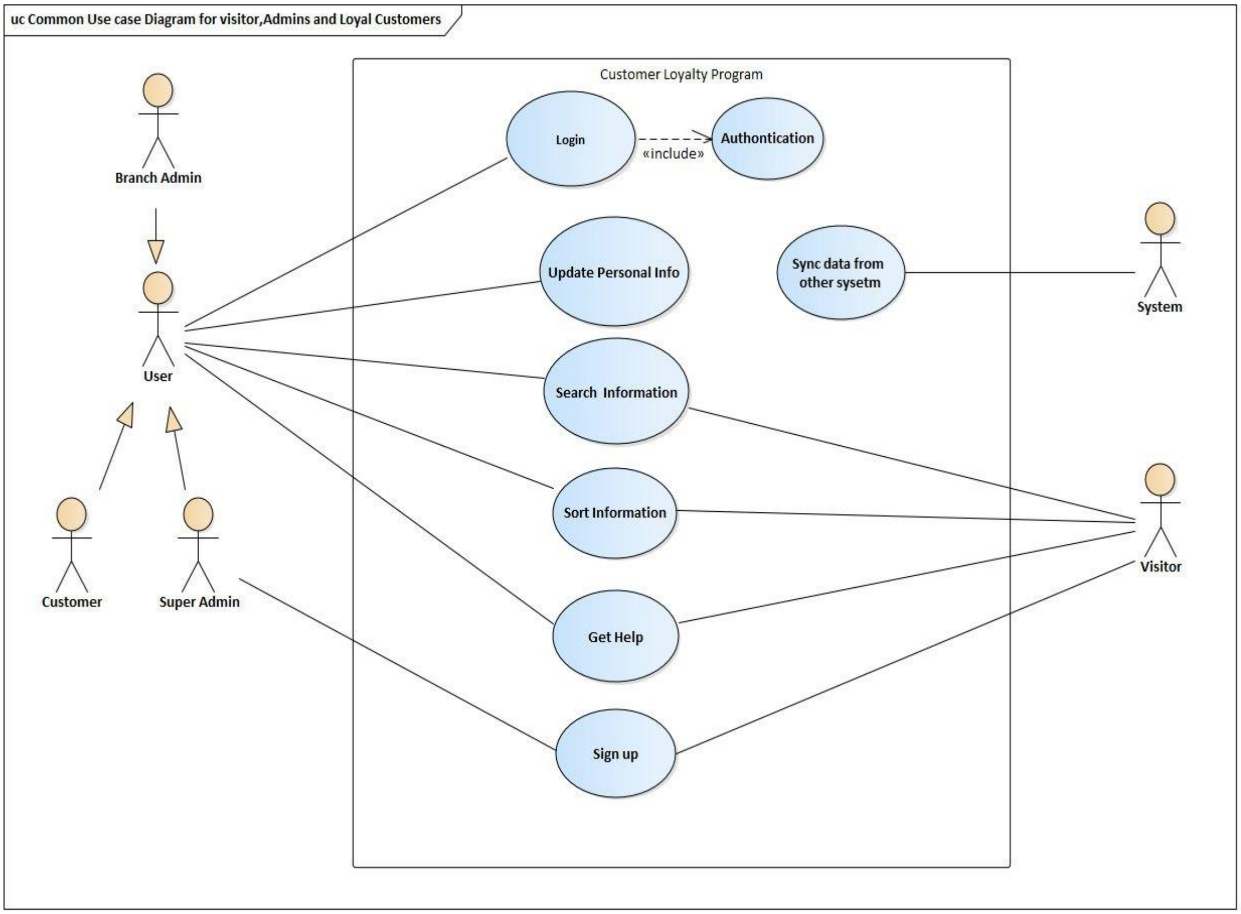


Figure 3 1: Common use cases for all actors

For this diagram login must be referenced from the use case diagram of common use cases for all actors as included here.

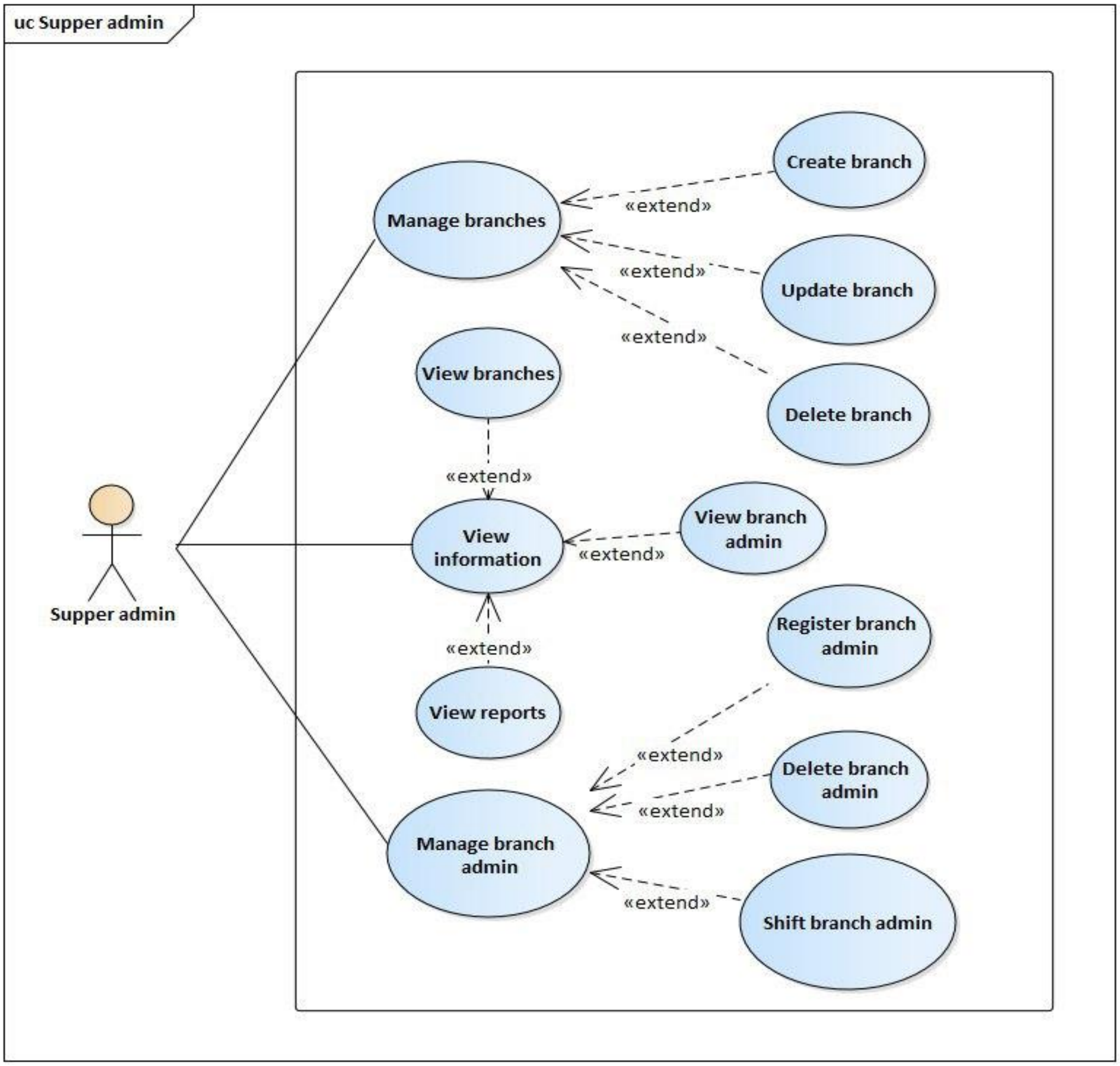


Figure 3 2: Use case diagram for super admin

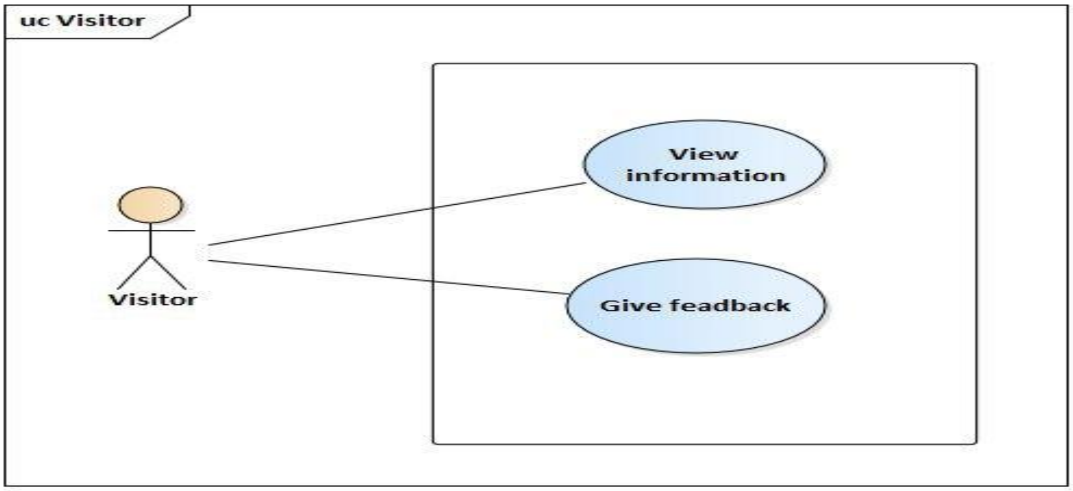


Figure 3 3: Use case diagram for visitor

For this diagram login must be referenced from the use case diagram of common use cases for all actors as included here.

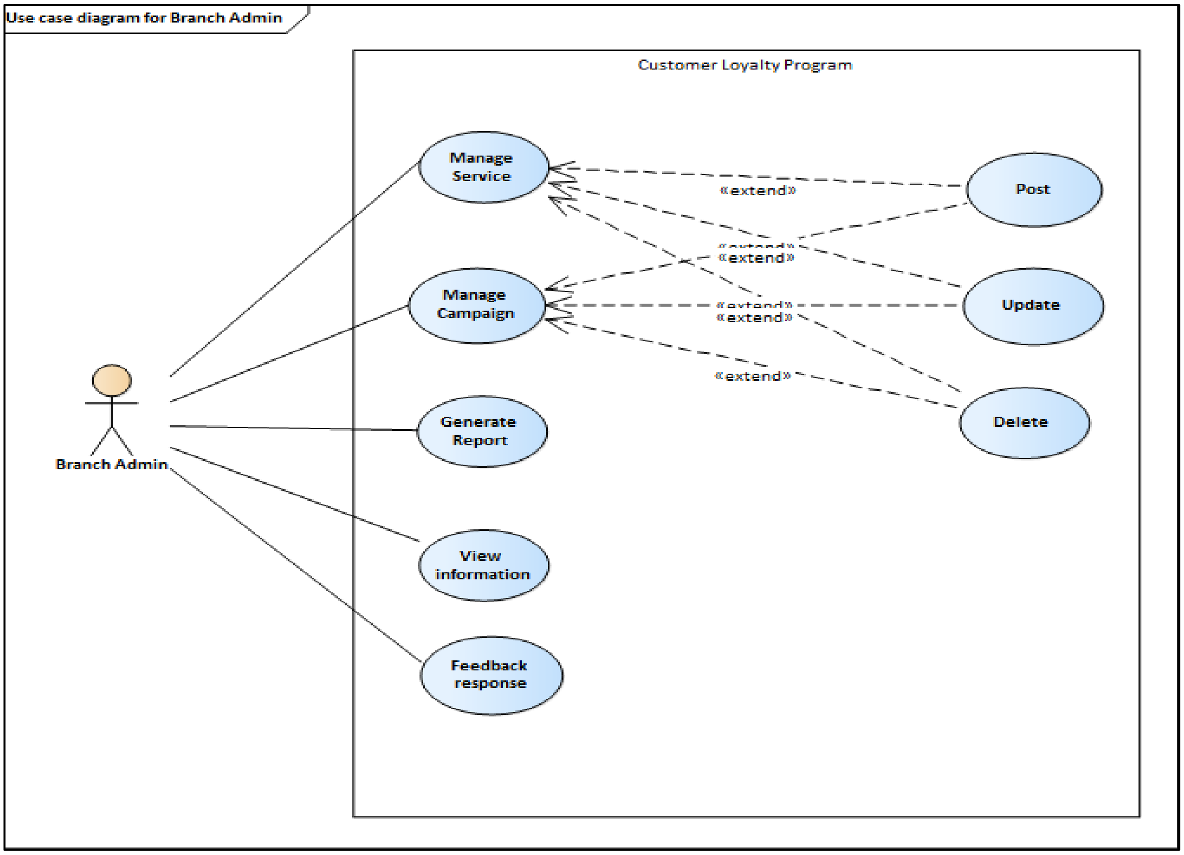


Figure 3 4: Use case diagram for branch admins

For this diagram login must be referenced from the use case diagram of common use cases for all actors as included here.

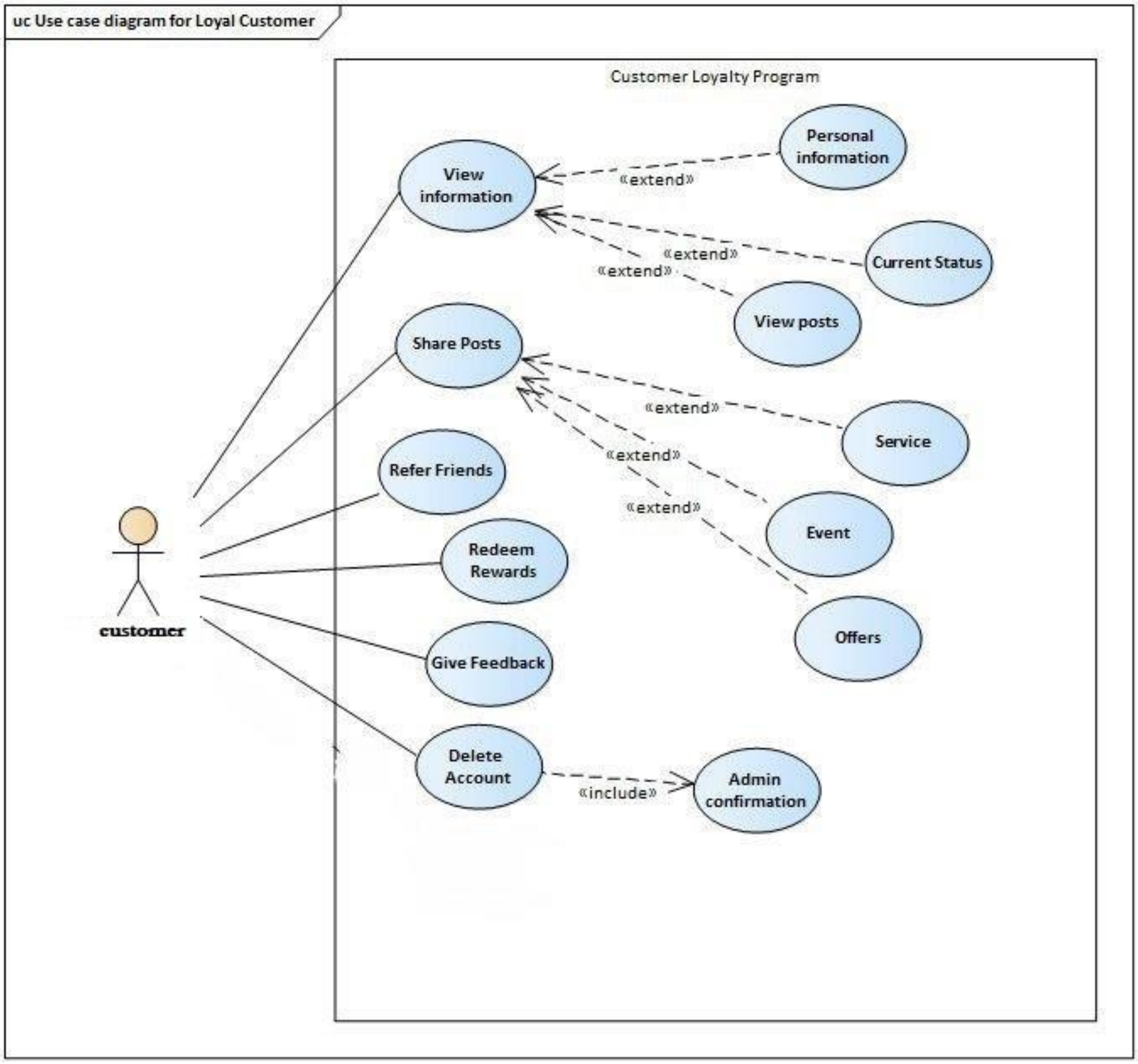


Figure 3 5: Use case diagram for customer

3.2.4. Use Case Description

The use case description is a narrative document that describes, in general terms, the required functionality of the use case. Typically, it describes the use case goal and gives a general description of what usually happens, the normal course of events, adding a brief description of any minor variations [5] as described in table-3.1 up to table-3.16.

Table 3 1: Use case description for Signup

Use case name	Signup
Use case ID	UC-1
Used Use case	---
Brief description	This use case allows the customer to create an account.
Participant Actor	Visitor
Precondition	---
Postcondition	A new user account to be used by a customer is created.
Basic course of action	Actions: 1. User click the signup button on a website or mobile app 2. The system displays the registration form 3. The user fills the provided form and clicks the submit button 4. The system checks the input[A1],[A2] 5. The system displays a confirmation text field and sends confirmation code via SMS 6. Te user insert confirmation code[A3] 7. The system checks the confirmation code 8. The system activates the user account after confirmation
The alternate flow of Action	A1. If the user did not fill in all required information or try to use an existing account, the system will display “fill all required fields” and “the account already exists” respectively and go back to step 3.
	A2. If the user inserts the wrong input, the system displays “wrong input” and go back to step 3.
	A3. If the user the confirmation code is not correct, the system displays “incorrect confirmation code” and go back to step 6.

Table 3 2: Use case description for Login

Use case name	Login
Use case ID	UC-2
Used Use case	Include: Authenticate
Brief description	This use case allows a user to login and accesses the system.
Participant Actor	Customers, Super admin, Branch admin
Precondition	The actor must have a valid user name and password
Postcondition	The user logged into the system to execute the privileged tasks
Basic course of action	Actions: 1. User open website or mobile app and click login button 2. The system displays a login form 3. The user inputs user name and password 4. The system checks the input validity[A1] 5. The system allows users to perform their privileged tasks
The alternate flow of Action	A1. When the user enters the wrong user name and password the system displays “incorrect username and password” and goes back to step 3.

Table 3 3: Use case description for Create Branch

Use-case Name	Create Branches
Use case ID	UC-3
Brief description	Used to register a new branch
Used use-case	---
Participant actor	Super admin
Pre-condition	The super admin should have to log in to his/her account
Post-condition	A new branch will be created
Course of action	Actions: 1. Super admin click branches tab 2. The system displays a list of branches in the table with different buttons above the list. 3. The user clicks the add new branch button 4. The system renders the add branch from 5. Fill the required fields and click on the register button 6. The system display “successfully registered” with branch name
The alternate flow of Action	Al. If the user clicks the register button before filling all required fields, the system will display “fill all required information” and go back to step 3.
	A2. If a user inserts the wrong input, the system displays “Incorrect input” and go back to step 3.

Table 3 4: Use case description for Delete Branch

Use-case Name	Delete branch
Use case ID	UC-4
Brief description	Used to delete the existing branch
Used use-case	---
Participant actor	Super admin
Pre-condition	The super admin should have to log in to his/her account
Post-condition	The branch will be deleted
Course of action	Actions: 1. User click branches tab 2. The system displays a list of branches in a table with different buttons above the list. 3. The user clicks on a specific branch from the list to delete[A1] 4. The user clicks on the delete button 5. The system asks for confirmation from the user before delete. 6. The user clicks “yes delete” to confirm 7. The system display “successfully deleted” with branch name
The alternate flow of Action	A1. If the user clicks the delete button before selecting a specific branch, the system display “select branch to delete” and goes to step 3.

Table 3 5: Use case description for Update Branch

Use-case Name	Update branch
Use case ID	UC-5
Brief description	Used to update the branch information
Used use-case	---
Participant actors	Super admin
Pre-condition	The super admin should have to log in to his/her account
Post-condition	The branch information will be updated
Course of action	Actions: 1. User click branches tab 2. The system displays a list of branches in a table with different buttons above the list. 3. The user clicks on a specific branch from the list to update[A1] 4. The user clicks on the update button 5. The system displays branch information in editable form 6. The user edits what he/she wants and clicks the update button[A2] 7. The system display “successfully updated” with branch name
The alternate flow of Action	A1. If the user clicks the update button before selecting a specific branch, the system display “select branch to delete” and goes to step 3.
	A2. If the user inserts the wrong input and clicks the update button, the system displays “please fill correct information” and goes to step 6.

Table 3 6: Use case description for Assign branch admins

Use-case Name	Assign branch admin
Use case ID	UC-6
Brief description	Used to assign new admin for a branch
Used use-case	---
Participant actors	Super admin
Pre-condition	The super admin should have to log in to his/her account
Post-condition	The new admin is assigned to a specific branch
Course of action	Actions: 1. The user clicks the branch admin tab 2. The system displays the branch admin window with a list of existing admins 3. Click add branch admin 4. The system renders an admin registration form 5. The user fill the required form and clicks an allocate button[A1],[A2] 6. The system displays branch admin successfully assigned with admin name and branch ID.
The alternate flow of Action	A1. If a user clicks an allocate button before filling in all required data, the system displays 'please fill all required information' and goes back to 5.
	A2. If the user fills the wrong input or wrong format, the system displays 'incorrect input' and goes back to step 5.

Table 3 7: Use case description for Delete branch admins

Use-case Name	Delete branch admin
Use case ID	UC-7
Brief description	Used to delete a specific branch admin
Used use-case	---
Participant actors	Super admin
Pre-condition	The super admin should have to log in to his/her account
Post-condition	Existing branch admin will be deleted
Course of action	Actions: 1. The user clicks the branch admin tab 2. The system displays the branch admin window with a list of existing admins 3. Select branch admin and click delete branch admin[A1] 4. The system asks to confirm for delete 5. The user selects “yes delete” 6. The system display branch admin successfully deleted.
The alternate flow of Action	A1. If the user clicks the delete button before selecting admin, the system notifies “please select before” and go back to step 3

Table 3 8: Use case description for Shift branch admin.

Use-case Name	Shift branch admin
Use case ID	UC-8
Brief description	Used to shift admin from one branch to another
Used use-case	---
Participant actors	Super admin
Pre-condition	The super admin should have to log in to his/her account
Post-condition	Specific branch admin shifted to another branch
Course of action	Actions: 1. The user clicks the branch admin tab 2. The system displays the branch admin window with a list of existing admins 3. Select branch admin and click shift branch admin[A1] 4. The system renders shift form 5. The user fills in the required form and clicks the shift button.[A1],[A2] 6. The system validates the input and displays the branch admin successfully shifted.
The alternate flow of Action	A1. If the user clicks the delete button before filling all fields the system notifies “please fill all required information” and goes back to step 5.
	A2. If the user inserts wrong input the system display “incorrect input” and goes back to step 5.

Table 3 9: Use case description for a post (service, offer, and event)

Use case name	Create a post
Use case ID	UC-9
Used Use case	---
Participant Actor	Branch Admin
Brief description	This use case used to post information like service, events, and offers
Precondition	The branch admin must log in to the system
Postcondition	The new post is added to the list of previous posts
Basic course of action	Actions: 1. Branch admin click post button 2. The system displays a post input form[A1][A2] 3. Branch admin fill all the required inputs with the respective type of post and clicks the submit button 4. System checks for the validity of the input
The alternate flow of Action	A1. If the branch admin click the submit button without filling all input, then the system will display a ‘please fill all’ error message and goes to step 2 A2. If branch admin inserts invalid input the system displays an “invalid input” error message and go to step 2

Table 3 10: Use case description for update (service, offer, and event)

Use case name	Update post
Use case ID	UC-10
Used Use case	---
Participant Actor	Branch Admin
Brief description	This use case used to update information like service, events, and offers
Precondition	The branch admin must log in to the system
Postcondition	The information is updated for the selected service, event, or offer.
Basic course of action	Actions: 1. Branch admin select a row[A1] clicks the update button 2. The system displays an update form with selected data[A2][A3] 3. Branch admin changes the data displayed in the form as he/she wants and clicks the save button 4. System checks for the validity of the changes
The alternate flow of Action	A1. If branch admin clicks the update button before selecting a row, the system displays a “please select a row” error message
	A2. If the branch admin clicks the save button with an empty field, then the system will display a ‘please fill all’ error message and go to step 3
	A3. If branch admin inserts invalid input, the system displays an “invalid input” error message and go to step 3

Table 3 11: Use case description for delete (service, offer, and event)

Use case name	Delete post
Use case ID	UC-11
Used Use case	---
Participant Actor	Branch Admin
Brief description	This use case used to delete information like service, events, and offers
Pre-condition	The branch admin must log in to the system
Post-condition	The selected campaign is removed from the campaign list.
Basic course of action	Actions: 1. Branch admin select a row[A1] 2. Branch admin click delete button 3. The system displays the confirmation dialog (are you sure to delete with yes/no buttons) 4. Branch admin clicks yes/no button 5. The system removes the campaign if the admin clicks yes or no change if the admin clicks no
The alternate flow of Action	A1. If the branch admin clicks the delete button without selecting a row, the system displays a “please select a row” error message

Table 3 12: Use case description for view information

Use case name	View information
Use case ID	UC-12
Used Use case	--
Participant Actor	Customers, Super admin, branch admin, visitor
Brief description	Used to view the information in the system using a different user with their restriction to use the system
Pre-condition	Users have to log in except for the visitor
Post-condition	---
Basic course of action	Visitor 1. Open website or mobile application 2. Click on the service, even,t or offer tab 3. The system displays the service, event, or offer as per user action. Customer 1. Personal information 1.1. Click on the profile tab 1.2. The system displays detail personal information. 2. View point 2.1. Click my point tab 2.2. The system displays active, used, and expired points. 3. View posted information 3.1. Click on the service, event, or offer tab 3.2. The system display services, events, or offers as per customer action. 4. View Reward 4.1.Click on the reward tab 4.2.The system displays a reward list, points need to redeem a reward, and available rewards to redeem.

	5. View transaction 5.1.Click on the transaction tab 5.2.The system displays the detail of the transaction list Branch admin 1. View feedback 1.1.Click on the Feedback tab 1.2.The system displays the feed list 1.3.Double click on specific feedback 1.4.The system display single feedback in detail 2. view transaction 2.1.Open the dashboard and click on a transaction 2.2.The system displays a list of transactions by table form 3. view customer profile except for user name and password 3.1.The user clicks on a list of customer menu 3.2.The system displays a list of customers in table form 3.3.Double click on the specific raw 3.4.The system will display detailed information for specific customer 4. view report 4.1.The user clicks on the report menu 4.2.The system displays the report by default date interval of time and default report type. 4.3.The user selects a specific date interval or type to view the report 4.4.The system displays the report based on the user selection Super Admin 1. view their branches 1.1.User clicks on branches tab 1.2.The system detail about the branch 2. view branch admin profile 2.1.User click on the branch admin tab

	2.2.The system list of admins in table form 2.3.The user double click on a specific admin raw 2.4.The system displays a detailed profile of branch admins 3. view report 3.1.The user clicks on the report menu 3.2.The system displays report for all branch information's by default date interval and type 3.3.The user select date interval or type to view the report 3.4.The system displays the report based on the customer selection.
The alternate flow of Action	---

Table 3 13: Use case description for search Information

Use case name	Search Information
Use case ID	UC-13
Used Use case	---
Participant Actor	Customers, visitor, super admin, branch admin
Brief description	Users can search for information from the system by using different key entries like date, name, etc…
Pre-condition	The user must log in to the system
Post-condition	The system will display the searched information
Basic course of action	Actions visitor 1. The user opens the mobile app or writes the universal resource locator (URL) on the browser. 2. The system opens the loyalty program

	3. select item type to search like service, event, and offers, by date or type 4. The system displays the searched data if the output is available Customer 1. The user search service, offers, events, or transaction by date interval, type, or by status whether passed or upcoming[A1] 2. The system displays the services, events, offers, or transactions based on user search if the output is available. Branch admin 1. The user search service, offer, events, transaction, feedback, report, customer by ID, date interval, type, level or by status whether passed or upcoming campaign[A1] 2. The system displays the searched data as per the input if the output is available. Super Admin 1. The user search transaction in each branch, report, customer, or branch admin by ID, date interval, type, or level[A1] 2. The system displays the searched data as per the input if the output is available
The alternate flow of Action	A1. If the user inserts an invalid key to search the system will respond empty and go back to step 1.

Table 3 14: Use case description for Sort Information

Use case name	Sort Information
Use case ID	UC-14
Used Use case	--
Participant Actor	Customers, visitor, super admin, branch admin

Brief description	Users can sort information from the system by using different key choices like date, name, price, etc…
Pre-condition	There should be a list of data to be sort
Post-condition	Sorted data will be presented to the user
Basic course of action	visitor: 1. Open website or mobile application 2. The user click sort the data by date or alphabet 3. The system displays the sorted data Customer 1. The user clicks the sort button after selecting sorting mechanisms like by date, alphabet, or price to sort a campaign or transaction. 2. The system displays the sorted output Branch admin 1. The user sort service, offer, events, transaction, feedback, report, customer by ID, date interval, type, level or by status whether passed or upcoming campaign 2. The system displays the sorted output Super Admin 1. The user sort transaction in each branch, report, customer or branch admin by ID, date interval, type or level 2. The system displays the sorted output
The alternate flow of Action	A1. If users try to sort before selecting a sorting mechanism the system displays the default output.

Table 3 15: Use case description for Share posts

Use case name	Share posts
Use case ID	UC-15
Used Use case	Extends: Service extends: Event, extends: Offer
Participant Actor	Customers

Brief description	For sharing to friends, families, and others to view the post on the hotel’s page or social media or specific contact
Pre-condition	P1. The user must log in to the system. P2. User must use active social media account or use SMS to share information
Post-condition	The system counts the number of shares and adds points for respected customer
Basic course of action	Actions: 1. The user browses the content they want to share and click the share icon 2. The system displays the list of social media or contacts of customer[A2] 3. The user selects the social media page or contact to share[A1] 4. Then share the content by clicking the share icon again 5. The system posts the shared content for specified social media or contact.
The alternate flow of Action	Al. If the user clicks the share icon before selecting the social media page or contacts the system will display “select social media platform or contact to share” and go back to step 3.
	A2. If the user tries to share for unknown social media page or unknown contact the system displays “unknown social media page or unknown contact” and goes back to step 2.

Table 3 16: Use case description for referring friends

Use case name	Refer friends
Use case ID	UC-16
Used Use case	--

Participant Actor	Customers
Brief description	For referring friends, families, and others to sign up for a loyalty program
Pre-condition	User must log in to the system
Post-condition	The referral code will send to the selected social media account or contact number as SMS.
Basic course of action	Actions: 1. The user clicks on the invite icon to refer 2. The system displays the list of social media and contacts a list. 3. The user select social media page or contact he/she wants to refer through and click the refer button[A1],[A2] 4. The system sends the referral link with the specified referral code.
The alternate flow of Action	A1. If the user tries to refer before selecting a contact or social media account “the system display select first” and goes back to step 3.
	A2. If the user sends a referral for an unknown account or contacts the system displays “unknown contact or account” goes back to step 3.

Table 3 17: Use case description for redeeming rewards

Use case name	Redeem Rewards
Use case ID	UC-17
Used Use case	---
Participant Actor	Customers
Brief description	Allow the customers to use their reward after getting sufficient points or some discount given by the system.
Pre-condition	customer must have an account and he/she has to collect sufficient point to redeem a reward
Post-condition	Customer will get free or discount service as a reward
Basic course of action	Actions: 1. Click my reward tab 2. The system will display a list of rewards with a list of points needed[A1] 3. Click the redeem button in front of the specific reward 4. The system checks for the point sufficient to redeem the selected reward or not 5. The system will generate a bill for free service or discount on a specific item.
The alternate flow of Action	A1. If the user tries to redeem with an insufficient point the system displays “you don’t have enough point to redeem the reward” and go back to step 2.

Table 3 18: Use case description to update personal information

Use case name	Update personal information
Use case ID	UC-18
Used Use case	---
Participant Actor	Customers, Super admin, branch admin
Brief description	Used to change the personal profile of the user like user name, phone number, and password or profile picture.
Pre-condition	Users must have to log in to the system
Post-condition	User profile updated
Basic course of action	Actions: 1. The user selects the profile menu 2. The system will display profile information with change profile information 3. The system displays the existing profile information 4. The user selects edit profile from the hamburger menu and edits the information that he/she wants to update then click update button[A1],[A2] 5. The system validates filled information 6. The system displays profile updated successfully
The alternate flow of Action	A1. If the user clicks update before filling all required fields the system displays “please all required fields” and goes back to step 4.
	A2. If a user inserts the wrong input the system will display “incorrect input” and goes back to step 4.

Table 3 19: Use case description for feedback response

Use-Case Name	Feedback Response
Use case ID	UC-19
Used use-case	---
Brief description	Giving a response to incoming feedback
Participant actors	branch admin
Pre-condition	User must log in to the system
Post-condition	Notification will be sent to the respective customer
Course of action	Actions: 1. The user click the feedback tab 2. The system displays the feedback list by order of time. 3. The user selects the feedback from the list and clicks on the response button 4. The system displays text area 5. The user fills the area and clicks on the give response button 6. The system display response as a reply for selected feedback.
Alternate Flow of action	A1. If the user tries to give an empty response the system displays “please write some response” and goes back to step 4.

Table 3 20: Use case description for giving feedback

Use-case Name	Give feedback
Use case ID	UC-20
Used use-case	---
Brief description	Allow the customers to give their satisfaction scale and comment on the services they have used.
Participant actors	Customer
Pre-condition	The customers have to see and use the service, offers, event posted on the website to give feedback
Post-condition	The given feedback is registered on the admin feedback page
Course of action	Actions: 1. The user clicks on the specific service, event, or offer from the post 2. The system displays detailed information. 3. The user clicks on a give feedback link 4. The system displays the satisfaction levels and text area 5. The user selects satisfaction level and comment[A1] 6. The system sends feedback to the admin page.
The alternate flow of action	A1. If the user clicks the feedback button before selecting the satisfaction scale or without writing some comment the system will display “please select satisfaction scale or write some comment” and go back to 5.

Table 3 21: Use case table for Delete Account

Use case name	Delete Account
Use case ID	UC-21
Used Use case	Include: admin confirmation
Participant Actor	Customers
Brief description	Used to delete customer account if they want to remove themselves from the system
Pre-condition	The user must log in to the system
Post-condition	The customer account will be deactivated for 10 days after that it will be removed from the system.
Basic course of action	Actions: 1. User click on a profile from the menu 2. The system will display user account information. 3. The user click delete account from the hamburger menu 4. The system displays caution and warning about deleting the account if you choose yes the system will send a confirmation message by SMS and display a confirmation text field. 5. Enter confirmation code [A1]. 6. The system automatically deactivates the account and display the system as a visitor. 7. The system displays your account is deactivated for 10 days and will be deleted after that.
The alternate flow of Action	A1. If the user enters the wrong confirmation then the system will display 'Wrong confirmation' and go back to step 5.

Table 3 22: Use case description for sync data from another system

Use case name	Sync data from another system
Use case ID	UC-22
Used Use case	---
Participant actor	System
Brief description	This use case is used to fetch data automatically from the hotel management system. Which is not part of our system.
Pre-condition	Customer’s transaction in the hotel management system
Post-condition	Data is synced into the loyalty system
Basic course of action	Actions: 1. Scan user bar code or QR code. 2. The hotel management system much the id with the code value. 3. The loyalty system copy the customer transaction data by referring customers ID
The alternate flow of Action	---

3.2.5 Activity Diagram

Activity diagrams are graphical representations of the workflow of stepwise activities and actions with support for choice, iteration, and concurrency. In UML, activity diagrams are intended to model both computational and organizational processes, as well as data flows intersecting with the related activities. These diagrams primarily show the overall flow of control and the flow of data between activities [5].

In this topic, we have included activity diagrams of our system which show the flow of activities to manipulate data, insert data, account creation, account management, and others. The main activities of the user are shown in figure 3.6-3.12.

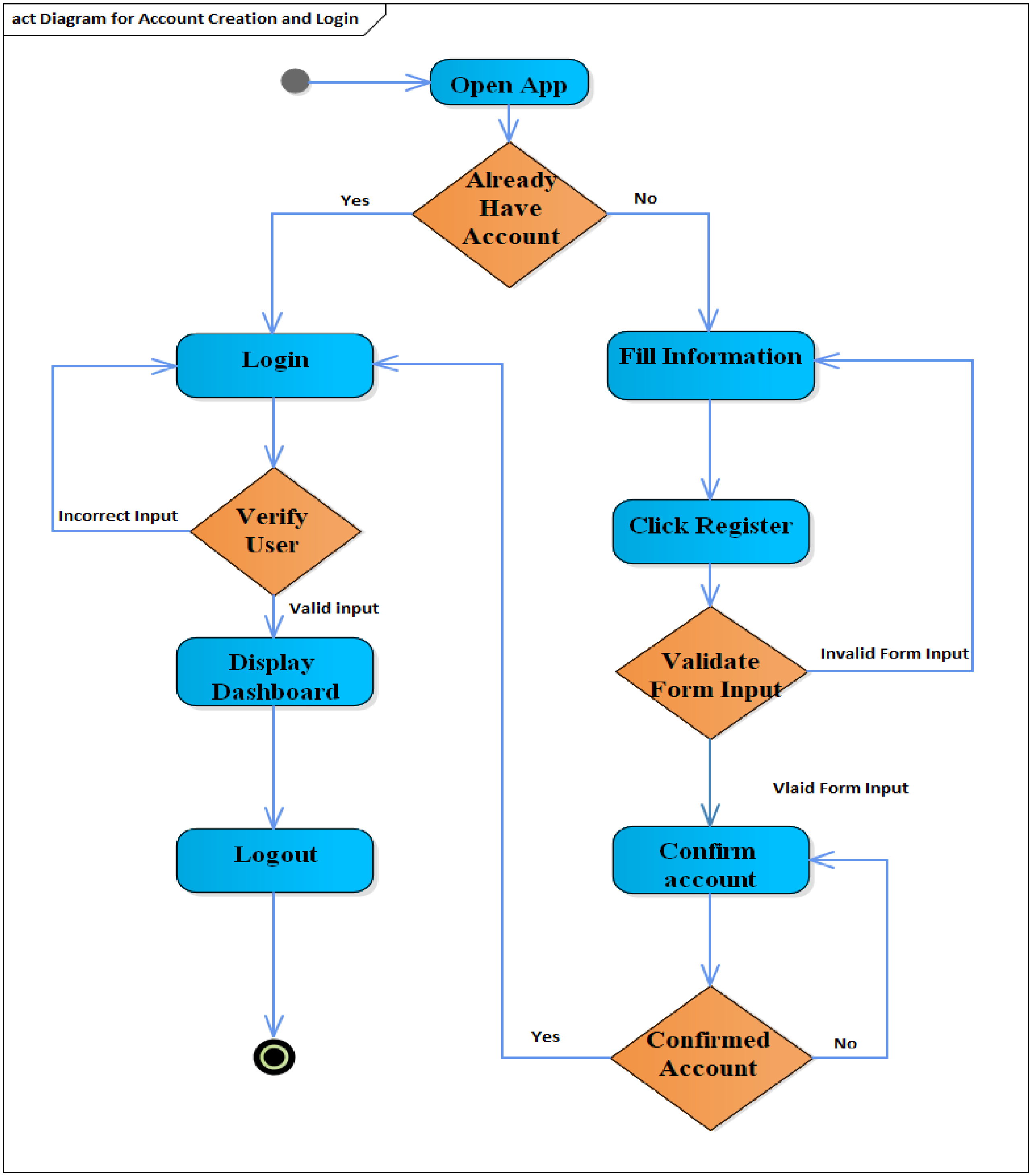


Figure 3 6: Activity diagram for account creation and login

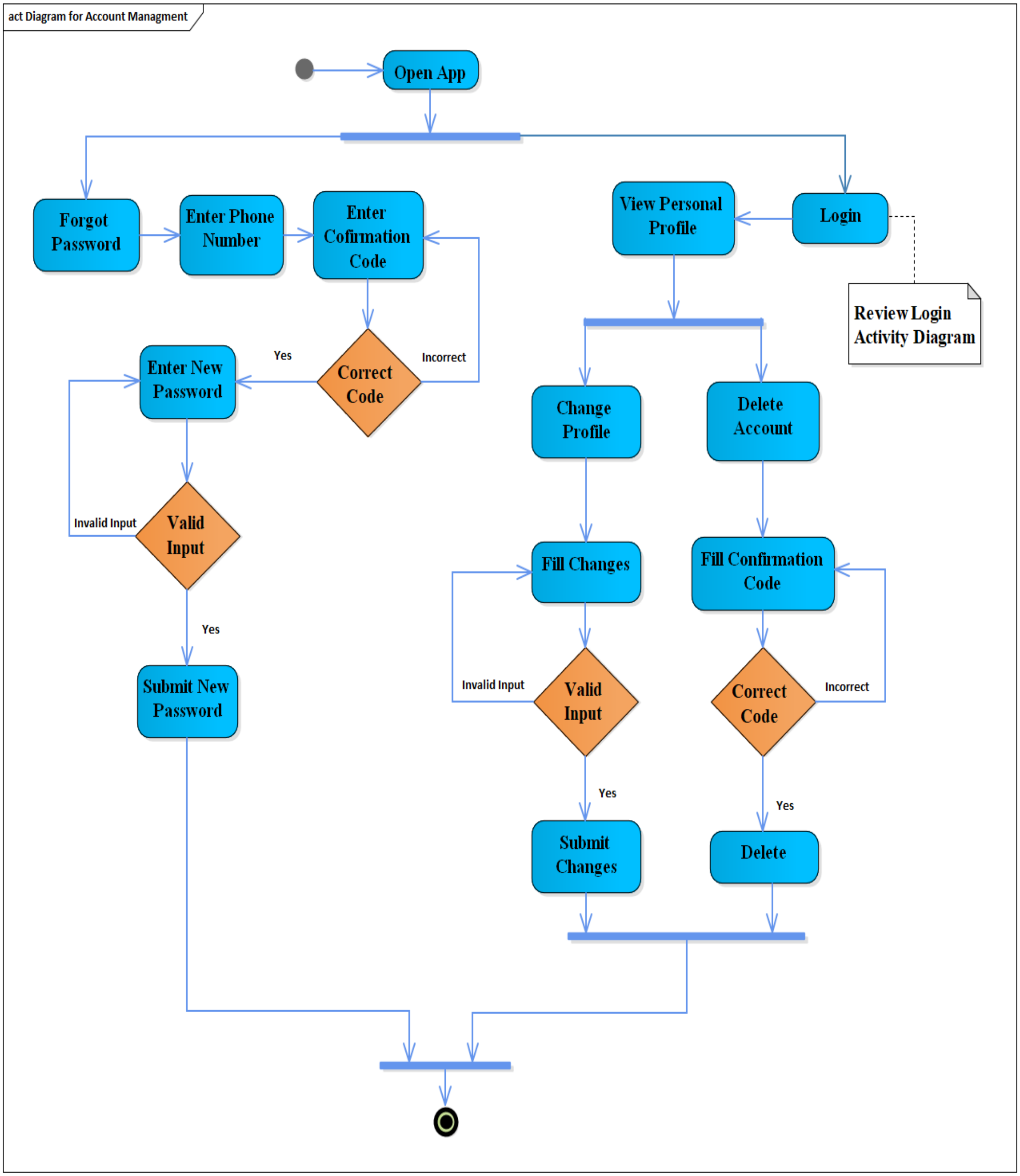


Figure 3 7: Activity diagram for account management

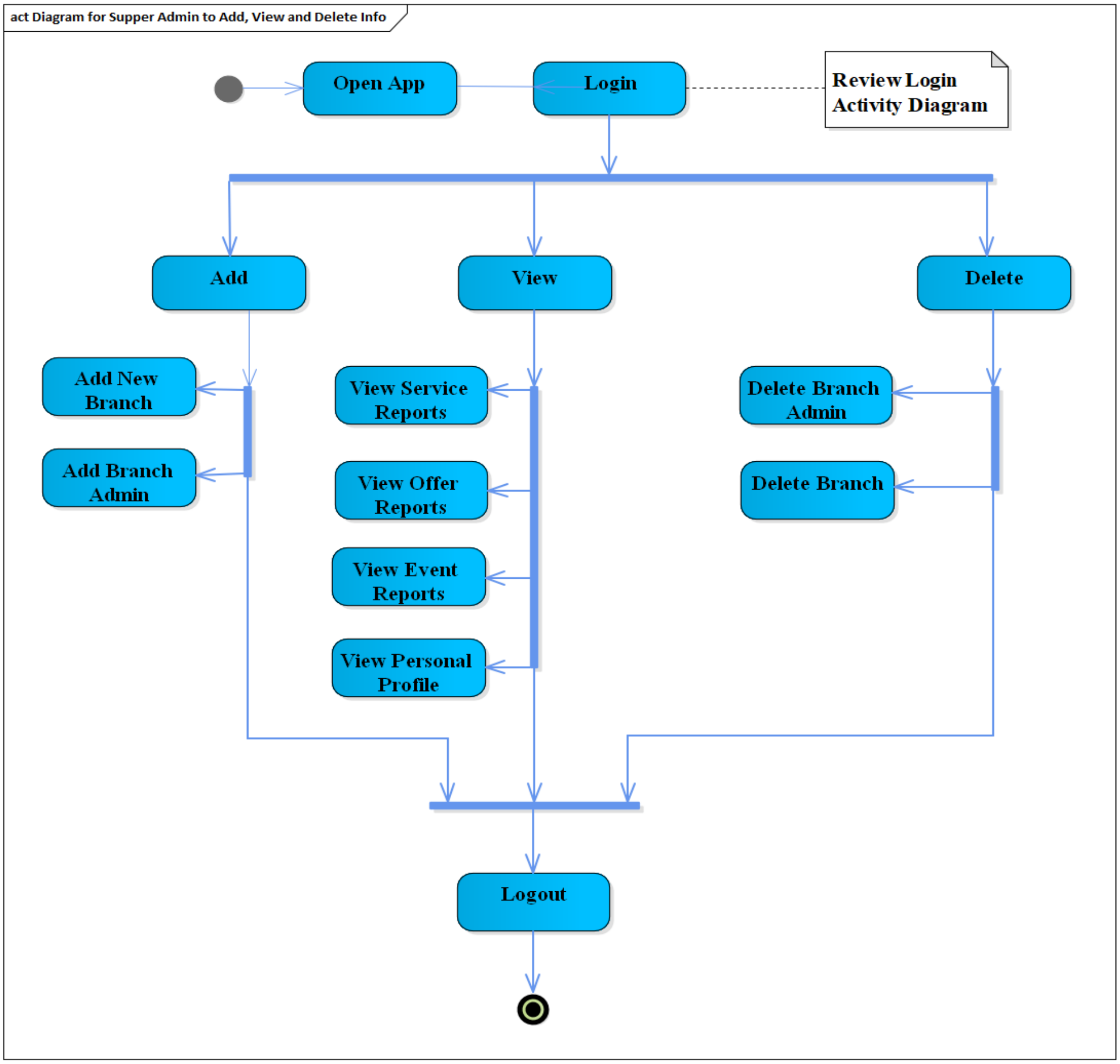


Figure 3 8: Activity diagram for add, view, and delete by supper admin

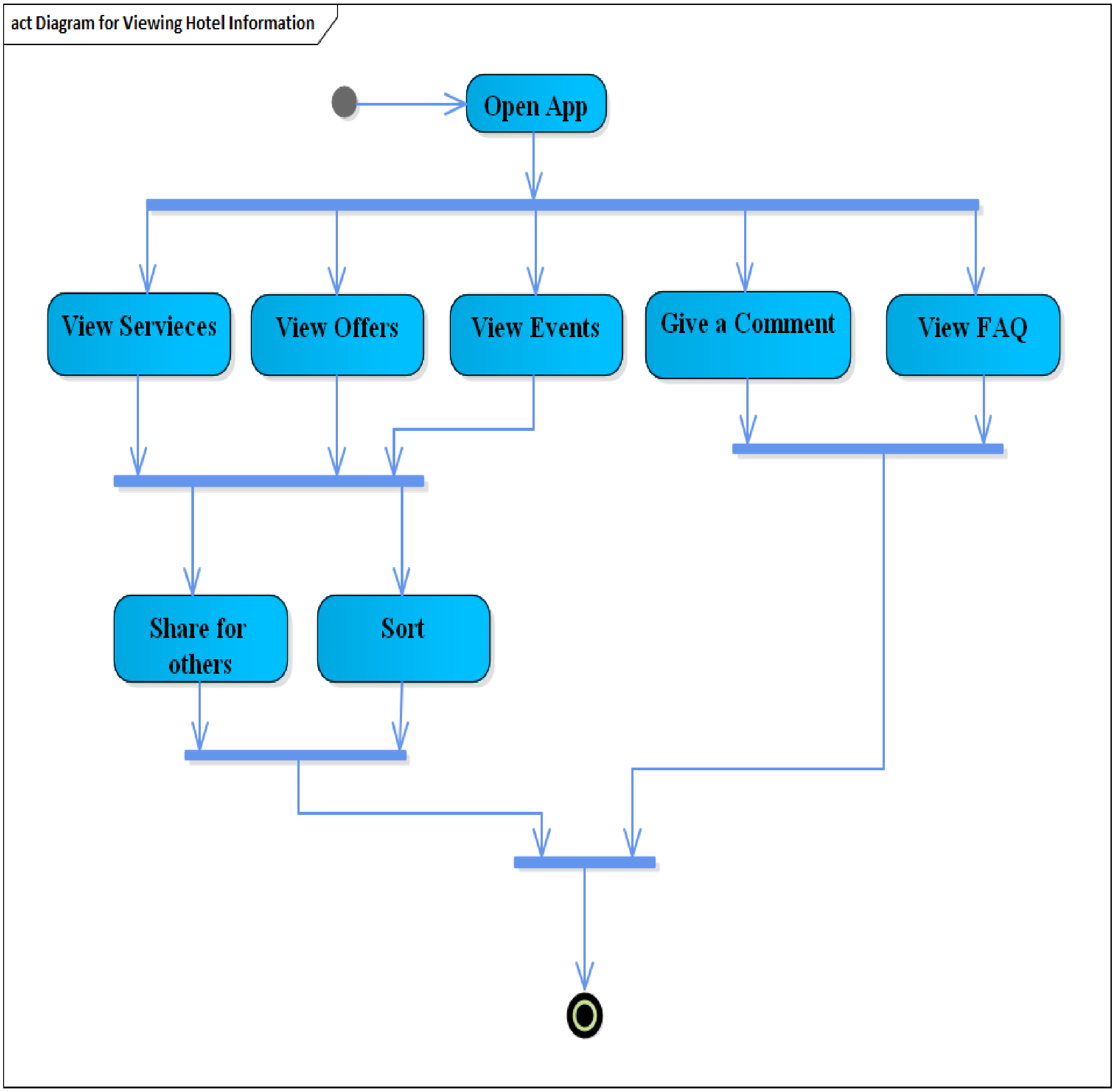


Figure 3 9: Activity diagram for viewing hotel information

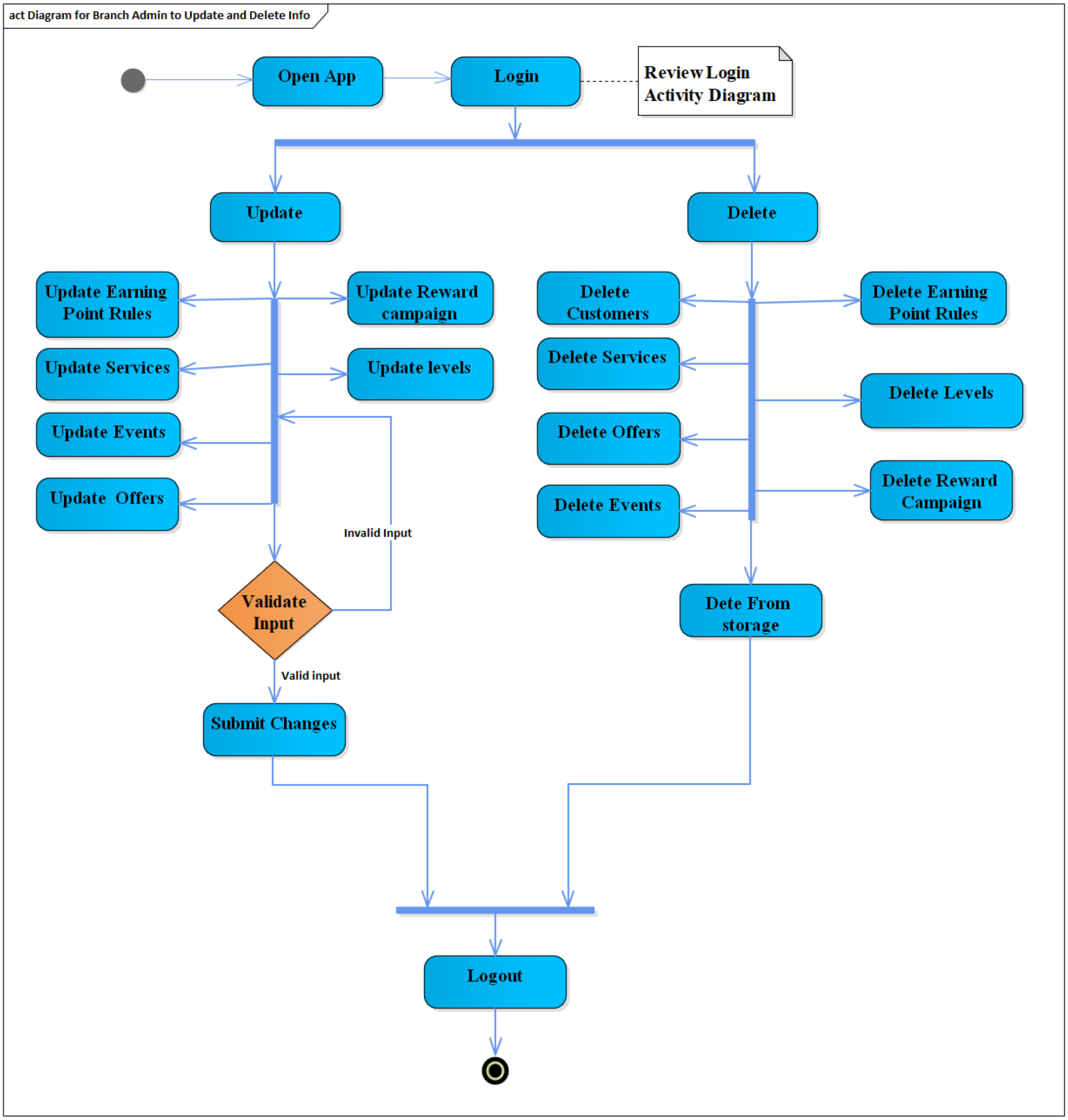


Figure 3 10: Activity diagram for branch admin to update and delete information

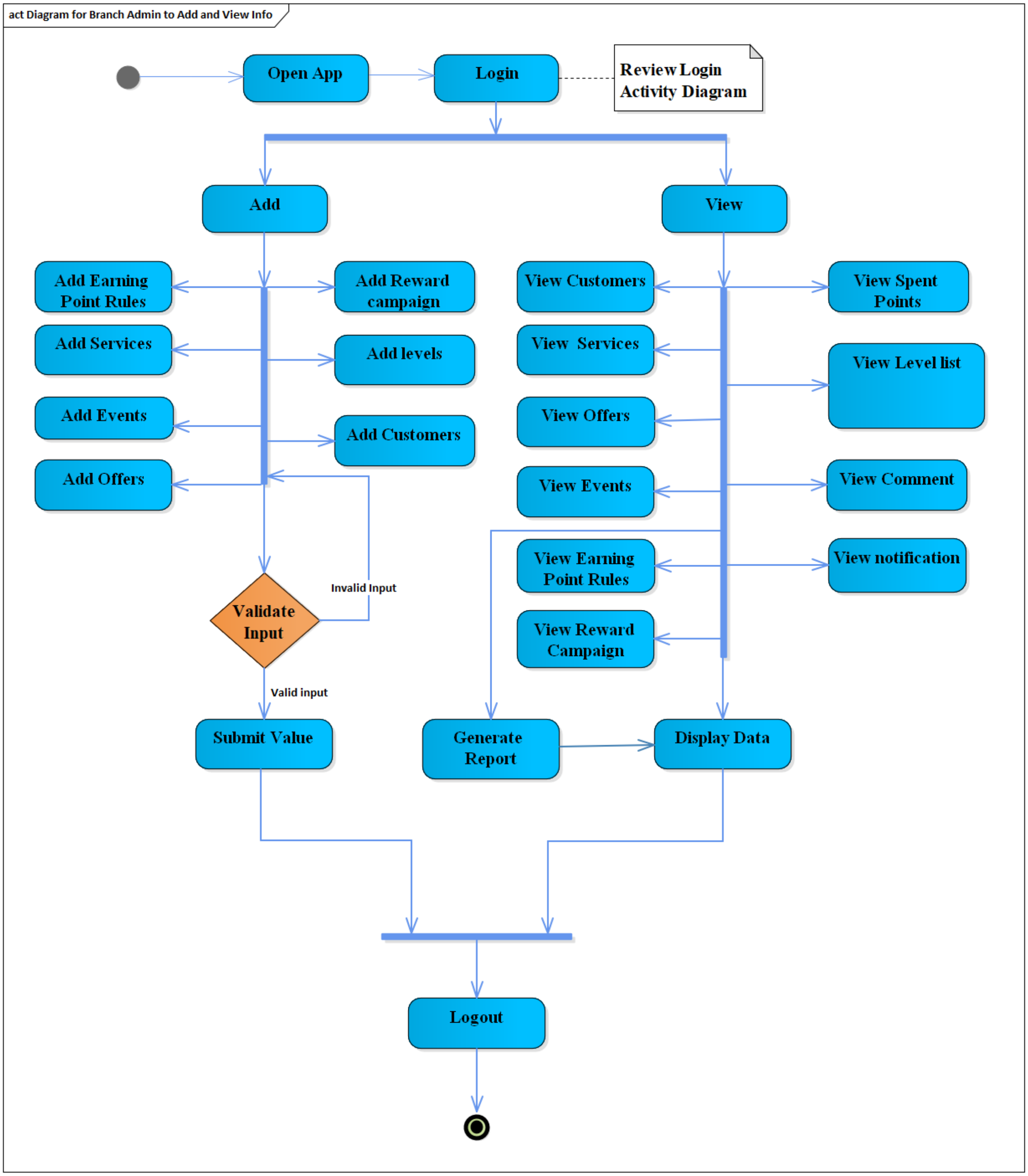


Figure 3 11: Activity diagram for branch admin to add and view information

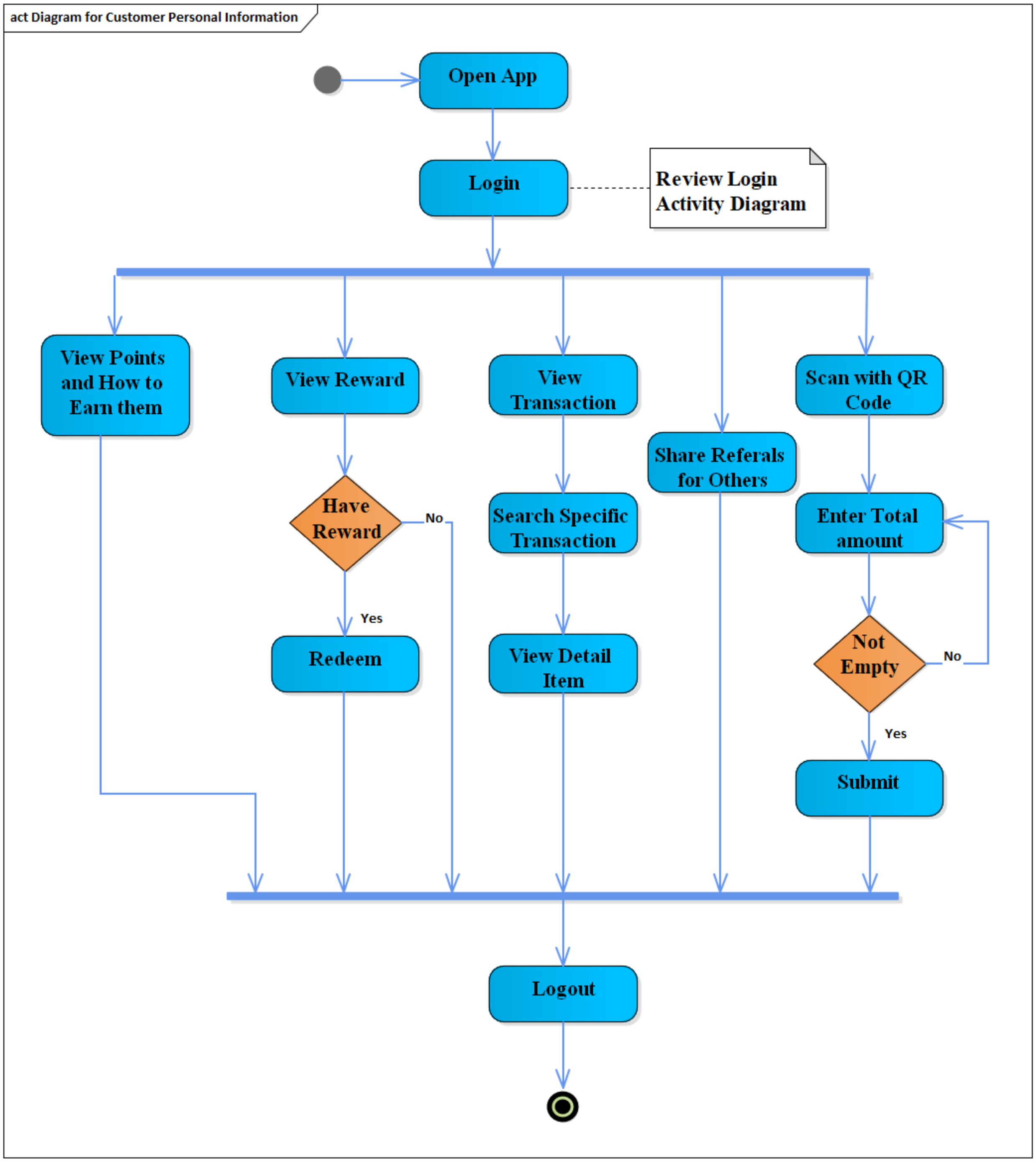

act Diagram for Customer Personal Information
Open App
Login
Review Login Activity Diagram
View Points and How to Earn them
View Reward
View Transaction
Share Referals for Others
Scan with QR Code
Have Reward
No
Yes
Redeem
Search Specific Transaction
View Detail Item
Enter Total amount
Not Empty
No
Yes
Submit
Logout

Figure 3 12: Activity diagram for a customer to manipulate personal information

3.3. dynamic modeling

Dynamic modeling is concerned with modeling the behavioral or dynamic dimension of a system, how the elements that collaborate to constitute a system interact to provide the functionality of the system and is used for system, subsystem, and class specification within the road map to capture in a specification model how the construct will satisfy its requirements, what elements and their relationships collaborate to constitute the construct, and how these elements interact to provide the functionality to its users.

3.3.1. Sequence diagram

A sequence diagram depicts the dynamic behavior of an entity via a temporal focus on interaction, how the elements that collaborate to constitute an entity interact over time to provide the functionality of the entity, using classifier roles and messages or using instances and stimuli.

In this part of the document, we aim to show how the different elements of the system interact with each other and exchange messages to fitful the objectives [5]. We have discussed the main sequences of the system as shown in the figure-3.13 — 3.16.

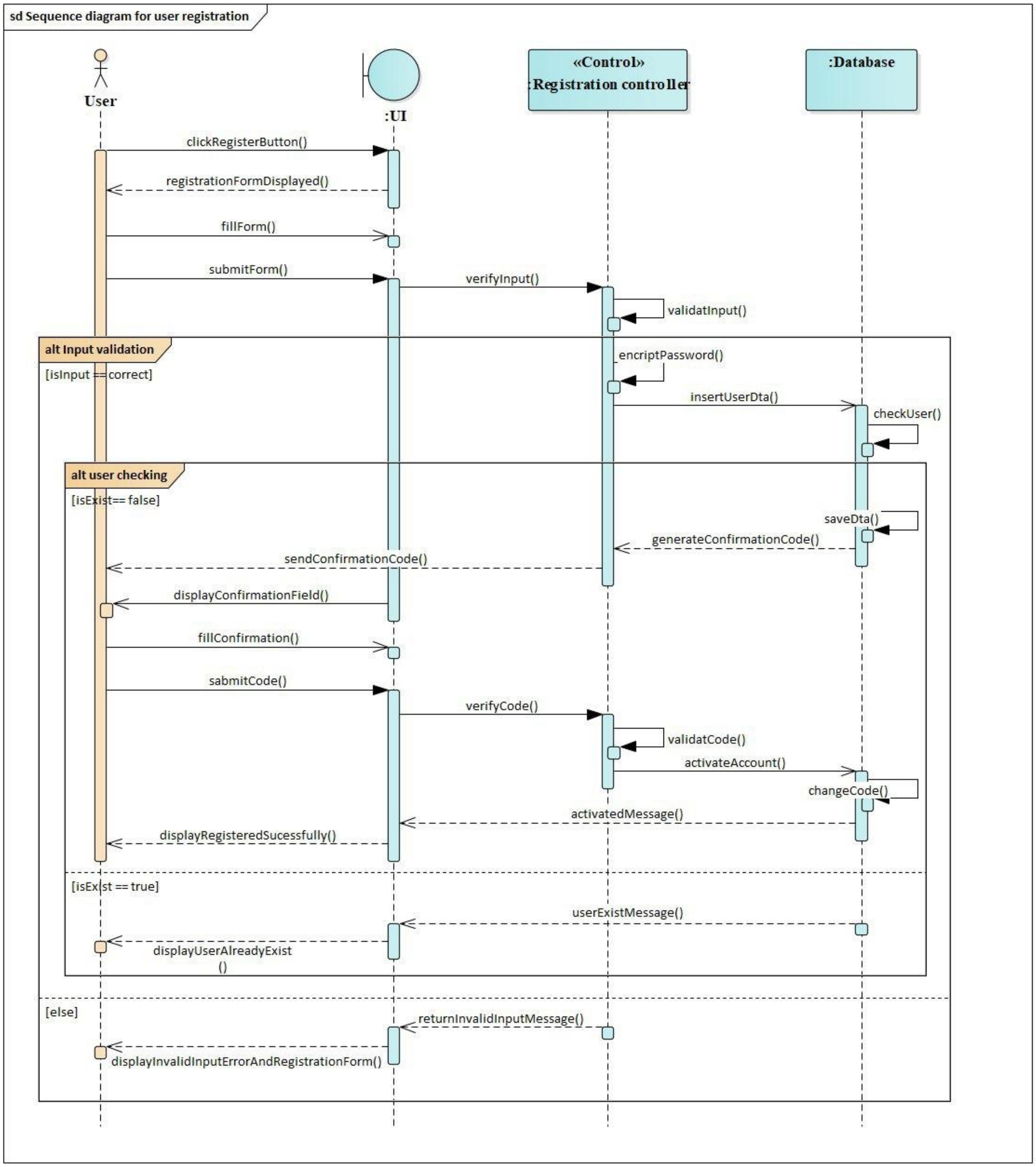


Figure 3 13: Sequence diagram for user registration

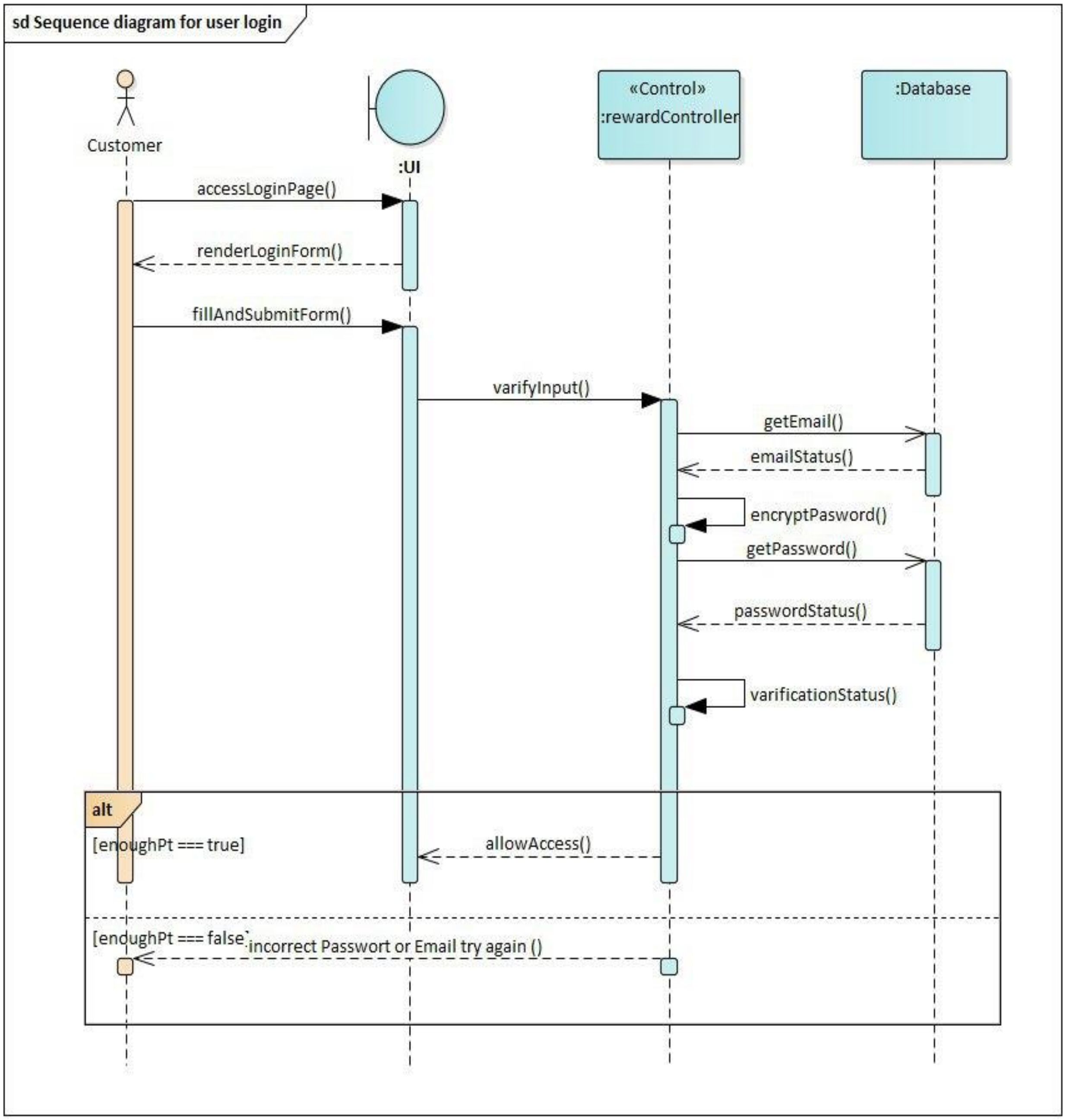


Figure 3 14: Sequence diagram for user login

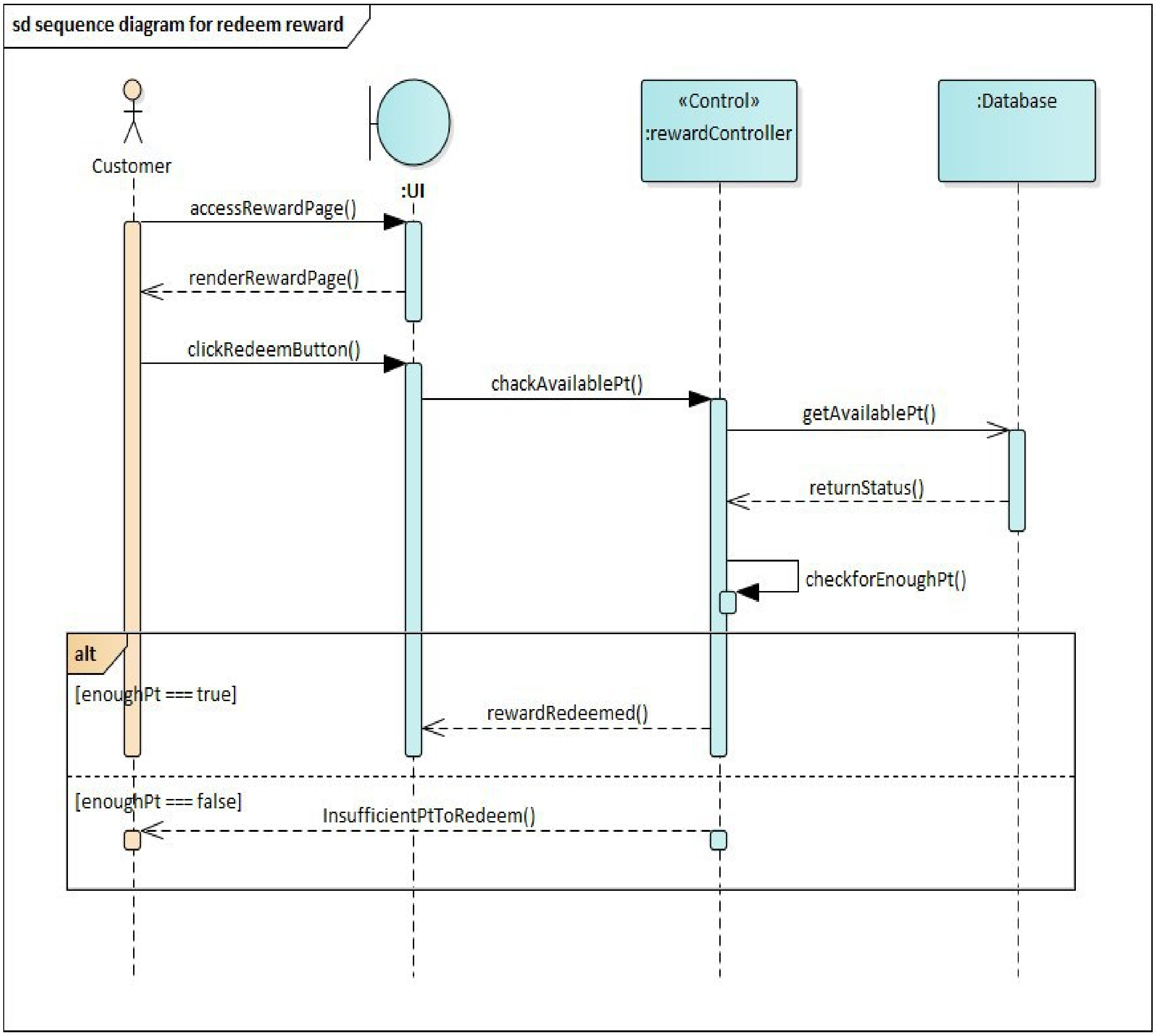


Figure 3 15: Sequence diagram for redeeming reward

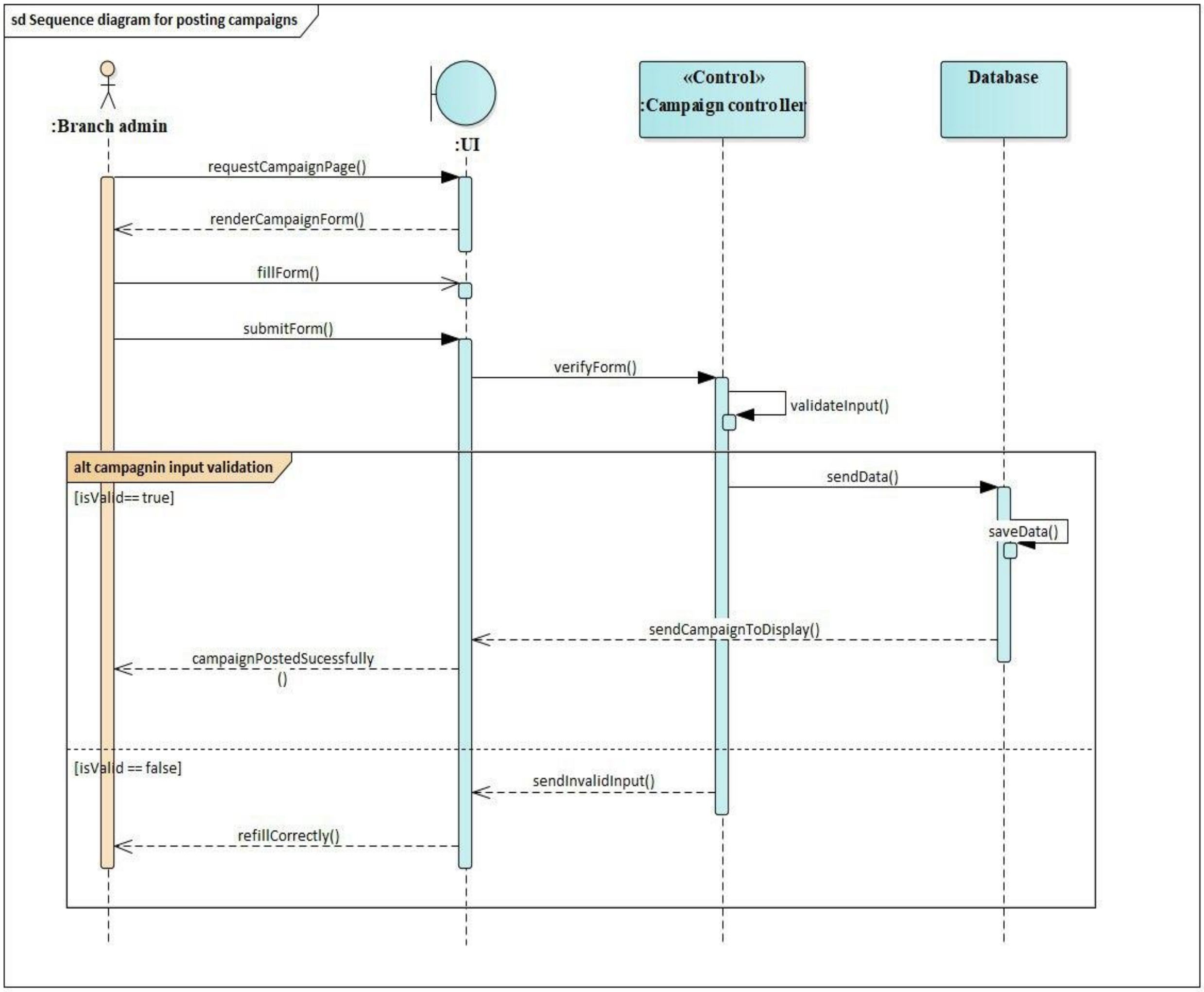


Figure 3 16: Sequence diagram for post-campaign

4.9.1 State Diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states, sometimes, this is indeed the case, while at other times this is a reasonable abstraction [5].

In this document, we aim to show the basic states of the proposed system. The main state diagrams of our system are shown in figure-3.16-3.21.

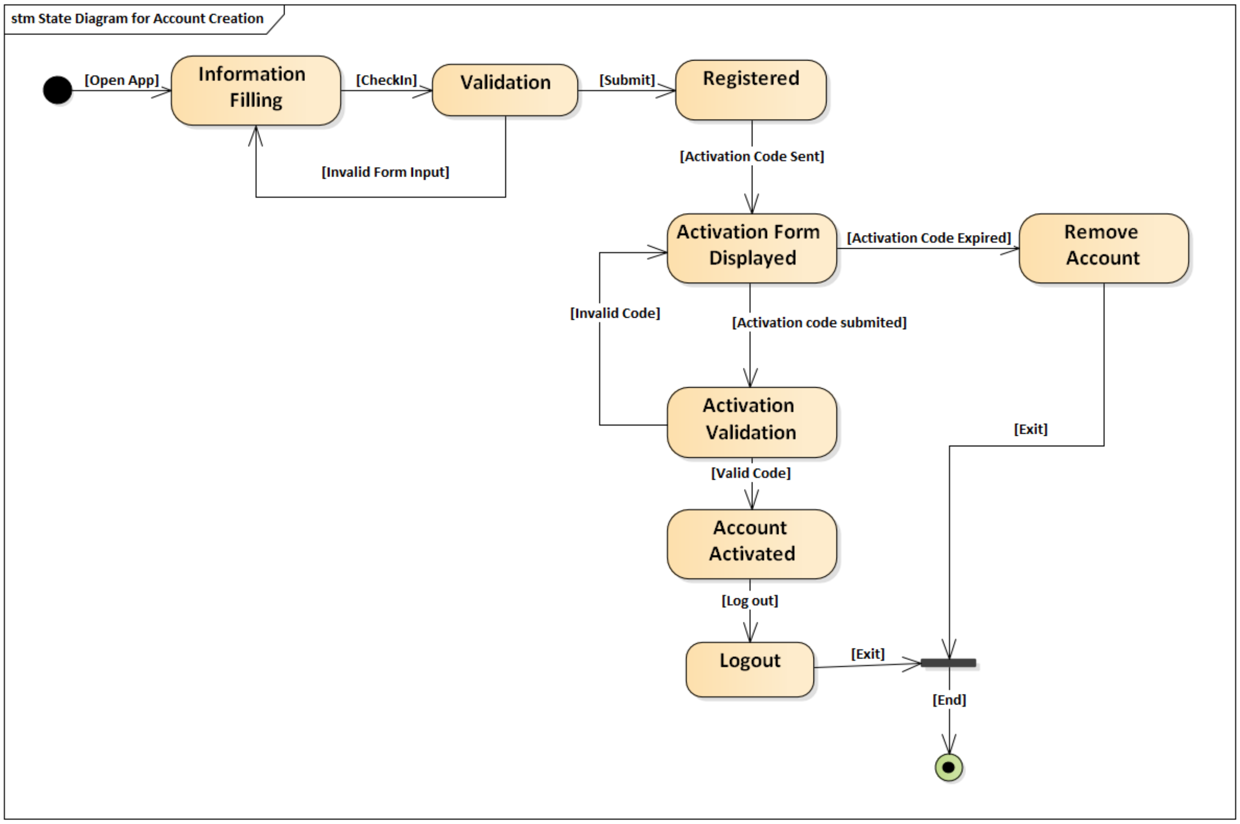


Figure 3 17: State Diagram for Account Creation

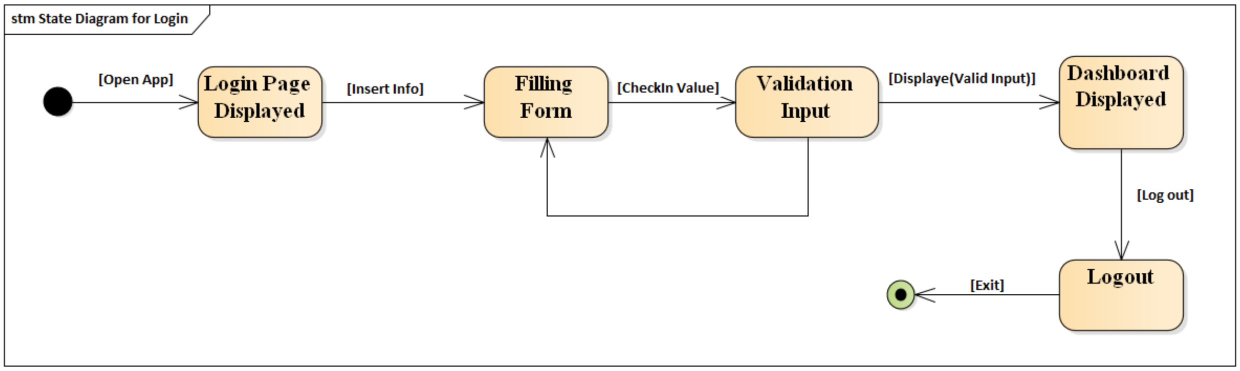


Figure 3 18: State Diagram for Login

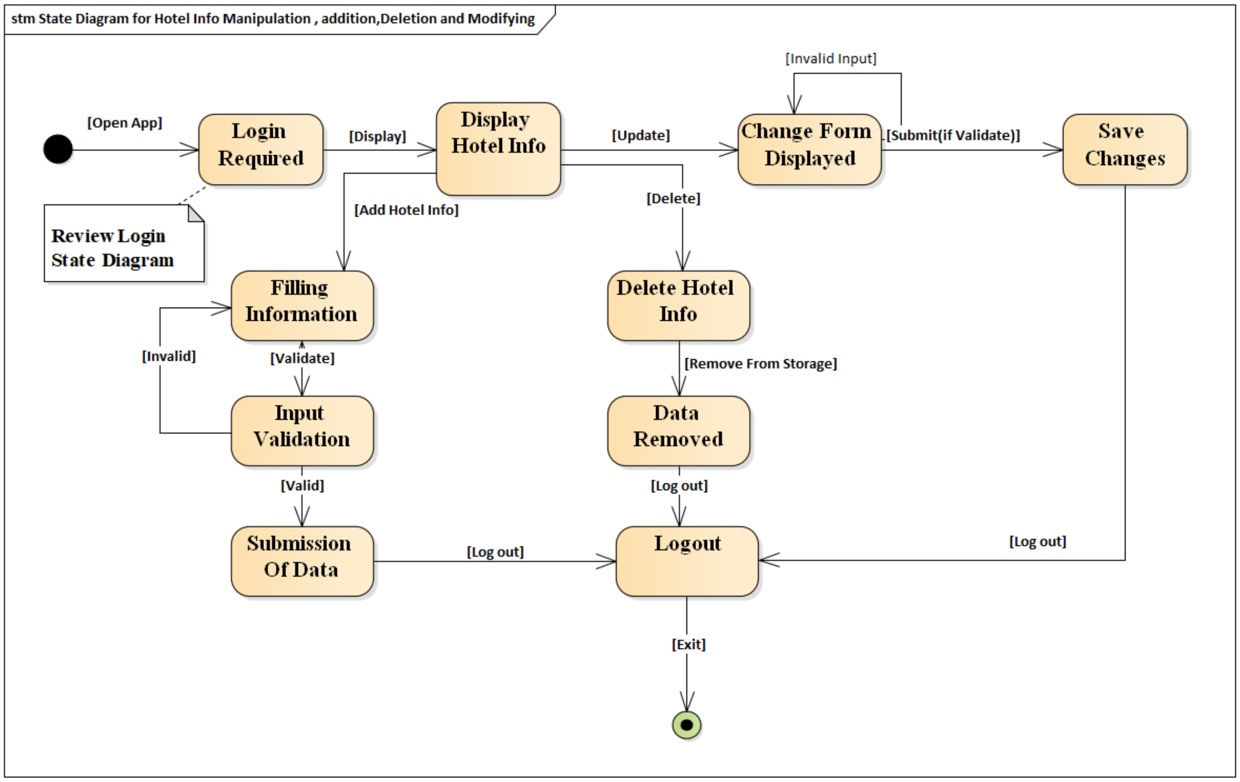


Figure 3 19: State Diagram for Manipulation of Hotel Information

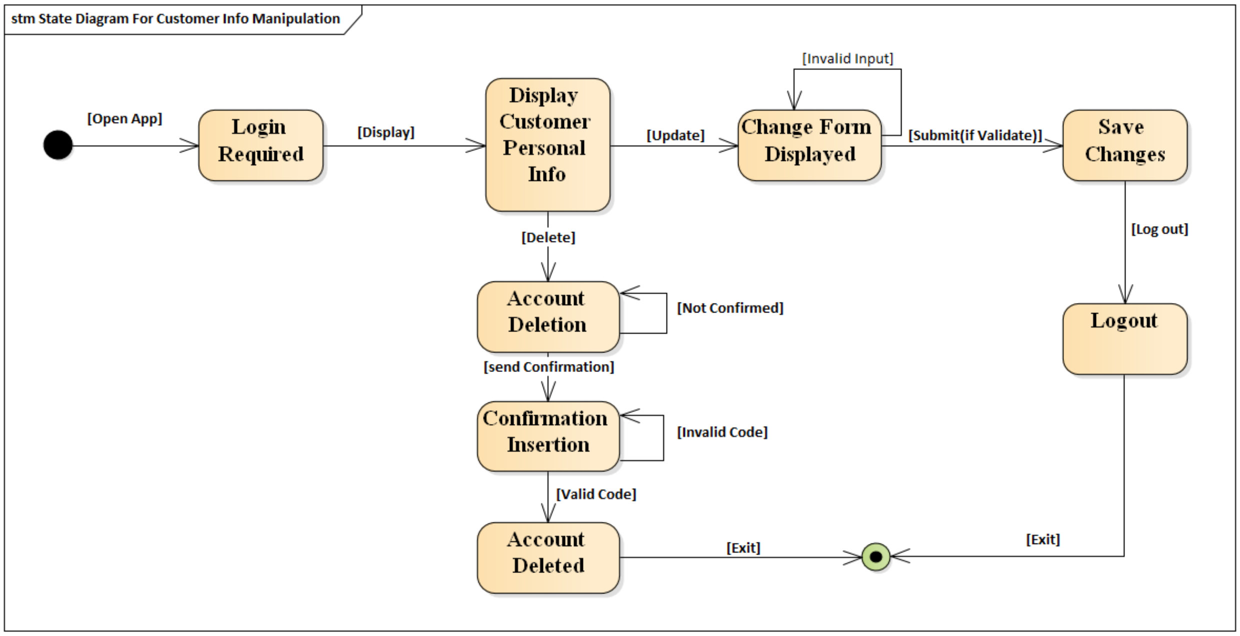


Figure 3 20: State Diagram for Customer’s Information Manipulation

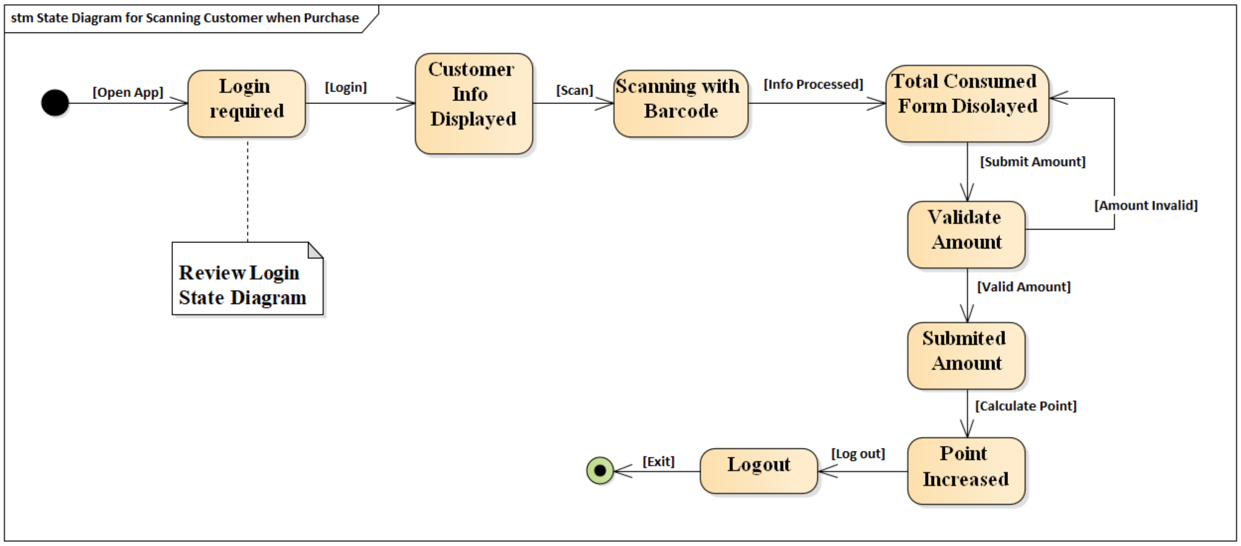


Figure 3 21: State Diagram for Scanning a Customer

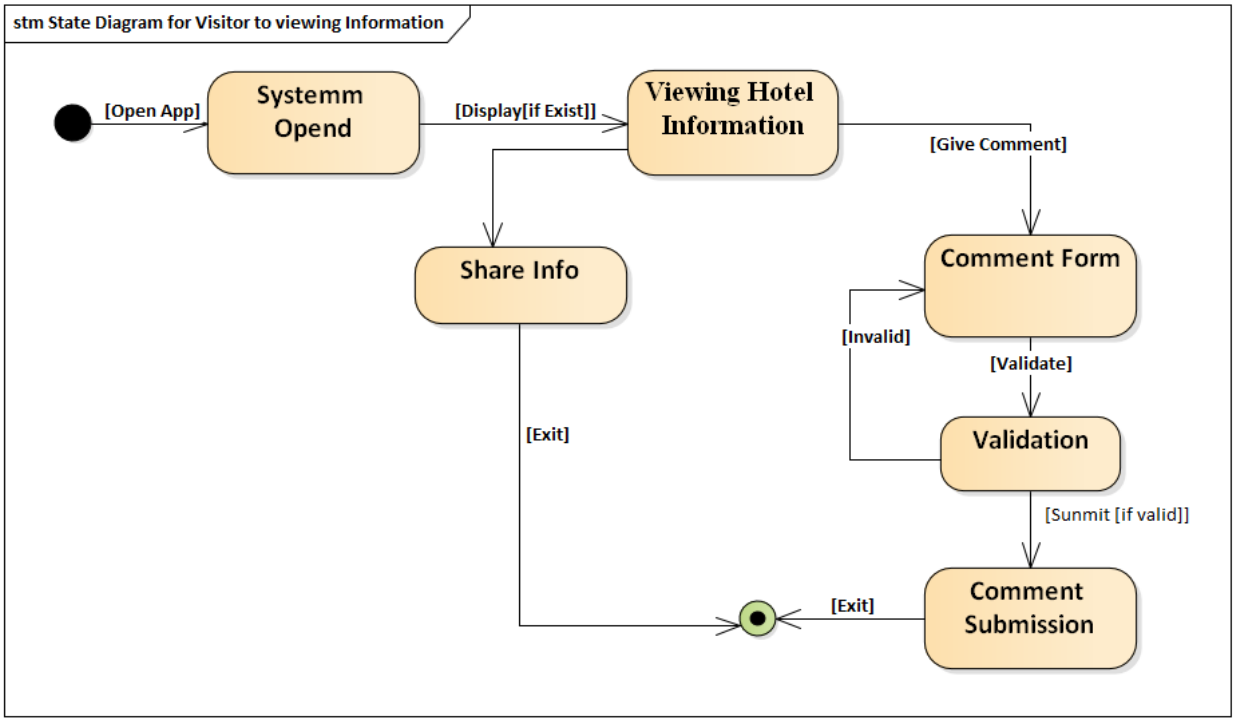


Figure 3 22: State Diagram for Visitor Viewing Information

3.4. Class-based modeling

A class diagram is a UML structure diagram that shows the structure of the designed system at the level of classes and interfaces, shows their features, constraints, and relationships — associations, generalizations, dependencies, etc.

In this part, we have shown how the system is structured at the class level and how these classes relate to each other and their multiplicity.

3.4.1. Class Identification

Class identification is a process of extracting or obtaining classes from a given scenario to represent a real-world problem [5]. In our project, we have identified the following classes which are the main elements of the system and we have used a noun extraction method to identify the classes.

- User
- Supper admin
- Branch admin
- Loyal customer
- Visitor
- Branch
- Recommendation
- Transaction
- Services
- Offers
- Events
- Point
- Reward
- Levels
- Comments
- Synchronization

In the above class list, we have a recommendation class, which is used for recommending items for customers by considering the behavior of customers from their frequent purchase by using some data mining algorithms.

Data mining is a process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers to develop more effective marketing strategies, increase sales, and decrease costs [12].

In our system, we have used an association rule mining technique. **Association rule** mining is used to identify relationships among a set of items in a database. These relationships are based on the co-occurrence of the data items. Therefore, the main purpose of using this technique in this system is to find out synchronous relationships among items purchased, by analyzing frequently used items with their transactions and to use these items as a reference during decision making [13].

Different algorithms can find any association rules. But in this system, we have used the Apriori algorithm.

Apriori algorithm is used to create candidate item sets of a given size and then, scan data sets to check if their counts are really large. This process is iterative. This algorithm uses the following steps to identify the recommended items:

Step 1. Apply minimum support to find all the frequent sets with k items in a database.

Step 2. All singleton item sets are the candidate in the first path. Any item with less than the specified support value is eliminated.

Step 3. Generate an association rule which has the confidence value greater than or equal to the specified support value.

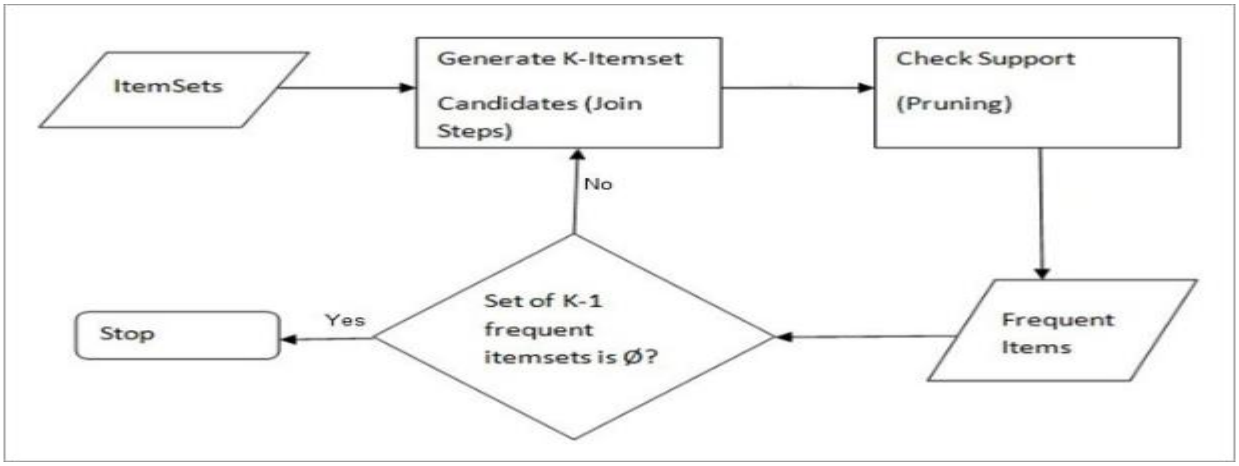


Figure 3 23: Flow chart of the Apriori algorithm

1.4.2. Class Diagram

The class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. In this topic, we have depicted the static structure of our system as shown in figure-3.22.

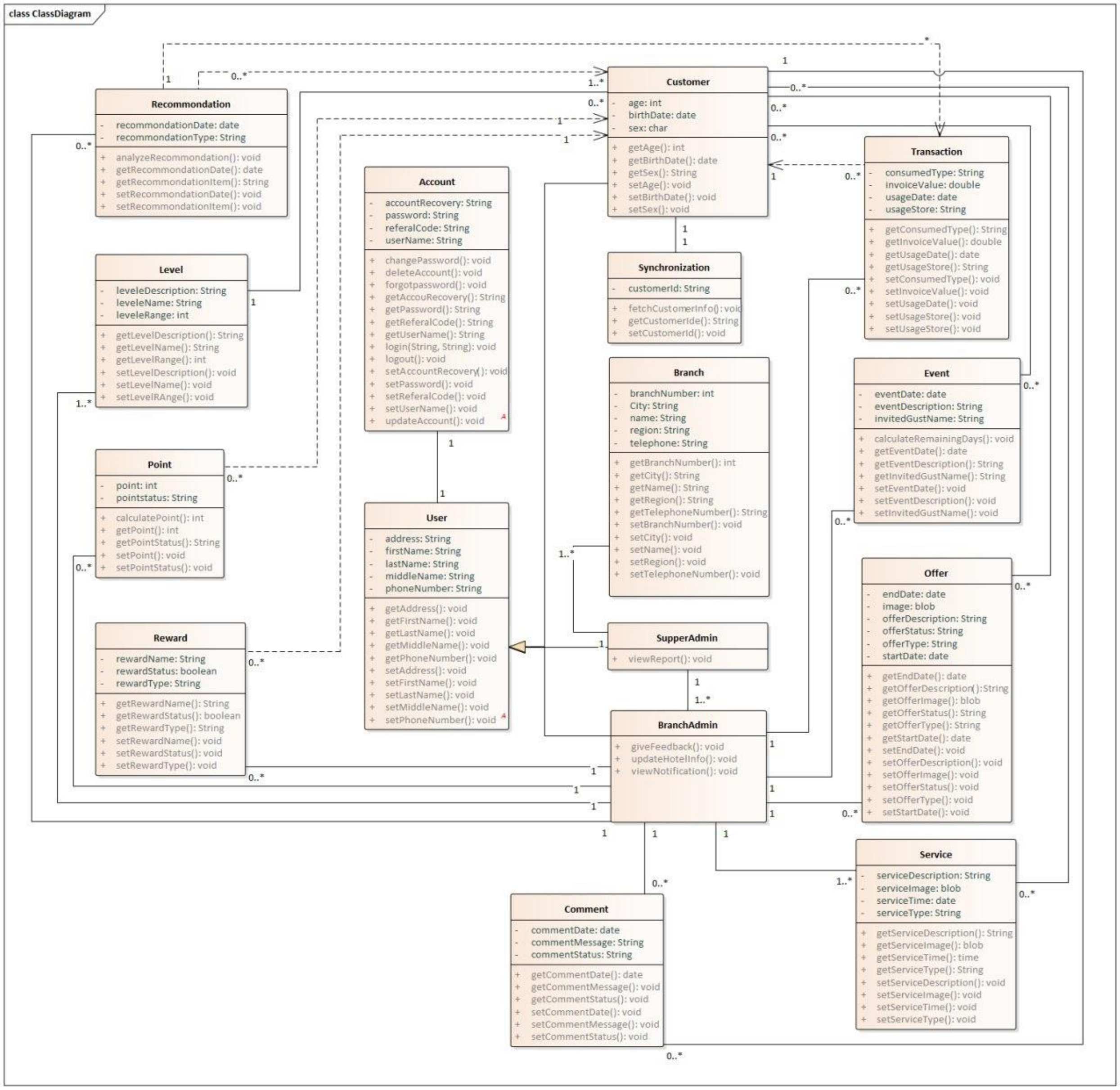


Figure 3 24: Class Diagram of CLP

Chapter 4

System Design

4.1. Overview

In this chapter, we focused on translating the analysis model into a design model, and this design model shows the direction of how our system is built and to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities. We designed our system as multilayered specifically three-layer architecture which shows the client-server architecture and consists of presentation, application processing, and data management layers.

4.2. System Design

System design is the process of designing the elements of a system such as the architecture, modules, and components, the different interfaces of those components, and the data that goes through that system [3].

4.3. Design Goals and Objectives

The design goal of the system is derived from the non-functional requirements. It can be used as an easy way to control the implementation tool to achieve the best result based on the particular design tool. The design objective is to develop the system with high quality.

Some of the design goals that our system consider are:

- **Usability** – The software user interface must be usable for its target user/audience (customer and system admins). Default values for the parameters must be chosen. So, they are a good choice for the majority of the users.
- **Reliability** – The software can perform a required function under stated conditions for a specified period.
- **Performance** – Our system has a short response time for given access, accuracy, efficiency, and high throughput (high rate of work), since the back end will be built with Nodejs framework, which is a non-blocking and asynchronous, no one is waiting for others process to complete when accessing in the same time that is many users access the system in an instant.

- **Availability** — The system is accessible anytime, anywhere, and via computer, mobile devices, and tablets with an internet connection and offline (in case of mobile application).
- **Maintainability** – It is easy to maintain the system during correcting defects or their cause, repair or replace faulty or worn-out components without having to replace still working parts, and prevent the unexpected working condition.
- **Security** – The software can withstand and resist hostile acts and influences.

4.4. System Architecture

The architecture of the system provides an overall structure of the components of the system and how that structure provides conceptual integrity for the system. As mentioned in the overview (4.1), we used a layered based architecture which has three layers namely Presentation, Process (Logic), and data layers.

1. **Presentation Layer —** This is the topmost level of the application. The presentation layer displays information related to such services as the user interacts and provides user interfaces. It communicates with the process layer by which it puts out the results. In simple terms, it is a layer that users can access directly (such as a web page, mobile application).
2. **Logic Layer –** Concerned with application-specific functionality by performing detailed processing.
3. **Data Layer —** The data layer includes the data persistence mechanisms (database servers, file shares, etc.) and the data access layer that encapsulates the persistence mechanisms and exposes the data. The data access layer should provide an application program interface (API) to the application layer that exposes methods of managing the stored data. This layer is in our system the database that is MongoDB. The architecture of the system is shown in fig. 4.1.

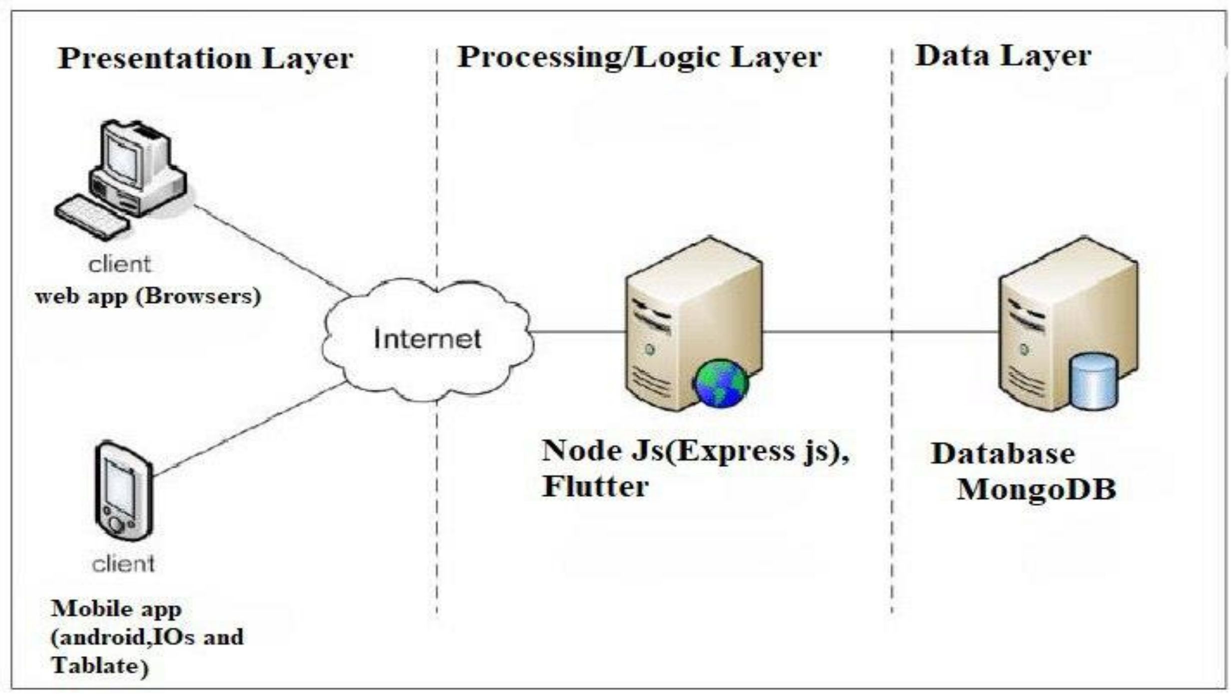


Figure 4 1: System Architecture of CLP

4.5. System Decomposition

System decomposition begins by decomposing the system into cohesive, well-defined subsystems. Subsystems are then decomposed into cohesive, well-defined components. In our system, we have five main modules as shown in figure 4.2.

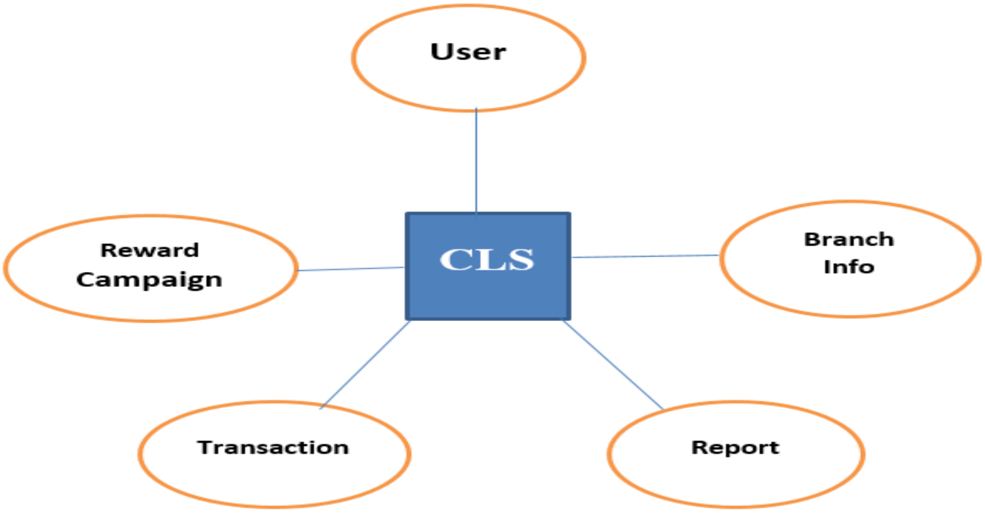


Figure 4 2: System Decomposition of CLP

4.5.1. Module Description

As we see the system decomposition in figure 4.2, our system consists of five modules: user, branch information, reward campaign, reports, and transactions. In this topic, we have covered a detailed description of these modules.

User

This module allows the system users to create, update, view, and delete different accounts based on their access privilege and access levels.

- **Provided Interface:** User account list
- **Required Interface:** Homepage

Processing:

Create Account

Users must fill in the required information and they can create their account but in the case of branch admins, the account is created by the super admin.

Table 4 1: User account information

Super Admin	Branch Admin	Customer
• Username(ID) • Password • First name • Middle Name • Last Name • Sex • Phone Number • Email	• Username(ID) • Password • First Name • Middle Name • Last Name • Sex • Phone Number • Email • Branch address • Branch number	• Username(ID) • Password • First name • Middle Name • Last Name • Sex • Birth Date • Phone Number • Email • Recovery answer • Referral code

Access Account

When the system users want to access the system, they have to enter their username and password.

If the users forgot their password or username, they can recover it by filling the recovery question they have stetted when they create their account.

Super Administrators have a higher level of authorization on the system. They can reset Branch Administrators' passwords, review all transactions, view customer information, can analyze generated reports from each branch, and can also generate the report.

Delete Account

When it is necessary to delete the branch admin account, the super administrator is responsible to do it. But in the case of deleting a customer account, the customer themselves delete the account after they receive the confirmation automatic SMS from the system to their phone.

View /Update Information – This proposed system will allow users (Admin(s) and Customers) to view and update information.

- All users can see their personal information (profile information)
- All users can update their personal information (profile information)
- All users can see posted services

II. Branch Information

This module is responsible for all the required interface and processing logic for retrieving, populating, creating, and sorting hotel information.

- **Provided Interface** – The list of all previous hotel information.
- **Required Interface –** The home page of Admin and the pages for customers.

Processing:

Super Admin

Super admin can add a branch, branch admin, and view all information related to the hotel.

- **Adding Branch** – Supper admin can add branch hotels with the following parameters: name, address, telephone, branch number.
- **Adding Branch Admin –** branch admin is registered with the following attributes: first name, last name, address, phone number, and user name.

- **Viewing Hotel Information –** Supper Admin can see all generated reports, but cannot add customers and other hotel services. And she/he can see the whole system information that the customer and the branch admins view.
- **Deleting Branch and Branch Admin Account** –Supper admin can delete the branch if it is not needed, and he/she also delete the branch admin account.

Branch Admin

Branch admin can add, delete, and update the hotel information.

- **Adding services and Levels**– Branch Admin can add services and levels with their respective types, date-time, and description.
- **Deleting, updating –** Branch admin can delete services and he/she also update the hotel's information as needed.

III. Generate reports

This module is responsible for all the required interfaces and processing logic for generating reports of the hotel information.

- **Provided Interface –** The list of all reports if exists.
- **Require Interfaces –** Supper Admin and branch Admin home page.

Processing:

Supper Admin/Branch Admin

This module will allow both suppers, admin, and branch admin, to generate the following reports.

- GR01-Generate report about the active loyal customer
- GR02-Generate report about customer referred customer
- GR03-Generate report about customer feedback with their time
- GR04-Generate report about a customer transaction
- GR5-Generate report about rewards, point spent, and also the date
- GR6-Generate report of customer distribution like Male, female

IV. Reward Campaign

Reward campaigns provide information about all rewards within the loyalty program with information about type, status, cost in points, limits of use, and how many times reward has been used by customers, and time boundaries of activity[2].

This module is responsible for all the required interfaces and processing logics for reward campaign activities like points, rewards, offers, events.

- **Provided Interface –** provides all reward campaign lists.
- **Required Interface –** user home page.

Processing:

Customer – customers will perform the following activities using this module.

- Customers can see their points (active points, used points, expired points), these points are calculated from profits when customers purchase something with the help of a scanner to identify who is purchasing and adding the points.
- Customers can see the reward available for redeeming.
- Customers can see offers and events.
- Customers can see earning point rules.
- Customers can see notifications about rewards.

Branch Admins – branch admin will perform the following activities using this module.

- Branch admin can add earning point rules with their point limit and description.
- Branch admin can add events with their time-date, gust names, and description.
- Branch admin can add offers.
- They can see the reward list and their recommendations coming from data mining concepts.
- They can see the earning point rule
- Branch admin can see offers and events.

Any reward campaign can be assigned manually with the help of item recommendations and the admin can create a new or use existing reward campaign when it is necessary with the following information.

Table 4 2: Reward campaign

Field	**Description**
Name	Name of the reward displayed in views
Active	Reward campaign current status. Option includes: Active/Inactive
Campaign type	Reward type Options include: • Discount code • Free delivery • Gift • Invitation for the event • Custom campaign
Cost in points	How many points customer must spend to redeem the reward
Active from	Day from which reward is active, i.e., visible and available to use for customers
Active to	Day until reward can be redeemed. After that day, the reward will not be visible for the customer and unavailable to use
Customers	Show the number of customers who could redeem the reward.

V. Transaction

This module is responsible for all the required interfaces and processing logics for managing transactions.

- **Provided Interface –** The list of all transactions if exists.
- **Require Interfaces –** Customer’s and branch Admin’s home page.

Processing:

Customers can get all their transactions they perform when purchasing, and admins can see all transactions made in the hotel.

- See all transactions
- Search specific transactions with their details like date, store, and amount.
- Item recommendations can be made using these transactions.

4.6 Component Diagram

Component diagrams are used to visualize the organization of system components and the dependency relationships between them. They provide a high-level view of the components within a system [5]. In our system, we have used different components to understand the system and how its components are connected as shown in figure 4.3.

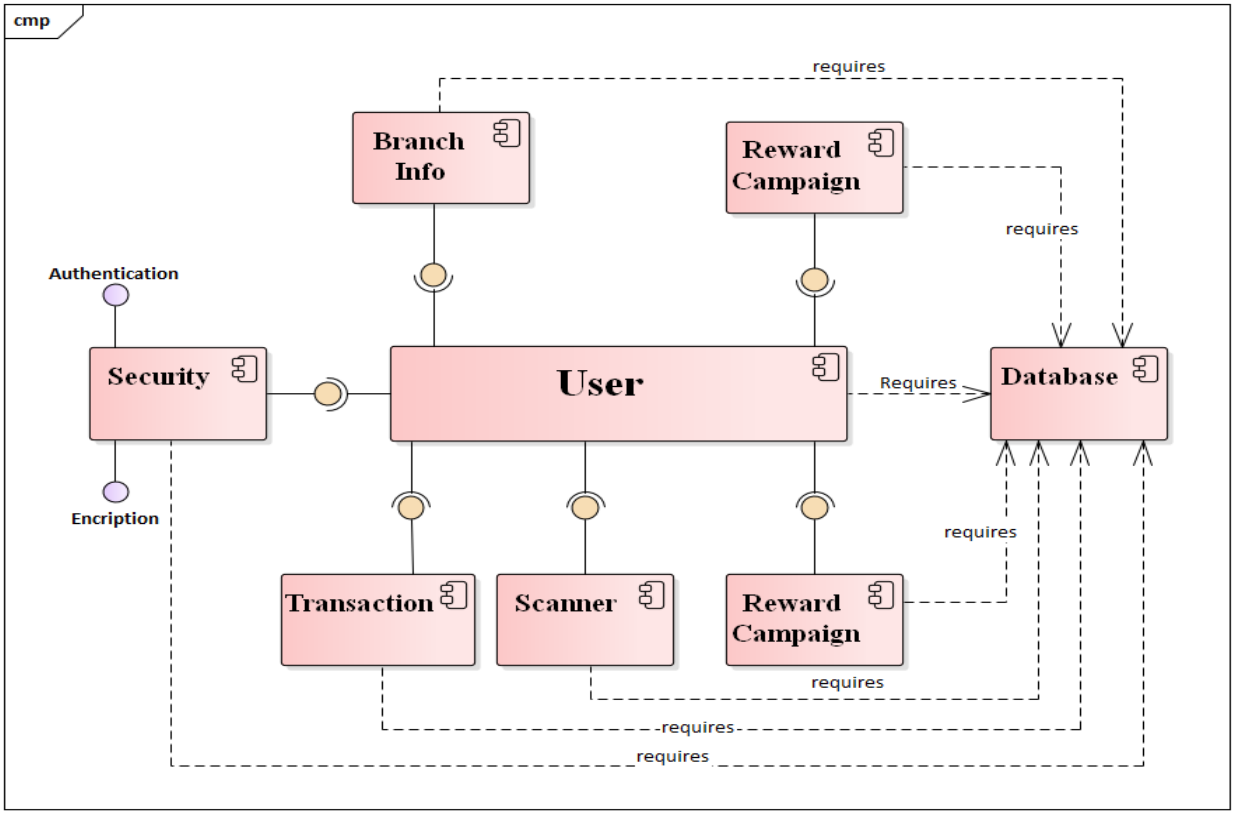


Figure 4 3: Component diagram and their interactions

4.7 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes, such as, hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. We planned to deploy our system as shown in figure 4.4.

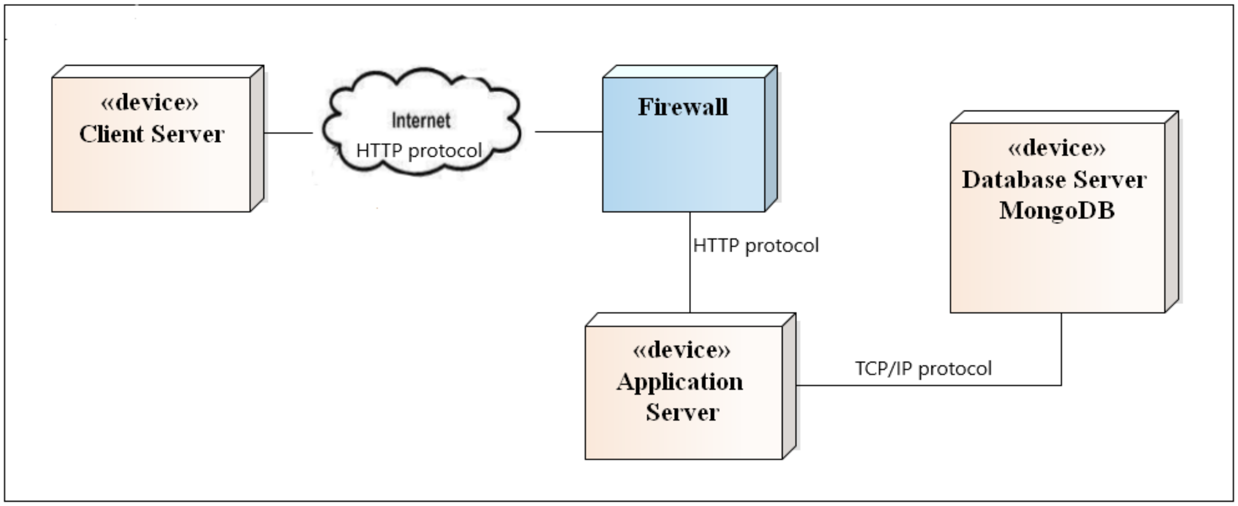


Figure 4 4: Deployment diagram

4.8 Database Design

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. It involves a process of identifying data and their relationships. The main objectives of database designing are to produce logical and physical design models of the proposed database system [4].

We have designed our system's database based on the database management systems (DBMS) [4] concepts to meet the user requirements and makes the system to have high performance.

4.8.1 Entity Relationship Diagram (ER Diagram)

ER diagram displays the relationships of entity sets stored in a database [4]. In other words, we can say that ER diagrams help us to explain the logical structure of the database. Our ER diagram is depicted as shown in figure 4.5.

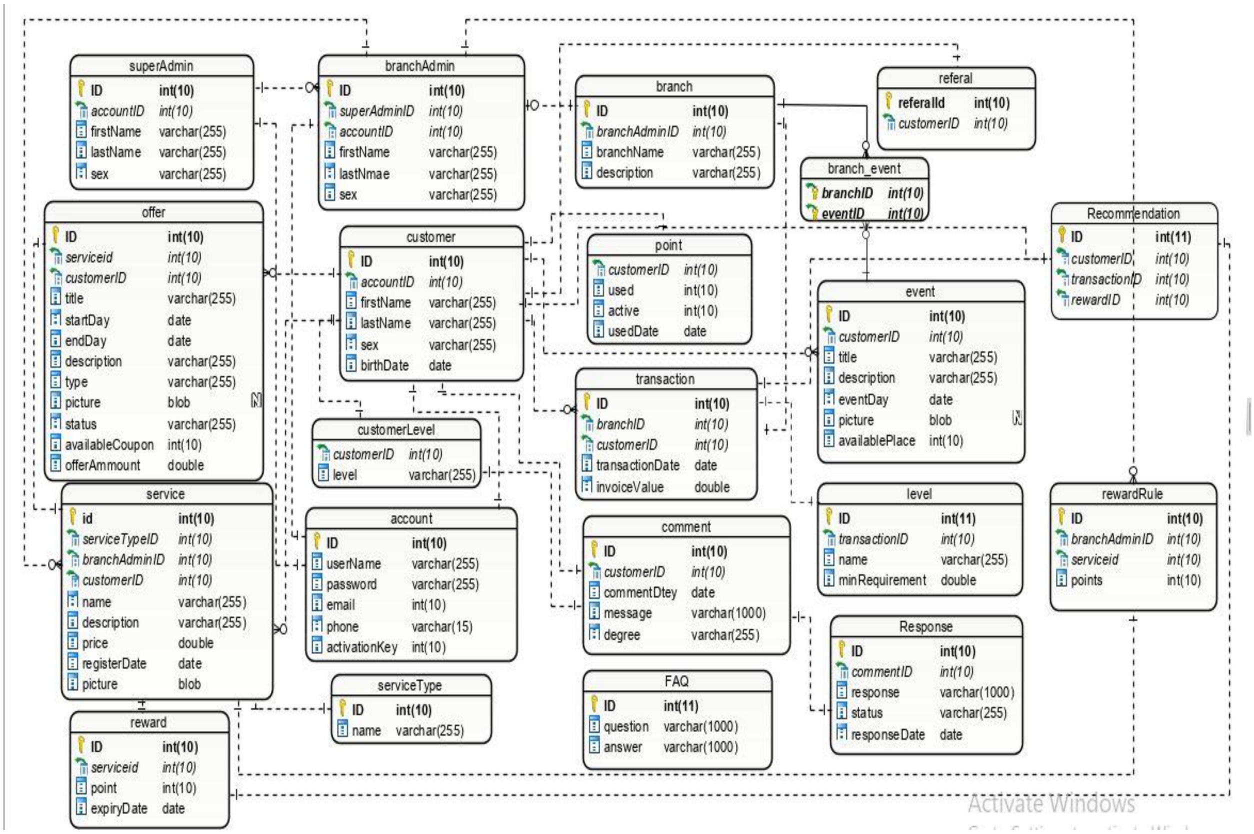
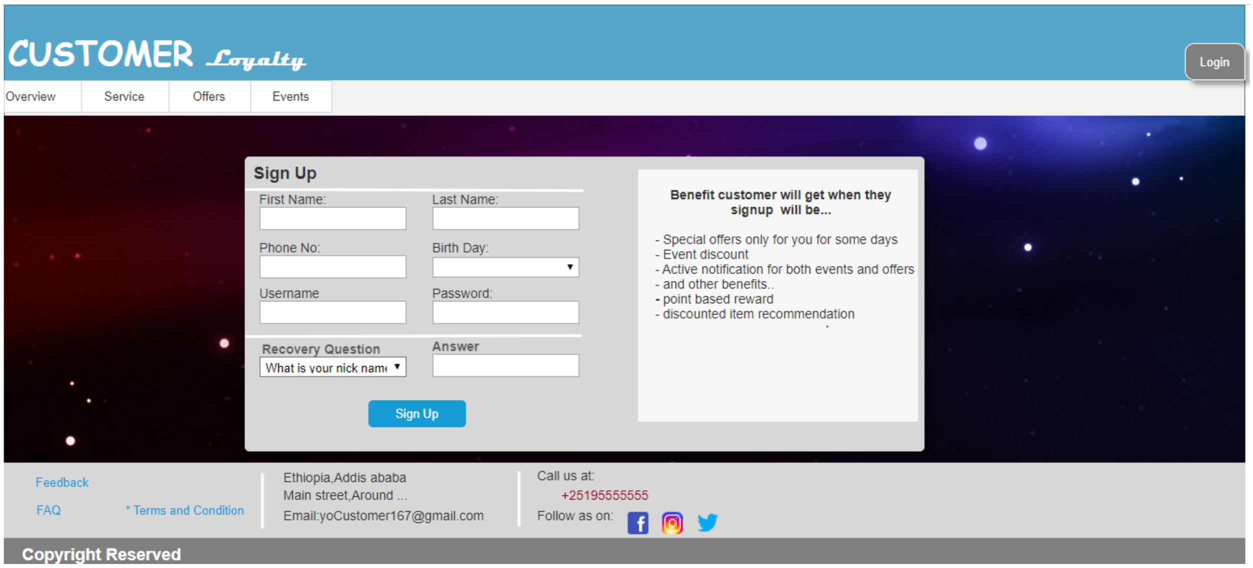


Figure 4 5 ER diagram

4.9 User Interface Design

User interface design (UI) generally refers to the visual layout of the elements that a user might interact within a website, or technological product. This could be the control of the visual layout of a webpage. User interface design can dramatically affect the usability and user experience of an application. We have designed our system's user interface simple and easy to use, it asks the user to insert a few inputs without any ambiguity. In this topic, we have included different user interfaces and these interfaces represent the upper layer of our system that is the presentation layer. See figure 4.6 – 4.12.

A user interface for customer registration – This interface allows customers to create an account. See figure 4 5.



Benefit customer will get when they signup will be...

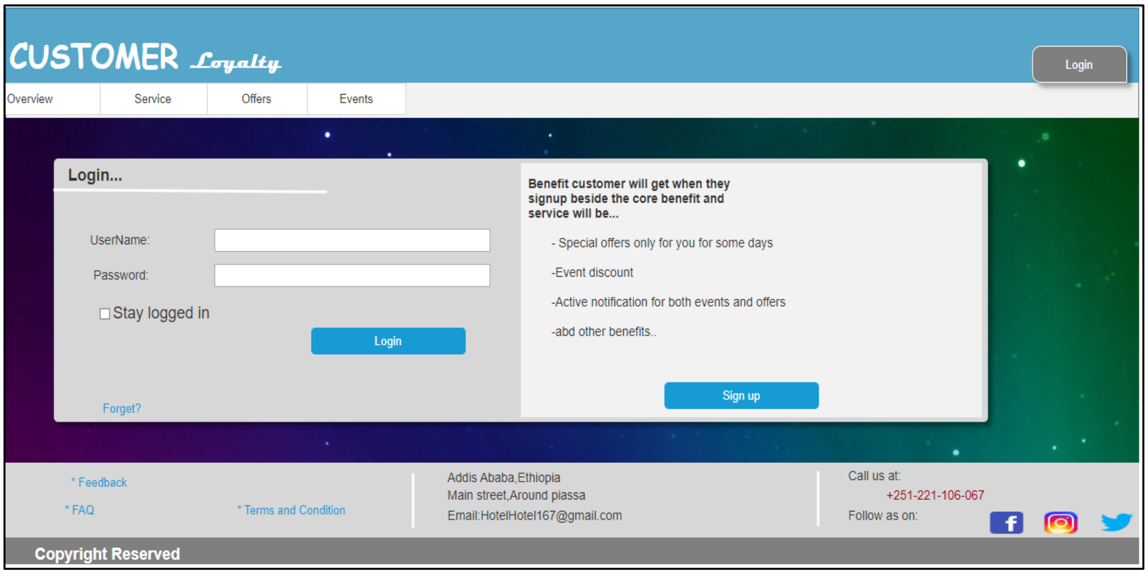
- Special offers only for you for some days
- Event discount
- Active notification for both events and offers
- and other benefits..
- point based reward
- discounted item recommendation

Ethiopia,Addis ababa
Main street,Around ...
Email:yoCustomer167@gmail.com

Call us at:
+25195555555

Figure 4 6: User interface design for customer registration

A user interfaces for Login – This interface allows customers to logged in to the system. See figure 4 7.



Benefit customer will get when they signup beside the core benefit and service will be...

- Special offers only for you for some days

-Event discount

-Active notification for both events and offers

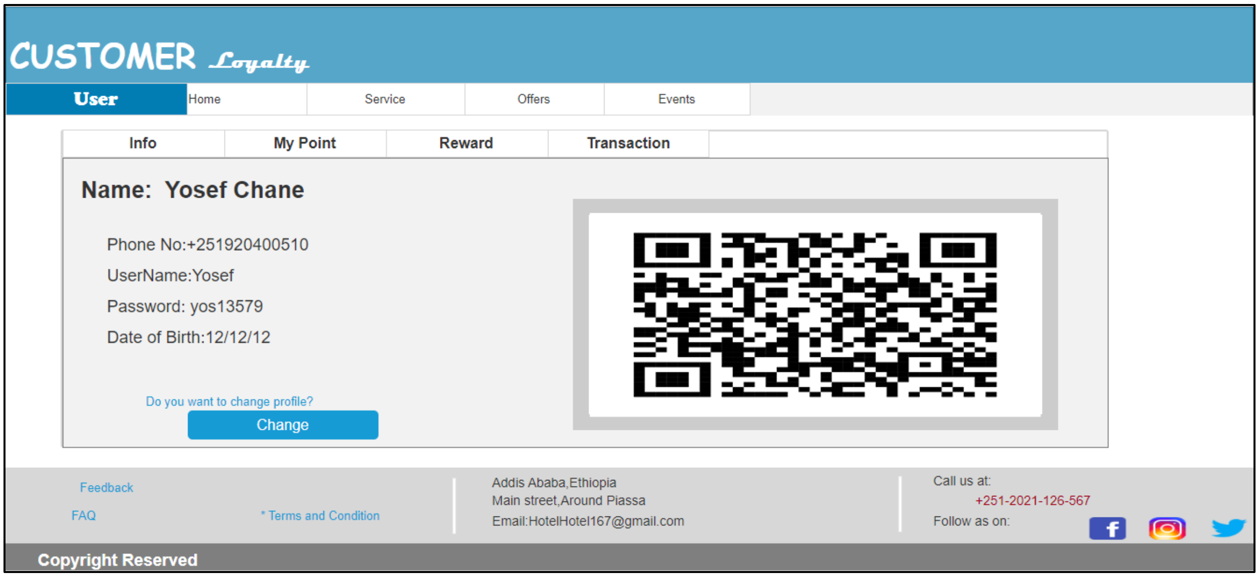
-abd other benefits..

Addis Ababa,Ethiopia
Main street,Around piassa
Email:HotelHotel167@gmail.com

Call us at:
+251-221-106-067

Figure 4 7: User interface design for login

User interfaces for customer Profile – This interface shows the customer profile information and allows them to update their profile information. See figure 4 8.



Name: Yosef Chane

Phone No:+251920400510

UserName:Yosef

Password: yos13579

Date of Birth:12/12/12

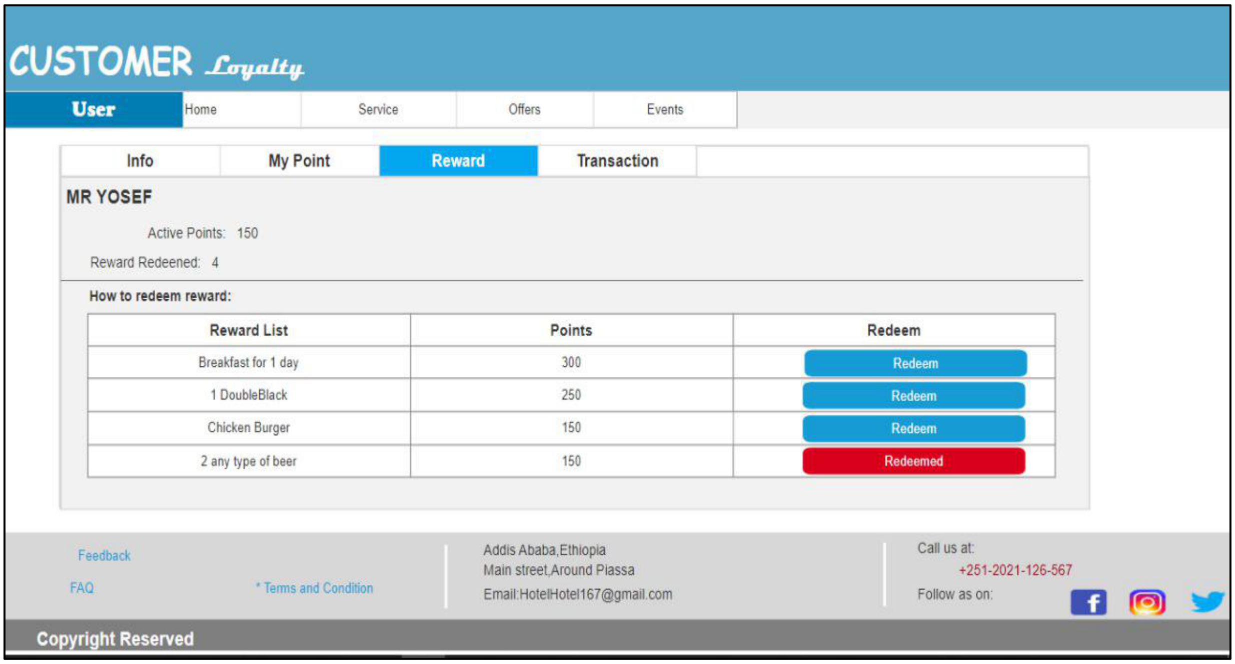
Do you want to change profile?

Addis Ababa,Ethiopia
Main street,Around Piassa
Email:HotelHotel167@gmail.com

Call us at:
+251-2021-126-567
Follow as on:

Figure 4 8: User interface design for customer personal information.

A user interface for reward redeem page – This interface allows customers to see the available rewards and their points. And also, it allows the customer to redeem rewards if they have enough points for the listed rewards. See figure 4 9.



MR YOSEF

Active Points: 150

Reward Redeened: 4

How to redeem reward:

Reward List	Points	Redeem
Breakfast for 1 day	300	Redeem
1 DoubleBlack	250	Redeem
Chicken Burger	150	Redeem
2 any type of beer	150	Redeemed

Addis Ababa,Ethiopia
Main street,Around Piassa
Email:HotelHotel167@gmail.com

Call us at:
+251-2021-126-567
Follow as on:

Figure 4 9: User interface design for reward redeeming

User Interface design for admin login

The above diagram is the gateway of the branch admin or super admin for accessing the content related to their authority. They will select their type first then for the super admin their username and password will be given for the first time and can change later using update profile but for the branch admin, their username and password will be sent through their phone number from the super admin. This is shown in figure 4 10.

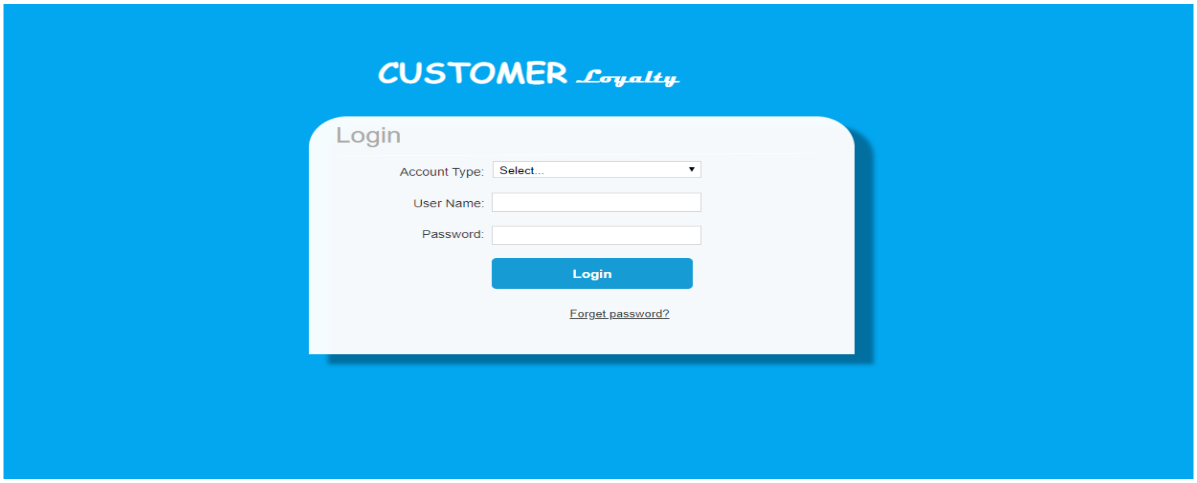
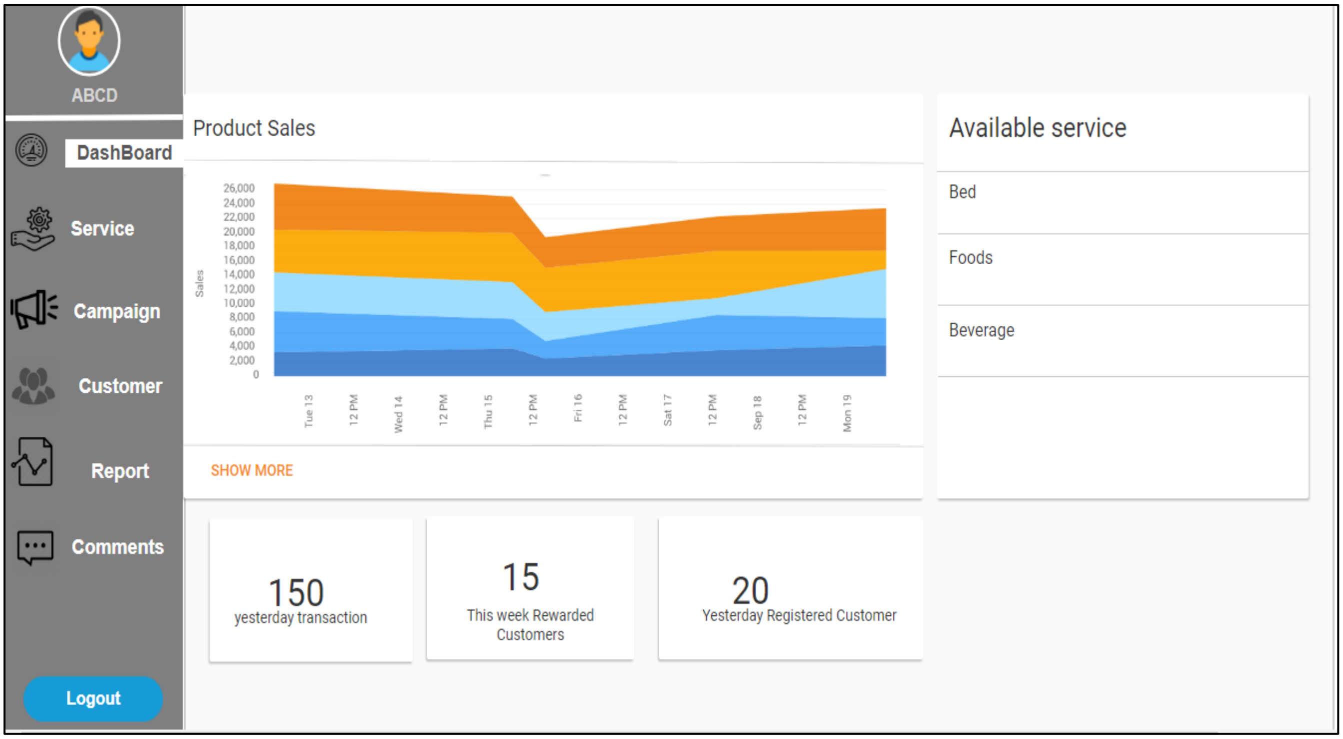


Figure 4 10: User interface design for administrative login

User interface design for admin dashboard

The above diagram is a part of the branch admins page and it is used to display the dashboard of many things especially something that is dome in the near past for example this week transaction, this week rewarded customer and others by simple overview as shown in figure 4 11.



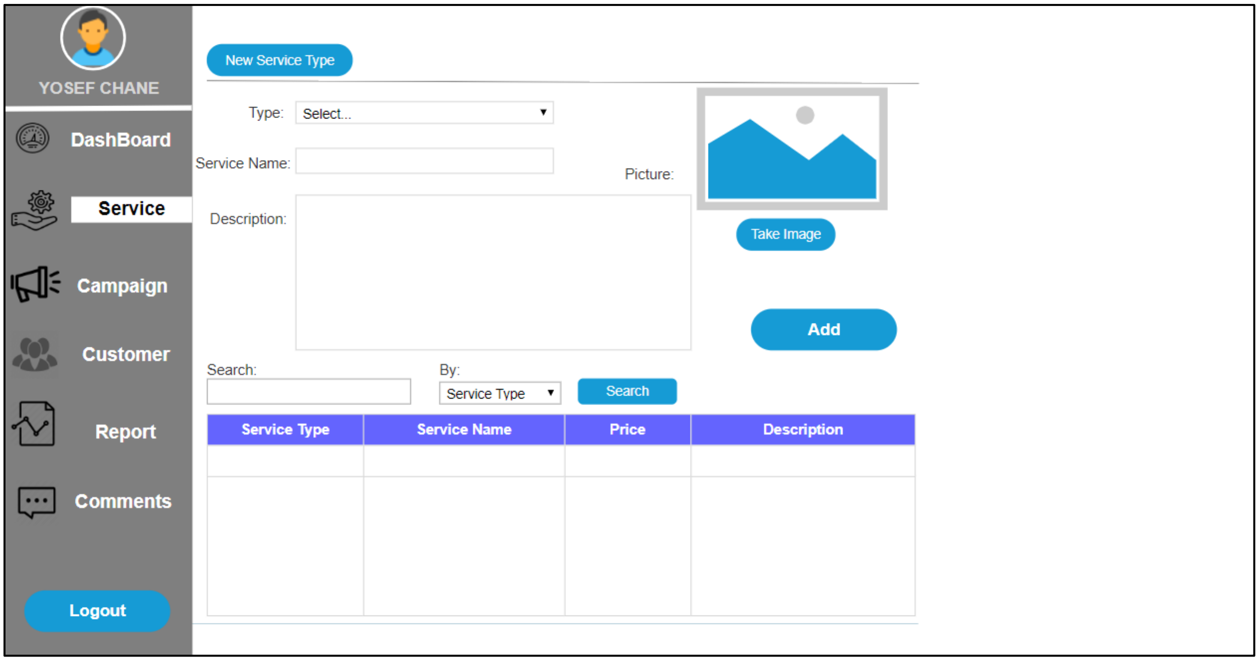
150
yesterday transaction

15
This week Rewarded Customers

20
Yesterday Registered Customer

Figure 4 11: User interface design for admin dashboard

A user interfaces for add service campaign – This interface allows the admin to add services for the customers. See figure 4 12 below.



Type: Select...

Service Name:

Description:

Picture:

Search:

By: Service Type

Service Type	Service Name	Price	Description

Figure 4 12: User interface design for services

Chapter 5

Implementation and Testing

5.1. Implementation

Implementation is one part of the software development cycle which means the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. Let us describe some of the schema designs for the loyalty program [17].

Back End implementation

Backend is server side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend [18].

- **Customer(client) schema**

This schema is the representation of the customers that participate in the customer loyalty program. It includes all the necessary information about the customer.

```
const mongoose = require("mongoose");
mongoose.pluralize(null);
var clientSchema = new mongoose.Schema({
  firstName: {
    type: String,
  },
  lastName: {
    type: String,
  },
  phoneNumber: {
    type: String,
    unique: true,
  },
  password: {
    type: String,
  },
  birthDate: {
    type: String,
  },
  isFemale: {
    type: Boolean,
  },
  points: {
    type: Number,
  },
  totalPoints: Number,
  level: String,
  registeredDate: Number,
  referredFrom: String,
  referralCode: {
    type: String,
    unique: true,
  },

  activationCode: String,
  profileImage: String,
  customerId: Number,
});
var clientSchema = mongoose.model("clientCollection", clientSchema);
module.exports = clientSchema;
```

- **Service schema**

This schema represents the data model of the service that are posted by the organization to the customers application.

```
const mongoose = require("mongoose");
mongoose.pluralize(null);

var serviceSchema = new mongoose.Schema({
  selectedServiceCatagory: {
    type: String,
  },
  selectedServiceSubCatagory: {
    type: String,
  },
  serviceName: {
    type: String,
  },
  servicePrice: {
    type: Number,
  },
  serviceDescription: {
    type: String,
  },
  serviceImage: {
    type: String,
  },
  branchName: String,
});
var serviceSchema = mongoose.model("serviceCollection", serviceSchema);
module.exports = serviceSchema;
```

- **<u>Event Schema</u>**

The event schema describes about the properties of the event that is going to be posted by admins as needed. It holds attributes that are needed for the event event to be posted.

```
const mongoose = require("mongoose");
mongoose.pluralize(null);

var eventSchema = new mongoose.Schema({
    eventTitle: {
        type: String,
        unique: true,
    },
    eventDescription: String,
    eventImage: String,
    eventDate: String,
    eventStartTime: String,
    selectedLevel: String,
    numberOfPeople: Number,
    eventBranch: String,
    like: Number,
    dislike: Number,
  share: Number,
});
var eventSchema = mongoose.model("eventCollection", eventSchema);
module.exports = eventSchema;
```

Recommendation: this implementation part represents the recommendation system which is applied by using association rule mining using the algorithm called apriori algorithm.

```
async getRecommondation(req, res) {
    try {
      await transactionConn.find((err, allTransactions) => {
        if (err) {
          return res.status(403).send({
            error: err,
          });
        } else if (allTransactions == null) {
          return res.status(404).send({
            error: "There is no transaction",
          });
        } else {
          var transactions = [[]];

          let i = 0;
          for (i = 0; i < allTransactions.length; i++) {
            if (i > 0) {
              var count = 0;
              for (let j = 0; j < transactions.length; j++) {
                for (let f = 0; f < transactions[j].length; f++) {
                  if (
                    allTransactions[i].transactionDate ==
                    transactions[j][f].transactionDate
                  ) {
                    transactions[j].push(allTransactions[i]);
                    count++;
                  }
                  break;
```

```
                    }
                    if (count > 0) break;
                }
                if (count == 0) {
                    transactions.push([]);
                    transactions[transactions.length - 1].push(allTransactions[i]);
                }
            } else {
                transactions[i].push(allTransactions[i]);
            }
        }

        var tempTransaction = transactions;
        transactions = [];
        for (let conuter = 0; conuter < tempTransaction.length; conuter++) {
            transactions.push([]);
        }

        for (let index = 0; index < tempTransaction.length; index++) {
            for (
                let rowIndx = 0;
                rowIndx < tempTransaction[index].length;
                rowIndx++
            ) {

transactions[index].push(tempTransaction[index][rowIndx].item);
            }
        }

        let apriori = new Apriori(0.25);

        // console.log(transactions1);
        apriori.on("data", (itemset) => {
            // Do something with the frequent itemset.
```

```
  let support = itemset.support;
  let items = itemset.items;
});

apriori.exec(transactions).then((result) => {
  let frequentItemsets = result.itemsets;

  let finalItemSet =
    frequentItemsets[frequentItemsets.length - 1].items;

  var finalRecommondation = [];

  for (let x = 0; x < finalItemSet.length; x++) {
    const reward = {
      rewardName: finalItemSet[x],
    };

    try {
      rewardConnection.findOne(reward, (err, reward) => {
        if (reward) {
          finalRecommondation.push(reward);
          if (x == finalItemSet.length - 1)
            res.send({
              allRewards: finalRecommondation,
            });
        }
      });
    } catch (err) {
      res.status(400).send({
        error: err,
      });
    }
  }
});
```

```
            }
          });
        } catch (err) {
          res.status(403).send({
            error: err,
          });
        }
    },
```

Front End implementation

The part of a website that user interacts with directly is termed as front end. It is also referred to as the ‘client side’ of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu. HTML, CSS, and Javascript are the languages used for Front End development. The structure, design, behavior, and content of everything seen on browser screen when websites, web applications, or mobile apps are opened up, is implemented by front End developers. Responsiveness and performance are two main objectives of the front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen [18].

AdminLoginPage.vue: this implementation represents the frontend admin login page.

```
 <template>
   <div>
      <div class="text-center mt-10" id="adminLoginForm">
         <div>
            <v-btn text dark id="helpBtn">
               <span class="text-capitalize">Help</span>
            </v-btn>
            <span class="display-2 grey--text mt-10">Customer Loyalty Program</span>
         </div>
        <v-form @submit.prevent enctype="multipart/form-data" class="white pa-10 pt-4 mt-10"
            id="loginPageLayout">
```

```
<v-layout row wrap>
    <v-flex xs12 md12>
        <span class="headline grey--text">Login</span>
        <div class="red--text" v-html="adminLoginError"></div>
        <span class="red--text" v-if="$store.state.isSessionExpiredIn"
            >Session expired. login again
        </span>
        <v-flex xs6>
            <v-select
                placeholder="Choose actor"
                v-model="selectedActor"
                :items="listOfActors"
            ></v-select>
        </v-flex>
    </v-flex>
    <v-flex xs10 md11>
        <v-text-field
            label="Username"
            prepend-icon="person"
            v-model="adminUserName"
        ></v-text-field>
    </v-flex>
    <v-flex xs10 md11>
        <v-text-field
            label="Password"
            :type="type"
            prepend-icon="lock"
```

```
     v-model="adminPassword"
   ></v-text-field>
 </v-flex>
 <v-tooltip top>
   <template v-slot:activator="{ on }">
     <a class="mt-6" v-on="on" @click="showHidePassword">
       <v-icon id="showHidePassword">visibility</v-icon>
     </a>
   </template>
   <span>{{ toolTip }}</span>
 </v-tooltip>
<v-checkbox
   class="ml-10"
   v-model="isStayLoggedInChecked"
 ></v-checkbox>
 <span class="mt-5">Stay logged in</span>
 <v-flex xs11 md11>
   <v-btn text dark class="primary pr-10 pl-10 mt-5"
     @click="adminLogin"
     :loading="loading"
     type="submit" >
     <span class="text-capitalize">Login</span>
   </v-btn>
 </v-flex>
 <v-flex xs11 md11>
   <v-btn text class="white mt-10 pl-10 pr-10">
     <a class="text-capitalize">Forgot password</a>
```

```
                    </v-btn>
                </v-flex>
            </v-layout>
        </v-form>
    </div>
</div>
</template>
```

5.2. Testing

This test plan describes the testing approach and overall framework that will drive the testing of the Customer Loyalty Program specifically developed for hotels. The objective of the test is to verify that the functionality of the customer Loyalty program works according to the specifications. The test will execute and verify the test scripts, identify, fix, and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing [16].

5.2.1. Scope

This document will describe the approach, resources, and schedule of the testing activities possible at this time. This document will also identify the particulars of testing (what to test, what not to test), the tasks that need to be performed, the method of testing. At this time, we are mainly focused on testing the functional requirements of the system[16].

5.2.2. Test Items

This section of the Test Plan deals with the testing of all those items that constitute the hotel loyalty program. The following is the list to be tested:

Functional Testing: Functional testing will be performed to check the functions of the application. The functional testing is carried out by feeding the input and validates the output from the application [16]. The function of the loyalty program to be tested includes:

- Signup process
- Log in process
- Registration process
- Updating information

- Redeem rewards
- Delete information
- Give feedback
- Searching and sorting
- Generating reports
- Scanning bar code
- Recommendation
- Share

5.2.3. Test Cases.

- Registration process

Test ID	Test Case	User input	Expected outcome	Actual Output	Pass / Fail
1	User registration	A user enters already existing user name	Display “user name already exists”	As expected,	pass
2	User registration	Enter a different password in the password confirm field	Display “Password and Confirm Password fields don't match”	As expected	pass
3	User registration	User forgets a particular required field	Display “the value in a field is required”	As expected	pass
4	User registration	A user enters all the details successfully	The user account created successfully	As expected	pass

- Login process

Test ID	Test Case	User input	Expected outcome	Actual result	Pass / Fail
1	User login	A user enters the incorrect user name or password	Display “user or password incorrect”	As expected	pass

2	User Login	Enter Correct username and password	User Login successfully	As expected	pass

- Offer registration process

Test ID	Test Case	User input	Expected outcome	Actual result	Pass / Fail
1	Offer registration	A user enters an already existing offer	Display "Offer already exists"	As expected	Pass
2	Offer registration	User select inappropriate date for the offer (before current date)	Display "please select a valid date"	As expected	Pass
3	Offer registration	The user forgot to fill in a particular required field	Display "the value in the field is required"	As expected	Pass
4	Offer registration	User enter all values correctly	Offer registered successfully	As expected	Pass

- Event registration process

Test ID	Test Case	User input	Expected outcome	Actual result	Pass / fail
1	Event registration	The user enters an already existing event	Display "Event already exists"	As expected	Pass
2	Event registration	User select inappropriate date for the Event (before current date)	Display "please select the valid date"	As expected	Pass
3	Event registration	The user forgot to fill in a particular required field	Display "the value in the field is required"	As expected	Pass

4	Event registration	User enter all values correctly	Event registered successfully	As expected	Pass

- Updating information

Test ID	Test Case	User input	Expected outcome	Actual result	Pass /fail
1	Updating information	User press update button without changing the value	Display “there is no update done”	As expected	Pass
2	Updating information	Enter invalid data on the field when updating	Display “Please enter valid data”	As expected	Pass
3	Updating information	User forget particular required field when updating	Display “the value in a field is required”	As expected	Pass
4	Updating information	A user enters all the data to update correctly	Information updated successfully	As expected	Pass

- Redeeming reward

Test ID	Test Case	User input	Expected outcome	Actual result	Pass/fail
1	Redeem reward	User tries to redeem without enough point	Display “you have insufficient points to redeem”	As expected	Pass
2	Redeem reward	Press the button without selecting a specific service or product	Display “first select the product to redeem”	As expected	Pass
3	Redeem reward	user select a product with enough point and redeem	You redeemed your reward successfully.	As expected	Pass

- Searching process

Test ID	Test Case	User input	Expected Outcome	Actual result	Pass/ fail
1	Searching	Click on the search button without entering the key to search	Display "first enter the key you want to search for"	As expected	Pass
2	Searching	Enter key that doesn't much to any of the fields	Display "no product or service found with the key you entered"	As expected	Pass
3	searching	User enter key which matches	Display products or services which contain the keyword	As expected	Pass

- Sorting process

Test ID	Test Case	User input	Expected Outcome	Actual result	Pass/ fail
1	Sorting	User click on sort without selecting any parameter	Display the list by the default sorting	As expected	Pass
2	Sorting	User click sort after selecting some parameter	Display the list based on the selected parameter	As expected	Pass

- Scanning barcode

Test ID	Test Case	User input	Expected Outcome	Actual result	Pass/ fail
1	Barcode Scan	A user scans an unknown barcode	Displays "unknown barcode"	As expected	Pass
2	Barcode scan	Scan the barcode with the scanner	Display the specific customer profile	As expected	Pass

- Recommendation process

Test ID	Test Case	Input	Expected Outcome	Actual result	Pass/ fail
1	Recommendation	The system traverses the throughout the customer transaction by the using algorithm	The recommend service or product displayed on the customer application if the minimum requirement is fulfilled.	As expected	Pass

- Give Feedback process

Test ID	Test Case	User input	Expected Outcome	Actual result	Pass/ fail
1	Give Feedback	A user tries to give feed back with filling all required fields.	Displays “please fill required fields”	As expected	Pass
2	Give Feedback	A user fill all required fields and press give feedback button	the feedback sent to the admin	As expected	Pass

- Delete information

Test ID	Test Case	User input	Expected Outcome	Actual result	Pass/ fail
1	Delete information	User clicks on Delete without selecting any information	Display “please select a you want to delete first”	As expected	Pass
2	Delete information	User click Delete after selecting single item	The system alerts the user are you sure you want to delete? If the user click yes	As expected	Pass

			it deletes the selected item, unless it doesn’t delete		
3	Delete information	User click Delete after selecting more than one item	The system alerts the user are you sure you want to delete all selected items? If the user click yes it deletes the selected item, unless it doesn’t delete	As expected	Pass

- Generating Report

Test ID	Test Case	User input	Expected Outcome	Actual result	Pass/ fail
1	Generate report	The user click export button Without changing the filtering parameters.	The system displays the default list and let the user to select the format to be generated? after the user select the one it will generate the report.	As expected	Pass
2	Generate report	The user click export button after changing the filtering parameters.	The system display the list based on the selected filtering parameters and let the user to select the format to be generated? after the user select the one it will generate the report	As expected	Pass

- Share

Test ID	Test Case	User input	Expected Outcome	Actual result	Pass/ fail
1	Share	A user tries to share service or product to social media.	The system let the user to select a social media to share in. and the user select the media and share the service or product.	As expected	Pass

5.2.4. Testing Approaches

This application is tested with the test case input and its correct output. For example, to test the login functionality of the web application, the tester will enter the user login name and its password. The expected output is the user is logged into the system. The approaches to testing include:

- Unit testing
- Integration testing
- System testing

5.2.5. Item Pass/Fail Criteria

Test cases executed on the application will pass if they meet the specific requirements as mentioned in the Requirement Document. The item is said to fail the system if the results that the test case produces does not match the desired result or the expected outcomes. Test cases are evaluated independently of one another, not affect the evaluation of the other test cases[16].

Chapter 6

Conclusion and future recommendation

6.1. Conclusion

A **customer loyalty program** is a program run by a company that offers benefits to frequent customers. Those benefits may be in the form of discounts, rebates, free products, or other promotions. An effective customer loyalty program rewards customer who buys from a business on a regular basis, encouraging the customer to return frequently. Loyalty programs have several goals: increase business, improve sales, strengthen the relationship between the customer and the business, and keep the customer coming back on a regular basis. One important element of meeting these goals is ensuring the program has a benefit to the customer. So, for organizations having this system very important to boost up their business [15].

6.2. Future work

At this time, the loyalty program we develop is by considering only hotels. In the future we are going to expand the system in different field of business organizations like supermarkets, barber shops, bus transportation, and other businesses.

When we develop this system, we were not including the online payment system because of the current status of our country’s technology. If this problem will be solved, our system going to include many organizations from different business areas as a platform and allow customers to get service from different organization using single application.

References

[1] https://blog.smile.io/a-history-of-loyalty-programs (History of customer loyalty Program) -- last visited on Nov 12, 2019.

[2] https://test-doc123.readthedocs.io/en/latest/userguide/reward_campaigns -- (about reward campaign)

[3] https://medium.com/the-andela-way/system-design-in-software-development (system design) -- last visited on Dec 18, 2019.

[4] https://www.guru99.com/er-diagram-tutorial-dbms.html – (ER Diagram) -- last visited on Dec 20, 2019.

[5] Martina Seidl • Marion Scholz Christian Huemer • Gerti Kappel UML @ Classroom An Introduction to Object-Oriented Modeling --- for UML diagram

[6] https://nodejs.org/en/docs/ Nodejs documentation -- last visited on Nov 25, 2019.

[7] https://expressjs.com/en/5x/api.html Expressjs documentation-- last visited on Nov 25, 2019.

[8] https://docs.mongodb.com/ mongoDB documentation-- last visited on Nov 25, 2019.

[9] https://flutter.dev/docs Flutter documentation-- last visited on Nov 25, 2019.

[10] https://vuejs.org/v2/guide/ vuejs documentation-- last visited on Nov 25, 2019.

[11] https://vuetifyjs.com/en/getting-started/quick-start Vuetify documentation-- last visited on Nov 25, 2019.

[12] https://www.investopedia.com/terms/d/datamining.asp Data mining-- last visited on Jan 10, 2020.

[13] Product Recommendation based on Shared Customer's Behavior by “Fatima Rodrigues, Bruno Ferreira” Association rule data mining.

[14] Benjamin L Kovitz: “A Manual of Content and Style” Practical Software Requirements

[15] https://study.com/academy/lesson/customer-loyalty-program-definition-purpose-example.html

[16] Live project test plan_softwareTestign_help.doc

[17] https://en.wikipedia.org/wiki/Implementation

[18] https://www.geeksforgeeks.org/frontend-vs-backend/

Appendix I

Interview questions

The following are the interview questions we used to gather information from some hotels about customer loyalty services.

1. How do you attract new customers to your hotel, is there any special method?
2. How do you keep track of your customers?
3. How do you know if the customer is satisfied with the service or not?
4. Is there any way of collecting the customers' feedback orally or with a form?
5. What do you do to retain a customer for the long run?
6. How do you know if a customer is loyal or how do you win over customers?
7. How does the customer know the services provided before arriving at the hotel?
8. How do you announce when there is an event in the hotel?
9. How do you advertise your hotel, how do customers know the services you provide before getting to the hotel, any online platform or social media advertisement?
10. How do you predict your most usable and satisfactory services?
11. What methods do you use to let the customers spend more of their time and money in the hotel?
12. How do you know if the customers stop using the hotel?
13. How do you measure your success rate, is there any visual way of determining the success rate?

Questionnaires

1. When is your company established? During the firsts weeks or months, how many customers visit the hotel?

____

____

2. How many branches, employees and customers do you have?

____

____

4. Do you have customer loyalty services in your hotel?

Yes □ No □

If your answer is yes, how does the system work?

____

____

____

____

____

If your answer is no, haven’t you considered having a customer loyalty service?

Yes □ No □

5. Is there any way of giving a discount or gift to customers?

Yes □ No □

If your answer is yes, how do you screen the customers that deserve the gift or the discount?

____

____

____

____

____.

6. Is there any formal system for advertising your services?

Yes □ No □