Flamingo- A Smart Guide

By

Amna Ijaz

Munazza Bano

Amna Batool

Zahra Qamar

Submitted to the Faculty of Computer Software Engineering

National University of Sciences and Technology, Islamabad in partial fulfillment

For the requirements of a B.E. Degree in Computer Software Engineering

JUNE 2021

ABSTRACT

Flamingo-A Smart Guide

The Application will allow the users to sign up, login, draw a hologram and anchor it to a specific location of significance in the world. They can also post a video of their surroundings which is connected to that anchor. The user will get recommendations based on previous visited locations. The user will also get an option to create a customized avatar, chat, view personal profile, view avatar map, newsfeed and sign out.

Flamingo is a new self-contained product which has the following major components AI based recommendation system, AR and location-based rendering system, Database and User interaction system.

The application will be built using Android studio, AR Core, Spatial Anchors for Smart phones with OS Android 8.0 or later. Firebase, and Spatial Anchors is used for storing the holograms, location and the application and user data.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis – Flamingo - A Smart Guide carried out by Amna Ijaz, Munazza Bano, Syeda Amna Batool and Zahra Qamar in supervision of Asst. Prof Bilal Rauf for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Asst. Prof Bilal Rauf

Department of CSE, MCS

DATED:

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and cooperation, a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

We would like to thank Allah Almighty for His incessant blessings which have been bestowed upon us. Whatever we have achieved, we owe it to Him, in totality. We are also thankful to our families for their continuous moral support which makes us what we are.
We are extremely grateful to our project supervisor Asst. Prof Bilal Rauf from MCS who in addition to providing valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.
We are highly thankful to all of our teachers and staff of MCS who supported and guided us throughout our course work. Their knowledge, guidance and training enabled us to carry out this whole work.
Finally, we are grateful to the faculty of Computer Software Department of the Military College of Signals, NUST.
In the end we would like to acknowledge the support provided by all our friends, colleagues, and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

TABLE OF CONTENTS

CHAPTER: 1

INTRODUCTION

1.1. Overview

Direct communication and knowledge of firsthand experience lack in many cases. When in a new place most people get to face many barriers due to language and lack of knowledge about the experiences of previous tourists. In case of blood arrangements, most people always know someone or would have experienced the late supply of blood or lack of knowledge about blood donors which nearly costs them lives of their loved ones. While opting for school most people face difficulty specially of special children so that their child does not face any difficulty in education or with the environment.

Google AR Core platform gives developers an SDK to build Augmented Reality applications so that the user can get an occlusion effect with correct light estimation and creating an immersion effect which makes user feel like the virtual objects are really placed in the world. Augmented Reality increases the level of interactivity with user which is more appealing than other social media applications.

Flamingo is a social media application that uses AR to connects people around the world. People interact by anchoring the holograms and attaching a review or posting a video with it. Be it reviews for a restaurant, theaters, cinemas, or a blood donor at a specific blood bank all you need to do is scan your phones' camera to look for holograms or see on maps while sitting at home and you can connect to anyone with one single tap. Even while scrolling your newsfeed if you come across an intriguing post user can check the place it is anchored to and get the distance and directions to that place.

1.2. Problem Statement

Direct interaction, communication along with sharing of firsthand experiences has been an issue in the society for a long time. When in a new place most people lack knowledge about the experiences of previous tourists. In case of blood arrangements, most people always know someone or would have experienced the late supply of blood or lack of knowledge about blood donors which nearly cost them lives of their loved ones. While opting for school most people face difficulty specially of special children so that their child does not face any difficulty in education or with the environment. Presently, during corona people find it hard to locate the donors of the plasma.

1.3. Approach

To counter these, the approach is to create an augmented reality application that will connect users in a unique way. Using Flamingo, the users can arrange blood or get reviews about a place with a single tap. If they're in the vicinity of the other user, they can interact using hologram, otherwise avatar map can be used to connect with users from the comfort of their home.

1.4. Scope

- Flamingo will use AR Core. Holograms will be created by user when at a specific place using technology of AR Core and Android Studio.
- User will view holograms and interact with other users via chat.
- If user is at home, then avatars will appear as markers on a map.
- Directions to desired location will be provided to the user.
- Recommendations will be made based on interests and needs.

- Expected functionality to be achieved by summer semester 2021.

1.5. Aim & Objectives

The objectives of project include:

1. Using software engineering techniques for gathering requirements during the development process, designing the software, implementing, and testing requirements gathered
2. This project will help us learn Augmented Reality and Azure Spatial Anchors
3. Learn app development.
4. Learn firebase and database design and development.
5. This project will help us in exploring domains and getting insights on the trending technologies of the future.

1.6. Organization

The first part of thesis is the abstract which describes the main details of Flamingo, followed by the introduction section which specifies the problem statement, approach, scope and objectives. The literature review section states the various resources read online before the commencement of the project. They include learning about all diseases and the need of physiotherapy. The design and development part illustrate the diagrams which describe the detailed design of Flamingo, its components, interfaces and data necessary for the implementation phase. The analysis and evaluation part give details of the black box testing, unit testing and system integration testing, actual results against expected results. The future work gives states the enhancements that can be applied to the application.

1.7. Deliverables

Table 2. 1: Deliverables

Deliverable Name	**Deliverable Summary Description**
Software Requirements Specification (SRS) Document	Complete Description of **what** the system will do, who will use it. Detailed description of functional and non-functional requirements and the system features.
Software Design Document	Complete description of how the system will be implemented i.e., the detailed design.
Code	Complete code with the API.
Testing Document	The whole system is tested according to Black box, unit, and System integration testing is done.
Complete System	Complete working system.

CHAPTER: 2

LITERATURE REVIEW

2. Literature Review

The goal of this chapter is to first highlight the importance of existing similar systems and then add a unique feature to it. In our case, there are social media applications that can be considered a little like Flamingo as discussed later. However, our system proposes a solution that is not only unique but also very interactive. After careful analysis of the problems at hand, we created a system that is not just socially connecting users but also addressing social issues.

2.1. Related Work

There are social media applications like Facebook, Instagram, and Snapchat. There is only one similar application that makes use of Augmented Reality that is Snapchat.

2.1.1. Facebook

Facebook is a social media application that allows the users to post text, photos and multimedia which are shared with any other users who have agreed to be their "friend" or, with different privacy settings, publicly. Users can also communicate directly with each other with Facebook Messenger.

2.1.2. Instagram

Instagram provides messaging features, the ability to include multiple images or videos in a single post, and a 'stories' feature, which allows users to post photos and videos to a sequential feed, with each post accessible by others for 24 hours each.

2.1.3. Snapchat

Snapchat is an American multimedia messaging app developed by Snap Inc., originally Snapchat

Inc. One of the principal features of Snapchat is that pictures and messages are usually only available for a short time before they become inaccessible to their recipients. It also employs Augmented reality to create engaging filters for users.

2.2. Proposed System

The application Flamingo-A smart guide connects people around the world even if they are at home and aims to make or daily tasks easier by connecting people around the world. Since the rapid growth of the AR industry, our project will be one of its kind in Pakistan where this field is relatively new. Be at any place in the world with internet connectivity, just create your favorite hologram and take a short video and anchor to that specific location of significance. Well, that is how you connect to people by interacting with the holograms left behind by them. Be it reviews for a restaurant, theaters, cinemas or a blood donor at a specific blood bank all you need to do is scan your phones' camera to look for holograms or see avatars on maps while sitting at home and you can connect to anyone with one single tap. Even while scrolling your newsfeed if you come across an intriguing post user can check the place it is anchored to and get the distance and directions to that place.

Traditionally, the social media applications do not involve the use of the Augmented reality. Snapchat makes use of augmented reality, but it does not allow users to directly interact with each other using Augmented Reality.
We proposed the creation of Flamingo-A Smart Guide application to make blood arrangement easy so that blood is arranged in time without any loss, to help parents opt the best school for their children, to help tourists find the best places to visit and to help the world find plasma for corona virus using Augmented Reality and all this is just a one tap away.

The user signs up into the application then he/she creates the avatar. The user views the newsfeed where he/she can view the posts and reviews of other people. In the personal profile the user can select the create hologram button and anchor the hologram to specific location of interest. User can also scan a specific location to view holograms anchored by others and upon click he/she can chat with the owners of the hologram.

CHAPTER: 3

OVERALL DESCRIPTION

3. OVERALL DESCRIPTION

This part of the document contains information about the product, its features, perspective, users' characteristics, and constraints.

3.1. Introduction

3.1.1. Purpose

The purpose of this chapter is to give the user a clear and precise description of the functionality of "Flamingo- a smart guide'.

This chapter is aimed to eliminate ambiguities and misunderstandings that may exist. For the user,

this chapter will explain all functions that the software should perform. For the developer, it will be a reference point during software design, implementation, and maintenance.

Since the rapid growth of the AR industry, our project will be one of its kind in Pakistan where this

field is relatively new. Be at any place in the world with internet connectivity, just draw your hologram with taking a short video and anchor to that specific location of significance. That is how

you connect to people by interacting with the holograms left behind by them. Be it reviews for a restaurant, school or a blood donor at a specific blood bank all you need to do is scan your phones'

camera to look for holograms or see on maps while sitting at home and you can connect to anyone.

with one single tap. Even while scrolling your newsfeed if you come across an intriguing post user
can check the place it is anchored to get the distance and directions to that place.

3.1.2. Intended audience

The intended audiences for the Flamingo- A Smart Guide include the project supervisor, the BESE 23 FYP developers, the UG FYP evaluation board, and other individuals at MCS CSE Department.

3.1.2.1. Project supervisor:

It will help the supervisor to supervise the project and guide the team in a better way. This document will be used by him to check whether all the requirements have been understood and, in the end, whether the requirements have been properly implemented or not.

3.1.2.2. BESE 23 FYP Developers:

Project developers have an advantage of quickly understanding the methodology adopted and personalizing the product.

3.1.2.3. Testers:

The testers of the system can check user requirements from this SRS and develop the test document accordingly.

3.1.2.4. UG Project Evaluation team:

The evaluation board can check user requirements from this SRS and develop the test document accordingly.

3.1.3. Reading suggestions

All level 1 and level 2 headings are given in the table of contents, but the lower sub headings are not included. Each main heading is succeeded by a number of sub headings, which are all in bold format. The product overview is given at the start, succeeded by the complete detailed features,

including both functional and non-functional requirements. The entire interfaces are also described. The chapter ends with appendices, including a glossary.

3.1.4. Product Scope

Table 3. 1 Scope

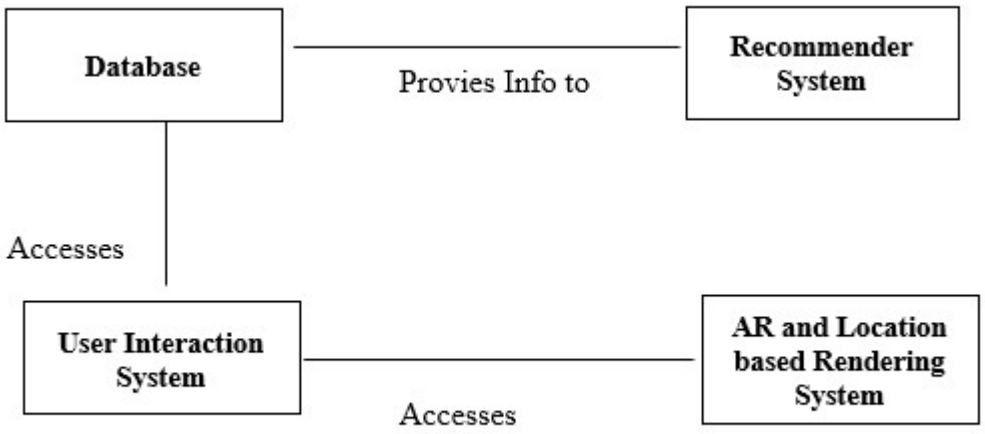
For	People who wish get or receive blood donation, tourist exploring restaurants and important places, getting reviews from students of a school, leaving covid information and any other information that can be of help.
What	A system consisting of a registered user, that anchors and shows holograms and reviews attached to them, lets users chat and make posts for newsfeed.
The	Flamingo- a smart guide
Is	An Android mobile application
That	Creates ease for people in fulfilling their important needs by viewing holograms of other people and connecting to them.

3.2. Overall Description

3.2.1. Product Perspective

Flamingo is a new self-contained product which has the following major components.

1) A recommendation system
2) AR and location-based rendering system
3) Database
4) User interaction system


Database
Provies Info to
Recommender System
Accesses
User Interaction System
Accesses
AR and Location based Rendering System

3.2.2. Product Functions

1. Sign Up

   User can create an account by adding their personal information.
2. Login:

   once the account is created the user can login to their account anytime
3. Update credentials:

   User can change their password as desired.
4. Add Review:

   User can add review for a particular place where they have left their hologram.
5. Create Hologram/Avatar:

   User can create hologram of their choice.
6. Newsfeed:

   System can display recommendations for places based on user's previously visited locations.
7. Chat:

   User can interact with another person by messaging them.
8. Avatar Map:

   User can look at avatars anchored on different locations through the avatar map.
9. Scan Location

   User can view holograms anchored at a location by scanning their surroundings.
10. Recommender:

    System can sort recommendations based on users' interests.
11. Sign out:

    User can end their session by signing out

3.2.3. User Classes and Characteristics

There are explanations of the user followed by the interactions the user(s) shall be able to make.

3.2.3.1. Blood Donor

For blood donor who wants to donate the blood will leave behind his hologram so that the people who need blood can interact them with hologram.

3.2.3.2. The Tourist

Tourists can get recommendations about the restaurants by interacting with the holograms in that restaurant as well as can know the distance to the specific restaurant.

3.2.3.3. Use By Common People

People can also use this smart guide for their ease for example for educational purpose they can get recommendations about a school through interaction with the holograms
A person can also interact with the holograms inside the cinema to get recommendations about the movies.

3.2.4. Operating Environment

3.2.4.1. Hardware

Flamingo operates, either directly or indirectly, with the following external hardware:

1. Smart phones (accelerometer, magnetometer, gyroscope, GPS, camera), with OS Android 8.0 or later.
2. GPU texture resolution 1080p, 720p, 480p.

3.2.4.2. Software

1. Android Studio.
2. AR Core (Augmented reality library).
3. Sceneform (Graphics Library).
4. Azure Spatial anchors.

3.2.4.3. Database

The database used is Firebase.

3.2.5. Design and Implementation Constraints

Operating system requirement:

The operating system to support AR Core enable applications should not be less Android 8.0 version.

3.2.6. User Documentation

- User manual
- Thesis
- Video explaining the functionality of application

The distribution of documentation is strictly going to be in college premises since it is an academic project.

3.2.7. Assumptions and Dependencies

- **Android Studio:**
  We are assuming that all the functionalities will be covered using pre-defined libraries and plugins in Android Studio.
- **Google Maps:**
  Google Location Services will be required in our application to anchor the holograms and to keep track of them.
- **Internet connection:**
  We need to have stable internet connection.
- **Camera:**
  The user needs to give camera access to application.

3.3. External Interface Requirements

3.3.1. User Interfaces

The requirements for user interfaces would be.

1. The interface shall be user friendly and very simple to use.

2. The buttons shall be big to make their selection easy for it might be hard for a user to select.

3. The interface of application shall be in tones of pastel wit dark font color because it makes focusing easier.

4. Each screen will be explanatory regarding the options and functionality provided by the system.

3.3.2. Hardware Interfaces

Hardware Interfaces For our project includes,

- Smartphone with Camera, gyroscope, accelerometer, GPS, and magnetometer.
- Communication protocol: TCP

3.3.3. Software Interfaces

Software used	**Description**
Operating system	We have chosen Android version 8.0 and above for its best support, user-friendliness, and compatibility with AR Core.
Database	To save the data in the form of images, text etc., we have chosen Firebase database.
Tools and Libraries	Android Studio was chosen because it has structured code modules which divide project into units of functionality that you can independently build, test, and debug. AR Core and Sceneform are used since they have motion tracking, which lets phones understand and track their positions relative to the world.

3.3.4. Communications Interfaces

This application uses HTTP and GPRS protocols. We are using simple electronic forms for the signup, sign in etc.

3.4. System Features

3.4.1. Sign Up

3.4.1.1. Description

User can create an account by adding their personal information.

3.4.1.2. Action/Response Sequences

Action: user opens the application

Response: application displays a login and sign up button.

Action: User clicks the sign-up button

Response: A prompt is displayed to input sign up information.

Action: User enters information and click sign up button.

Response: User is signed up to application and information is stored in database server. The avatar screen is displayed.

3.4.1.3. Functional Requirements

R1: User should be able to connect to servers successfully.

R2: A signup form should be displayed to the user.

R3: User should be able to sign up by filling the details in the form and submitting it through sign up button.

3.4.2. Login

3.4.2.1. Description

Once the account is created the user can login to their account anytime.

3.4.2.2. Action/Response Sequences

3.4.3. Action: User opens the application.

Response: Application displays a login and sign up button.

Action: User clicks the login button

Response: A prompt is displayed to input login credentials.

Action: User enters username and password and clicks login button if they remember password.

Response: If user credentials are correct, newsfeed page will be rendered, else error message is displayed.

Action: If user forgets password, the user will click forgot password.

Response: A prompt to enter email address will be displayed.

Action: User enters email where they want to get the forgot password link.

Response: An email is sent on that address with the password change link.

3.4.3.1. Functional Requirements

R1: User should be able to connect to servers successfully.

R2: A login form should be displayed to the user.

R3: User should be able to login by filling the correct credentials in the form and submitting it through login button.

R4: User should be able to reenter credentials if they are entered incorrectly up to 3 times.

R5: User should be able to click forgot password button if the credentials are wrong thrice.

3.4.4. Update Password

3.4.4.1. Description

User can change their password as desired.

3.4.4.2. Action/Response Sequences

Action: User logs into the application.

Response: Application displays newsfeed page with settings button.

Action: User clicks the settings button

Response: Options are displayed including change password.

Action: User clicks change password button.

Response: User is prompted to add current password twice and new password once.

Action: User enters the desired credentials and clicks update password button.

Response: Success message is displayed and database is updates.

3.4.4.3. Functional Requirements

R1: User should be able to update their password.

R2: System should update password in the database and display success message.

R3: User should be able to asked to enter current password before updating.

3.4.5. Add Review

3.4.5.1. Description

User can add review for a place where they have left their hologram.

3.4.5.2. Action/Response Sequences

Action: User anchors the hologram to a location.

Response: User is prompted to add review of the location. **Action:**

User writes review and clicks submit button.

Response: System records review in database and displays live view page.

3.4.5.3. Functional Requirements

R1: Users should be able to anchor their hologram.

R2: Users should be able to write a review.

R3: The review should be stored in database.

R4: The review should be accessible to other users.

3.4.6. Delete Review

3.4.6.1. Description

User can delete previously added reviews.

3.4.6.2. Action/Response Sequences

Action: User logs in and clicks on personal profile.

Response: Options including 'reviews written' are displayed. **Action:**

User clicks on reviews written option.

Response: System displays all review written by the user.

Action: User selects a review.

Response: A delete or update option is displayed.

Action: User clicks delete button.

Response: The review is deleted from the database and user is redirected to personal profile.

3.4.6.3. Functional Requirement

R1: Users should be able to view previous reviews.

R2: Users should be able to delete the review.

R3: System should update or delete the review in database after users' request.

R4: The deleted review should not be visible to other users.

3.4.7. Create Hologram

3.4.7.1. Description

User can create or leave their hologram at any place they like so others can interact with them.

3.4.7.2. Action/Response Sequences

Action: User logs in and clicks on live view.

Response: Options including 'Draw Hologram' are displayed.

Action: User clicks on 'Draw hologram' option.

Response: System displays a screen for user to draw hologram.

Action: User draws the hologram on screen.

Response: Suggestions for hologram categories are displayed.

Action: User selects hologram category and clicks confirm,

Response: A leave hologram button is displayed along with try again button.

Action: User clicks on leave hologram button.

Response: System stores the location and shape of hologram in the respective databases and redirects user to live view page.

3.4.7.3. Functional Requirement

R1: Users should be able to draw holograms on the screen.

R2: Users should be able to retry if the hologram is not drawn to their liking.

R3: System should store hologram data in database after users' request.

R4: The holograms of one user should be visible to other users.

R5: User should be able to interact with hologram via chat.

3.4.8. Create Avatar

3.4.8.1. Description

User can create or leave their avatar on the map at any place they like so others can interact with them.

3.4.8.2. Action/Response Sequences

Action: User logs in and clicks on maps.

Response: Options including 'Create Avatar' are displayed.

Action: User clicks on 'Create Avatar' option.

Response: System displays a screen for user to create avatar.

Action: User creates avatar by choosing multiple options on screen.

Response: A leave avatar button is displayed along with create again button.

Action: User clicks on leave avatar button.

Response: System stores the location and appearance of avatar in the respective databases and redirects user to maps page.

3.4.8.3. Functional Requirement

R1: Users should be able to create avatars on the screen.

R2: Users should be able to retry if the avatar is not created to their liking.

R3: System should store avatar data in database after users' request.

R4: The avatar of one user should be visible to other users.

R5: User should be able to interact with avatar via chat.

3.4.9. Update Avatar

3.4.9.1. Description

The user can update their avatar upon their need.

3.4.9.2. Action/Response Sequences

Action: User logs in and clicks on personal profile.

Response: Options including 'View Avatar' are displayed.

Action: User clicks on 'View Avatar' option.

Response: Buttons including 'Update Avatar' are displayed.

Action: User clicks 'Update Avatar' button, makes the wanted changes and presses 'Confirm update' button.

Response: System updates the appearance of avatar in the respective databases and redirects user to personal profile page.

3.4.9.3. Functional Requirement

R1: Users should be able to update avatars.

R2: Users should be able to retry if the avatar is not updated to their liking.

R3: System should store updated avatar data in database after users' request.

R4: The updated avatar of one user should be visible to other users.

3.4.10. Add Post

3.4.10.1. Description

User can add a video or any other media of a location.

3.4.10.2. Action/Response Sequences

Action: User logs in and presses the add post button.

Response: System opens a camera window and displays post media button.

Action: User takes picture or video of the location and clicks the post media button.

Response: The post is saved in database, displayed to user in personal profile and displayed to other users.

3.4.10.3. Functional Requirement

R1: User should be able to make a post such as image or video.

R2: User should be able to access camera.

R3: The post should be stored in database.

R4: The post should be accessible to other users when they search the location.

R5: The post history should be accessible to user who wrote it.

3.4.11. Delete Post

3.4.11.1. Description

User can delete previously added posts.

3.4.11.2. Action/Response Sequences

Action: User logs in and clicks on personal profile.

Response: Options including ‘previous posts’ are displayed.

Action: User clicks on ‘previous posts’ option.

Response: System displays all posts written by the user.

Action: User selects a post.

Response: A delete, or update option is displayed.

Action: User clicks delete button.

Response: The post is deleted from the database and user is redirected to personal profile.

3.4.11.3. Functional Requirement

R1: Users should be able to view previous posts.

R2: Users should be able to delete the posts.

R3: System should update or delete the posts in database after users’ request.

R4: The deleted posts should not be visible to other users.

R5: The deleted posts should be detached from the location.

3.4.12. View Personal Profile

3.4.12.1. Description

The user can view their personal profile which contains all the holograms which they

anchored to different locations and other info.

3.4.12.2. Action/Response Sequences

Action: User logs in and clicks on personal profile button.

Response: Personal profile is displayed.

3.4.12.3. Functional Requirement

R1: Users should be able to view and edit personal profile.

3.4.13. Delete Hologram Anchor

3.4.13.1. Description

The user can also delete the hologram he/she left anchored to a place.

3.4.13.2. Action/Response Sequences

Action: User logs in and clicks on personal profile.

Response: Options including 'previous holograms' are displayed.

Action: User clicks on 'previous holograms' option.

Response: System displays all holograms anchored by the user.

Action: User selects a hologram to remove.

Response: A delete option is displayed.

Action: User clicks delete button.

Response: The hologram is deleted from the database and user is redirected to personal profile.

3.4.13.3. Functional Requirement

R1: Users should be able to view previous holograms.

R2: Users should be able to delete the holograms.

R3: System should delete the hologram in database after users' request.

R4: The deleted holograms should not be visible to other users.

R5: The deleted holograms should be unanchored from the location.

3.4.14. Newsfeed

3.4.14.1. Description

System can display recommendations for places based on user's previously visited locations.

3.4.14.2. Action/Response Sequences

Action: User logs in.

Response: Newsfeed is displayed based on recommendation system if user is not a new one. The new user is shown newsfeed based on their personal profile.

3.4.14.3. Functional Requirement

R1: Users should be able to view newsfeed.

R2: System should store location history.

R3: System should apply recommender based on location history.

R4: System should display posts of recommended locations on newsfeed.

3.4.15. Chat

3.4.15.1. Description

User can interact with another person by messaging them.

3.4.15.2. Action/Response Sequences

Action: User clicks on a hologram on a location.

Response: System sends a message request to the hologram owner. If the owner accepts the request, system opens a chat window, else a rejected message is displayed.

Action: User can type a message to another user.

Response: The reply message from the user is displayed.

3.4.15.3. Functional Requirement

R1: Users should be able to send request to another user.

R2: System should store chat history.

R3: User should be able to send message to the other user.

R4: User should be able to accept message request of another user.

R5: User should get notification of a message.

3.4.16. Avatar Map

3.4.16.1. Description

User can look at avatars anchored on different locations through the avatar map.

3.4.16.2. Action/Response Sequences

Action: User logs in and clicks on view map.

Response: Avatars of other users are displayed on various locations of the map.

3.4.16.3. Functional Requirement

R1: Users should be able to view map.

R2: System should store avatars and their locations on the map.

R4: System should display avatars on locations.

R5: User should be able to interact with the user of avatar by tapping the avatar.

3.4.17. Scan Location

3.4.17.1. Description

User can view holograms anchored at a location by scanning their surroundings.

3.4.17.2. Action/Response Sequences

Action: User logs in and clicks on live view.

Response: Anchored holograms of other users are displayed on various locations of area.

3.4.17.3. Functional Requirement

R1: Users should be able to view area via camera.

R2: System should render AR holograms for the user.

R4: System should display the holograms on the location the user anchored them at.

R5: User should be able to chat with the holograms.

3.4.18. Sign out

3.4.18.1. Description

User can end their session by signing out.

3.4.18.2. Action/Response Sequences

Action User clicks on sign out button.

Response: User is redirected to login and signup page and session is ended.

3.4.18.3. Functional Requirement

R1: Users should be able to sign out.

3.4.19. Search Location

3.4.19.1. Description

User can search for a location or hologram in map.

3.4.19.2. Action/Response Sequences

Action: User clicks on Avatar Map

Response: User is redirected to Avatar Map page with a search bar.

Action: User types a location or hologram category to search.

Response: Location based on the user's search query is displayed.

3.4.19.3. Functional Requirement

R1: Users should be able to search for a location by hologram category or location name.

R2: System should display filtered location.

3.4.20. View Post

3.4.20.1. Description

User can view post of another user and get directed to the place where the post was made.

3.4.20.2. Action/Response Sequences

Action: User clicks on a post on newsfeed.

Response: A 'get directions' button is displayed.

Action: User clicks on 'get directions' button.

Response: System directs user to the hologram that was posted.

3.4.20.3. Functional Requirement

R1: Users should get directions to a hologram by clicking the post.

R2: Users should be able to view posts on news feed.

3.5. Other Non-Functional Requirements

3.5.1. Performance Requirements:

There should be enough storage to add hologram, images, text and other data in the database. DB system may easily handle 10 read transaction per hour but only 3 update transactions per hour.

3.5.2. Safety Requirements. .

The user session is protected, and credentials and holograms are preserved without being changed in the database.

3.5.3. Security Requirements:

Access permissions for the database system information may only be changed by the system's data administrator.

Holograms, recommendations and credentials can only be updated in database as per user's desire.

One user cannot delete the hologram of another user.

3.5.4. Software Quality Attributes:

3.5.4.1. Availability

When updating information, the database servers should be running.

The system may be available 90 percent of the time during a month.

3.5.4.2. Correctness

The anchors of avatars should be placed in correct location on the map.

The recommendation should generate correct results which match the user interests.

3.5.4.3. Extensibility and Maintainability

If holograms, recommendations and credentials change, the database should be updated.

3.5.4.4. Reliability

The probability of the algorithms to produce desired results should be more than 95%.

When hologram is deleted, system should also delete the corresponding avatar.

3.5.4.5. Usability

The font should be readable (greater than 12pt).

A online user manual should be available for guiding new users.

The icon sizes should be 36x36 or 48x48 pixels.

AR will make a more user friendly and engaging experience for user.

Up to 20 holograms will be displayed at a time to make it visually appealing.

3.5.4.6. Compatibility

The application must be compatible with Android firewall or antivirus protection.

CHAPTER: 4

DESIGN AND DEVELOPMENT

1. DESIGN AND DEVELOPMENT

1.1. Introduction

1.1.1. Purpose:

This chapter describes the architecture and system design of Flamingo. It mostly contains different design diagrams and their explanation. This portion of the document is intended to inform stakeholders of the details of the design and the design process. This document will help the developer(s) in implementation and maintenance of the Application (app).

1.1.2. Scope:

Flamingo will use AR Core. Holograms will be created by user when at a specific place using technology of AR Core and Android Studio.

- User will view holograms and interact with other users via chat.
- If user is at home, then avatars will appear as markers on a map.
- Directions to desired location will be provided to the user.
- Recommendations will be made based on interests and needs.

1.1.3. Document Overview:

This document shows the design and working of Flamingo. It starts from higher level details for a non-technical reader to understand just by seeing the diagrams to the lower-level details that aid the developer to code and understand other technical details of the application. Section 2 the **System Architecture Description** gives a detailed overview of the application. Section 2.1 **Structure and Relationships** shows the higher-level details system working by the means of System Block, Activity, and Use Case diagrams. Lower-level details are described using the Class, Entity Relationship diagrams, Sequence diagrams and Structure Chart. Section 2.2 describes how the application is designed to curb the tendency of **User Interface Issues** and problems during User Interaction.

In Section 3, **Detailed Description of Componen*t*** is given to show the working of modules with low level details. It shows the purpose, function, subordinates, dependencies, interfaces, resources, processing and data of the components and their relationships with each other.

Section 4 shows the **Reuse and Relationship to other Products** i.e., information about work done in the same project before and any reuse of the same work. The section also provides a key to reuse this system for further upgrades.

Section 5 **Design Decisions and Tradeoffs** shows the architecture style and design pattern of the application, while in the Section 6 the **Pseudo Code** of the components is given in for human reading rather than machine reading.

1.2. Work Breakdown Structure:

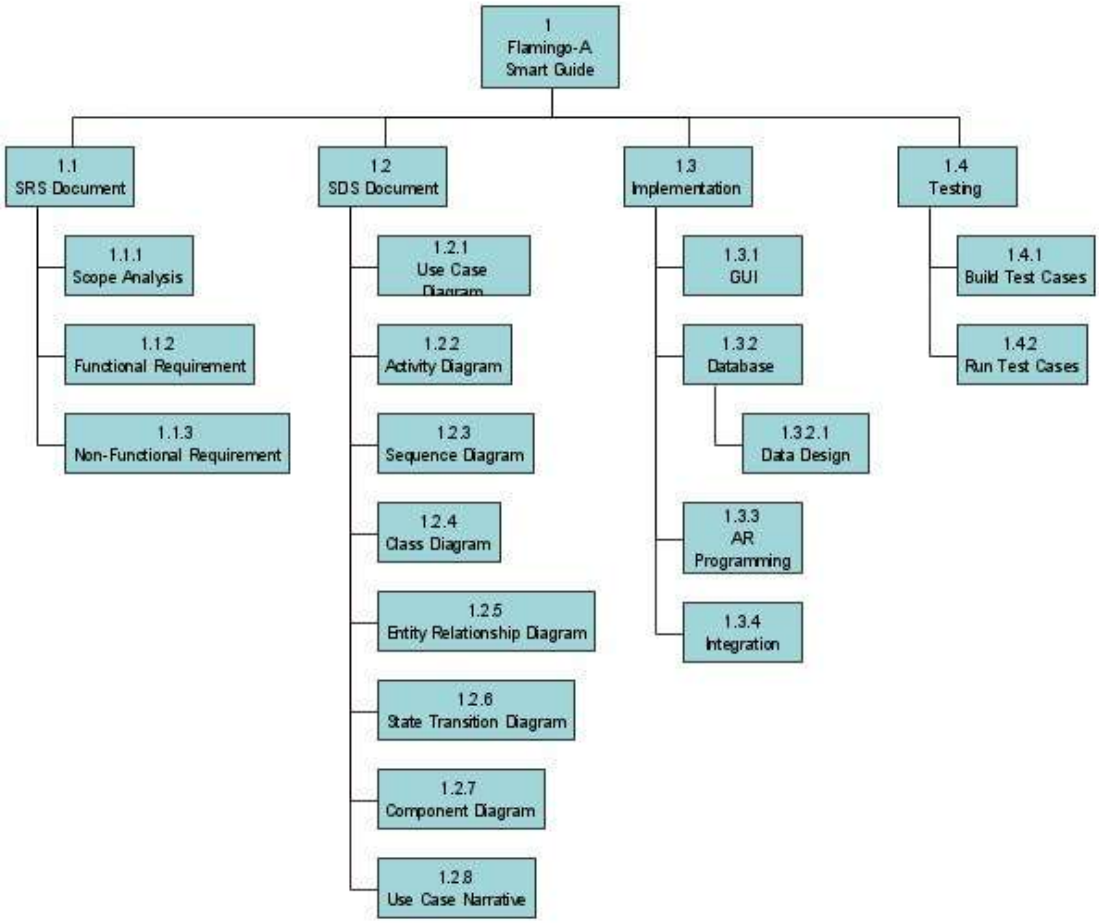


Figure 4. 1 Work Breakdown

1.3. System Architecture Description

Detailed description of system architecture and design pattern which this system is going to use is discussed later in the document in section 5 ‘Design Decisions and Tradeoffs’. This Section gives overview of application, its higher and lower levels details and user interfaces.

1.4. Overview of Modules

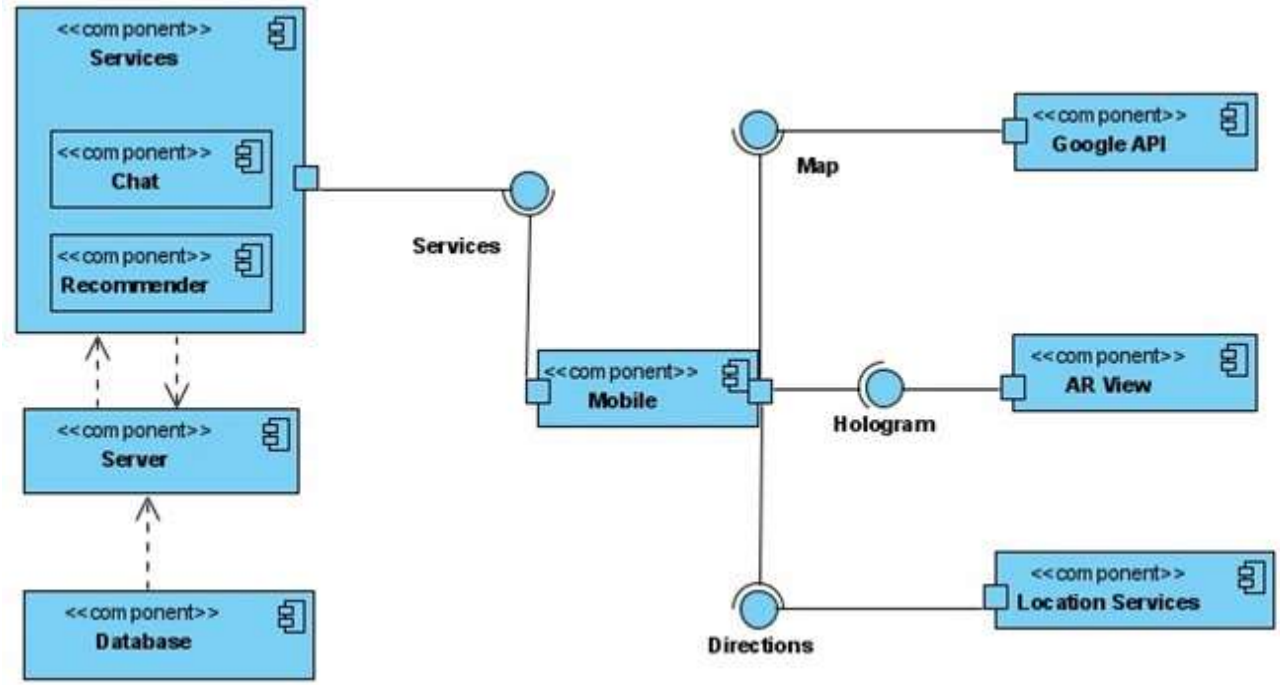


Figure 4.2 Component Diagram

The architectural design of Flamingo is Model-View-Controller Architecture. Model-View-

Controller divides the system into the following modules to achieve the complete functionality.

1) User interaction system-View

The user interacts with the user interface system through signup or login.

2) Recommendation system-Controller

The user will get recommendations based on previous visited locations.

3) Database- Model

It stores the holograms, Location of the hologram and the application and user data.

1.5. Structure and Relationships

This section covers the technical description of Flamingo. It shows relationships between different components and how system modules are connected. This section also covers working with respect to different point-of-views. This also covers its higher and lower levels details, user interfaces, and system architecture and design pattern.

1.5.1. System Block Diagram

This diagram shows the higher-level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

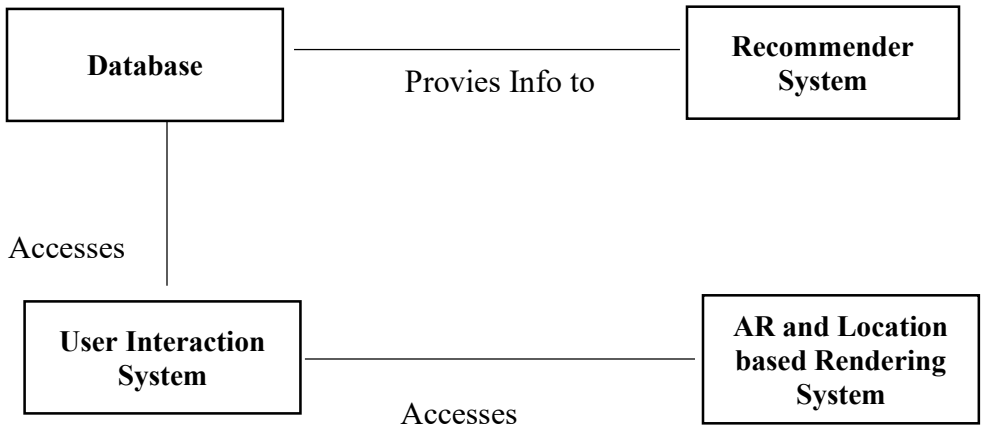


Figure 4. 3 Block Diagram

1.5.2. Database Design

Our application uses three kinds of databases to store the major data entries of the application described as follows

- Firebase stores the user credentials, posts, reviews, avatar, category, and the anchor Ids of the hologram anchored to the specific location.
- Azure Spatial Anchors store the location of the hologram.

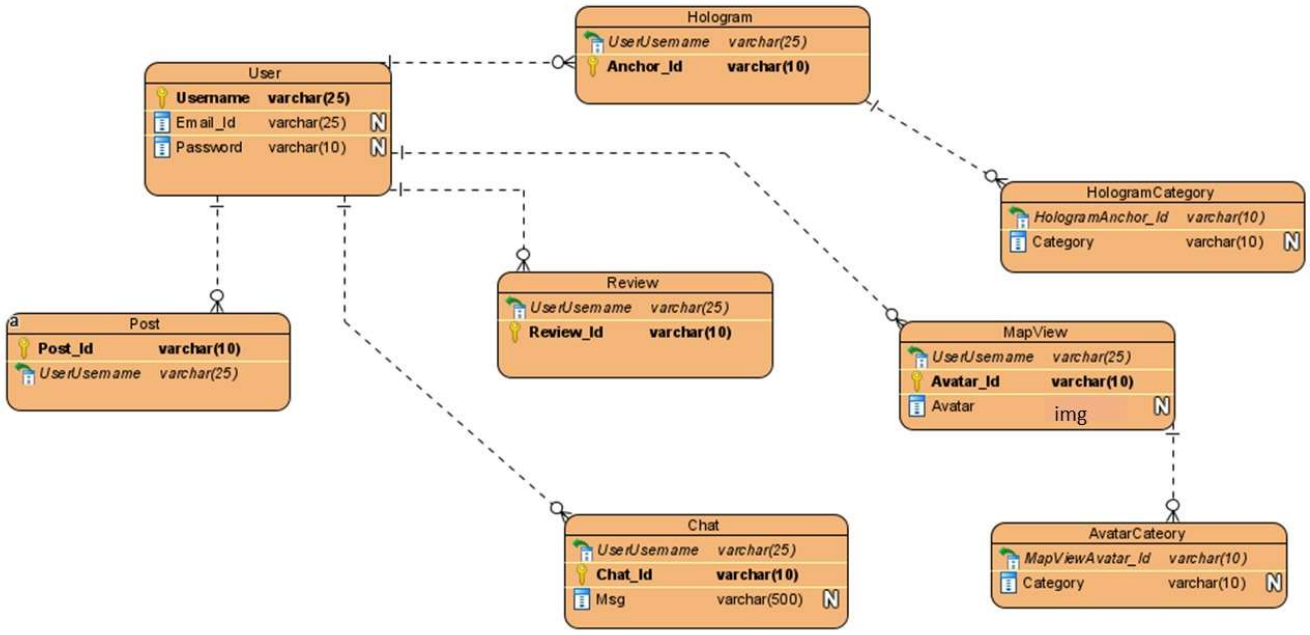


Figure 4. 4 ERD Diagram

1.5.3. User View (Use case Diagram)

Following diagram shows course of events that take place when an actor (user and other allowed interactions) interacts with system.

Figure 4. 5 Use Case Diagram

1.5.3.1. Actors

1. Admin
2. User

1.5.3.2. Use Cases

1. Login
2. Suspend User
3. Monitor User
4. Unsuspend User
5. Signup
6. Update Password
7. Add Review
8. Delete Review
9. Create Hologram
10. Create Avatar
11. Add Post
12. Delete Post
13. Logout
14. View Personal Profile
15. Delete Hologram
16. View Hologram
17. View Newsfeed
18. Chat
19. Update Avatar
20. View Avatar Map
21. Scan Location
22. Report User

1.5.3.3. Use Case Description

4.5.3.3.1. Signup

Use Case: Sign Up
Actors: User
Use Case Description: User can create an account by adding their personal information.
Normal Flow: • User opens the application. • User clicks the signup button. • User enters information and signs up.
Alternate Flow: • An error message is displayed if server is down.
Preconditions: User should be able to connect to servers successfully. A signup form should be displayed to the user. User should have an email address.
Postconditions: User is signed up to application and information is stored in database server. The create avatar screen is displayed.
Includes: N/A
Extends: N/A

4.5.3.3.2. Login

Use Case: Login
Actors: User
Use Case Description: Once the account is created the user can login to their account anytime.
Normal Flow: • User opens the application. • User clicks the login option. • User enters username and password and clicks login button.
Alternate Flow: • If incorrect username or password is entered show an error message and login will not be successful.
Preconditions: User should be able to connect to servers successfully. A login form should be displayed to the user. A user must also already be signed up.
Postconditions: If credentials are correct, newsfeed page will be rendered
Includes: N/A
Extends: N/A

4.5.3.3.3. Monitor User

Use Case: Monitor User
Actors: Admin
Use Case Description: Admin can view the posts of the User and also view the posts of user.
Normal Flow: • Admin opens the application.

• Admin clicks the view posts or view reviews button. • Admin views the posts and reviews of desired user.
Alternate Flow: • An error message is displayed if server is down.
Preconditions: Admin should be able to connect to servers successfully. Reviews and posts of a specific user should exist. Admin should have an email address.
Postconditions: The posts and reviews screen is displayed.
Includes: N/A
Extends: N/A

4.5.3.3.4. Suspend User

Use Case: Suspend User
Actors: Admin
Use Case Description: Admin can suspend a user temporarily or permanently if they don't follow a decent code of conduct.
Normal Flow: • Admin opens the application. • Admin views the reported users. • User views content they have posted or their reviews. • Admin suspends user temporarily or permanently.
Alternate Flow: • An error message is displayed if server is down.
Preconditions: Admin should be able to connect to servers successfully.

Reviews and posts of a specific user should exist.
Postconditions: The successfully suspended screen is displayed.
Includes: N/A
Extends: N/A

4.5.3.3.5. Unsuspend User

Use Case: Unsuspend User
Actors: Admin
Use Case Description: Admin can Unsuspend a user.
Normal Flow: • Admin opens the application. • Admin views the suspended users. • Admin click Unsuspend User.
Alternate Flow: • An error message is displayed if server is down.
Preconditions: Admin should be able to connect to servers successfully. Admin should be logged in.
Postconditions: The success message is displayed.
Includes: N/A
Extends: N/A

4.5.3.3.6. Update Password

Use Case: Update Password
Actors: User
Use Case Description: User can change their password as desired.
Normal Flow: • User logs into the application. • User clicks the settings button. • User clicks change password button. • User enters the desired credentials and clicks update password button.
Alternate Flow: • If the user enters, the incorrect current password, error message is displayed.
Preconditions: User should be able to be asked to enter current password before updating.
Postconditions: Success message is displayed, and database is updated.
Includes: N/A
Extends: N/A

4.5.3.3.7. Add Review

Use Case: Add Review
Actors: User
Use Case Description: This use case allows the user to add review for a place where they have left their hologram.
Normal Flow: • User anchors the hologram to a location. • User writes review and clicks submit button.
Alternate Flow: • If the user leaves the review field empty, an error message will be displayed.
Preconditions:

Users should be able to anchor their hologram.
Postconditions: System records review in database and displays live view page.
Includes: N/A
Extends: N/A

4.5.3.3.8. Delete Reviews

Use Case: Delete Review
Actors: User
Use Case Description: This use case allows the user to delete previously added reviews.
Normal Flow: • Normal User logs in and clicks on personal profile. • User clicks on reviews written option. • User selects a review. • User clicks delete button.
Alternate Flow: • If the user is not able to delete the review, an error message will be displayed.
Preconditions: Users should be able to view previous reviews.
Postconditions: The review is deleted from the database and user is redirected to personal profile.
Includes: N/A
Extends: N/A

4.5.3.3.9. Leave Hologram

Use Case: Create/Leave Hologram
Actors: User
Use Case Description: This use case allows the user to create or leave their hologram at any place they like so others can interact with them.
Normal Flow: • User logs in and clicks on live view. • User clicks on 'Draw hologram' option. • User draws the hologram on screen. • User clicks on leave hologram button.
Alternate Flow: • If the hologram is not anchored to a location, error message will be displayed.
Preconditions: A user must be able to login to their account. Users should be able to draw holograms on the screen. Users should be able to retry if the hologram is not drawn to their liking.
Postconditions: System stores the location and shape of hologram in the respective databases and redirects user to live view page.
Includes: N/A
Extends: N/A

4.5.3.3.10. Create Avatar

Use Case: Create Avatar
Actors: User
Use Case Description: This use case allows the user to create or leave their avatar on the map at any place they like so others can interact with them.
Normal Flow:

• Normal User logs in and clicks on maps. • User clicks on ‘Create Avatar’ option. • User creates avatar by choosing multiple options on screen.
Alternate Flow: • If the avatar is not created, error message will be displayed.
Preconditions: A user must be able to login to their account. Users should be able to create avatars on the screen. Users should be able to retry if the avatar is not created to their liking.
Postconditions: System stores the location and appearance of avatar in the respective databases and redirects user to maps page.
Includes: N/A
Extends: N/A

4.5.3.3.11. Add Post

Use Case: Add Post
Actors: User
Use Case Description: This use case allows the user to add a video or any other media of a location.
Normal Flow: • User logs in and presses the add post button. • User takes picture or video of the location and clicks the post media button.
Alternate Flow: • Display an error message if camera cannot be accessed.
Preconditions: A user must be able to login to their account. User should be able to access camera.
Postconditions: The post is saved in database, displayed to user in personal profile and displayed to other users.
Includes: N/A
Extends: N/A

4.5.3.3.12. Delete Post

Use Case: Delete Post
Actors: User
Use Case Description: This use case allows the user to delete previously added posts.
Normal Flow: • User logs in and clicks on personal profile. • User clicks on 'previous posts' option. • User selects a post. • User clicks delete button.
Alternate Flow: • Display an error message if previous posts cannot be viewed.
Preconditions: A user must be able to login to their account. User should be able to view previous posts.
Postconditions: The post is deleted from the database and user is redirected to personal profile.
Includes: N/A
Extends: N/A

4.5.3.3.13. View Personal Profile

Use Case: View personal profile
Actors: User
Use Case Description: This use case allows the user to view their personal profile which contains all the holograms which they anchored to different locations and other information.
Normal Flow: • User logs in and clicks on personal profile button.
Alternate Flow: • Display an error message if login fails.
Preconditions: A user must be able to login to their account.
Postconditions: Personal profile is displayed
Includes: N/A
Extends: Update Avatar

4.5.3.3.14. Delete Hologram Anchor

Use Case: Delete Hologram Anchor
Actors: User
Use Case Description: This use case allows the user to delete the hologram he/she left anchored to a place.
Normal Flow: • User logs in and clicks on personal profile. • User clicks on 'previous holograms' option. • User selects a hologram to remove. • User clicks delete button.
Alternate Flow: • Display an error message if previous holograms cannot be viewed.
Preconditions: A user must be able to login to their account. Users should be able to view previous holograms.
Postconditions: The hologram is deleted from the database and user is redirected to personal profile.
Includes: N/A
Extends: N/A

4.5.3.3.15. View Anchored Hologram

Use Case: View anchored hologram
Actors: User
Use Case Description: This use case allows the user to look at a history of all the locations of their previously placed holograms.
Normal Flow: • User logs in and clicks on personal profile. • User clicks on 'previous holograms' option.
Alternate Flow: • Display an error message if the holograms history cannot be accessed.
Preconditions: A user must be able to login to their account. System should store holograms history.
Postconditions: System displays all holograms anchored by the user.
Includes: N/A
Extends: N/A

4.5.3.3.16. Newsfeed

Use Case: Newsfeed
Actors: User
Use Case Description: This use case allows the system to display recommendations for places based on user's previously visited locations
Normal Flow: • User logs in.
Alternate Flow: • Display an error message if the login fails.
Preconditions: A user must be able to login to their account. System should store location history.
Postconditions: Newsfeed is displayed based on recommendation system if user is not a new one. The new user is shown newsfeed based on their personal profile.
Includes: N/A
Extends: N/A

4.5.3.3.17. Chat

Use Case: Chat
Actors: User
Use Case Description: This use case allows the User to interact with another person by messaging them.
Normal Flow: • User logs in and clicks on live view. • User clicks on a hologram on a location. • System sends a message request to the hologram owner.
Alternate Flow: • System sends a message request to the hologram owner. If the owner rejects the request, a rejected message is displayed.
Preconditions: A user must be able to login to their account. User must be able to have live view and also view the holograms anchored by other users. Message request must be accepted by the hologram owner.
Postconditions: If the owner accepts the request, system opens a chat window.
Includes: N/A
Extends: N/A

4.5.3.3.18. Avatar Map

Use Case: Avatar Map
Actors: User
Use Case Description: This use case allows the User to look at avatars anchored on different locations through the avatar map.
Normal Flow: • User logs in and clicks on view map.
Alternate Flow: •
Preconditions: A user must be able to login to their account. System should store avatars and their locations on the map.
Postconditions: Avatars of other users are displayed on various locations of the map.
Includes: N/A
Extends: N/A

4.5.3.3.19. Update Avatar

Use Case: Update Avatar
Actors: User
Use Case Description: This use case allows the user to update their avatar upon their need.
Normal Flow: • User logs in and clicks on personal profile. • User clicks on 'View Avatar' option. • User clicks 'Update Avatar' button, makes the wanted changes and presses 'Confirm update' button.
Alternate Flow: • Display an error message if update is not successful.

Preconditions: A user must be able to login to their account.
Postconditions: System updates the appearance of avatar in the respective databases and redirects user to personal profile page.
Includes: N/A
Extends: N/A

4.5.3.3.20. Scan Location

Use Case: Scan Location
Actors: User
Use Case Description: This use case allows the user to holograms anchored at a location by scanning their surroundings.
Normal Flow: • User logs in and clicks on live view
Alternate Flow: • Error message is displayed if camera cannot be accessed.
Preconditions: A user must be able to login to their account. Users should be able to view area via camera System should render AR holograms for the user.
Postconditions: Anchored holograms of other users are displayed on various locations of area.
Includes: N/A
Extends: N/A

4.5.3.3.21. Report User

Use Case: Report User

Actors: User
Use Case Description: A user can report a user if they don't follow a decent code of conduct.
Normal Flow: • User opens the application. • User views content others users have posted or their reviews. • User clicks report user button.
Alternate Flow: • An error message is displayed if server is down.
Preconditions: User should be able to connect to servers successfully. Reviews and posts of a specific user should exist.
Postconditions: The successfully reported screen is displayed.
Includes: N/A
Extends: N/A

4.5.3.3.22. Sign out

Use Case: Sign out
Actors: User
Use Case Description: This use case allows the user to end their session by signing out.
Normal Flow: • User logs in. • User clicks on sign out button.
Alternate Flow: • Error message is displayed if internet connection is poor.
Preconditions: A user must be able to login to their account.
Postconditions: User is redirected to login and signup page and session is ended.

Includes: N/A
Extends: N/A

1.5.4. Sequence Diagram

Following sequence diagrams show the sequence of activities performed in key use cases described in section 4.5.3.

1.5.4.1. Sign Up

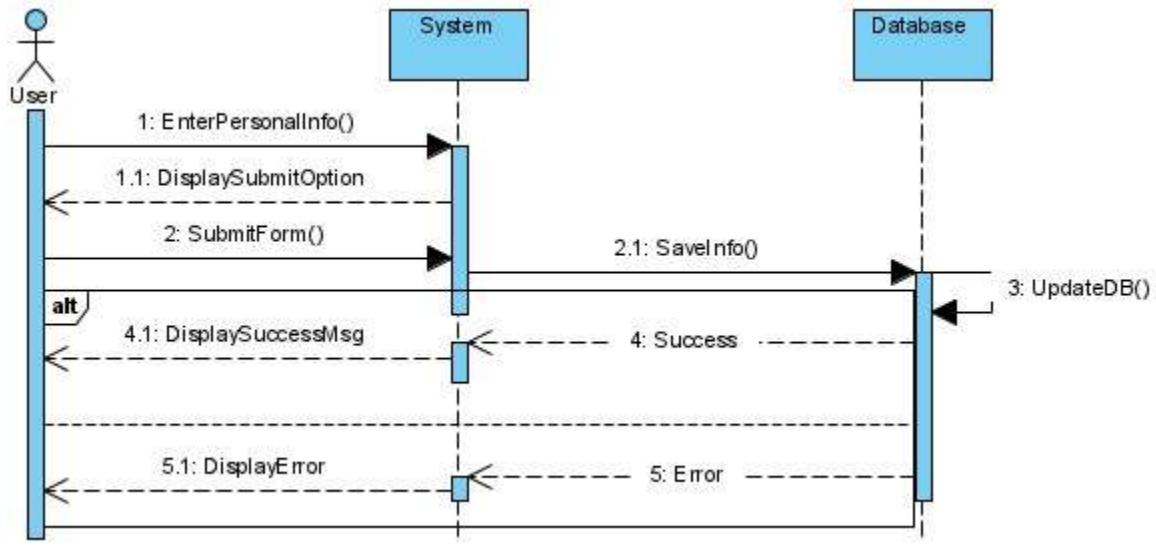


Figure 4. 6 Sign Up Sequence Diagram

1.5.4.2. Login

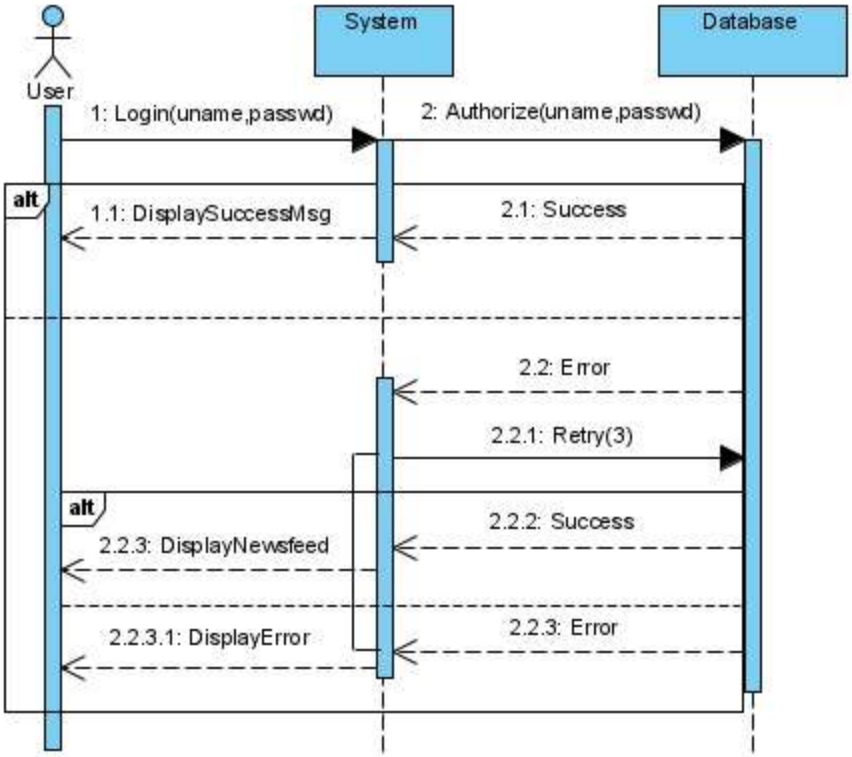


Figure 4. 7 Login Sequence Diagram

1.5.4.3. Update Password

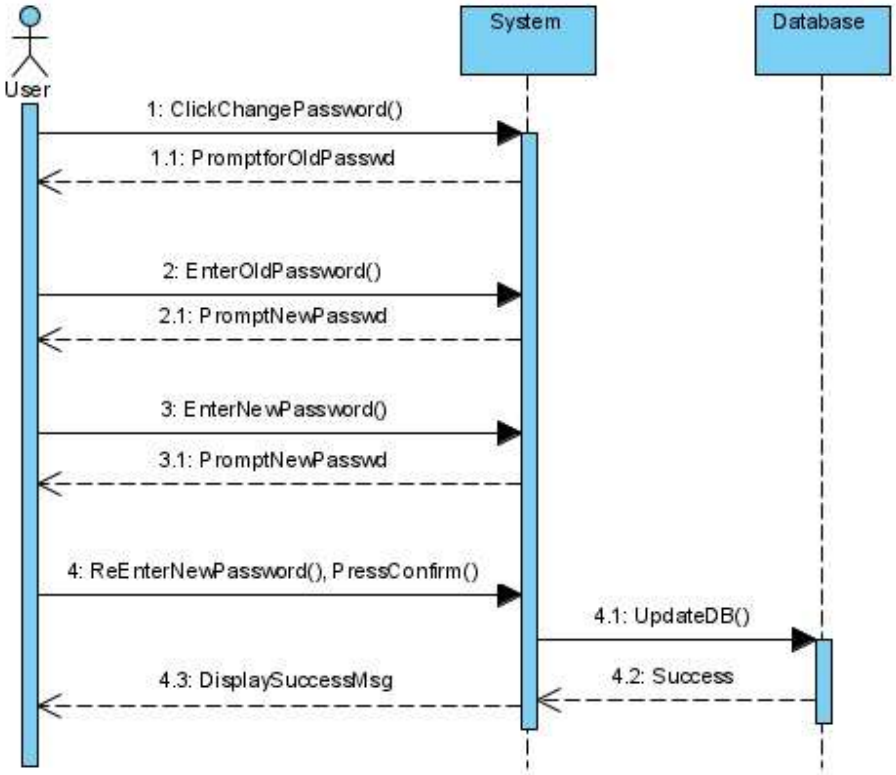


Figure 4. 8 Update Password Sequence Diagram

1.5.4.4. Add Review

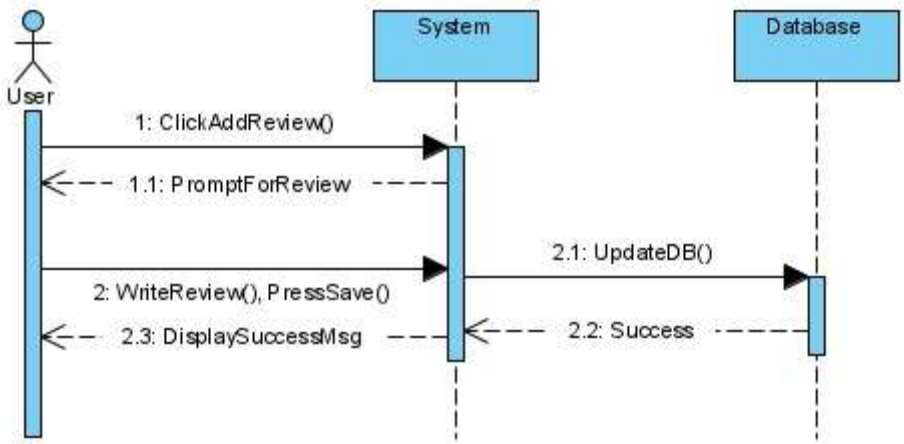


Figure 4. 9 Add Review Sequence Diagram

1.5.4.5. Delete Review

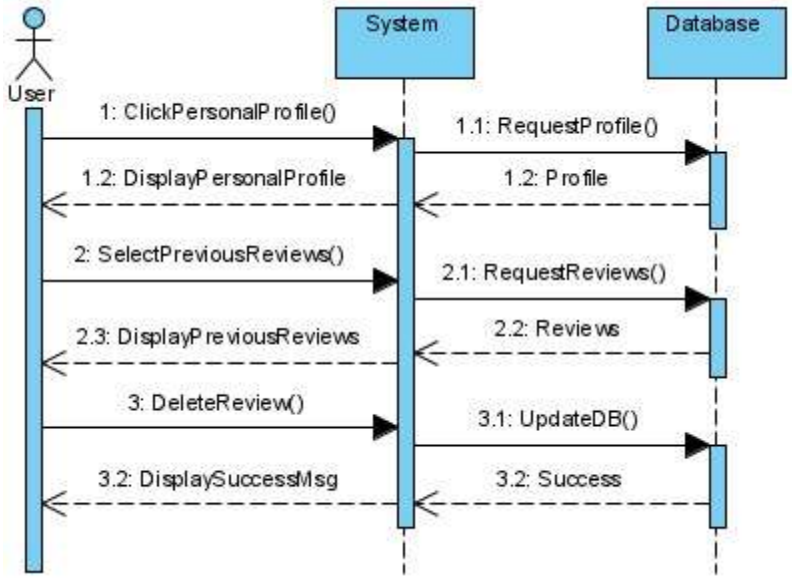


Figure 4. 10 Delete Review Sequence Diagram

1.5.4.6. Create Hologram

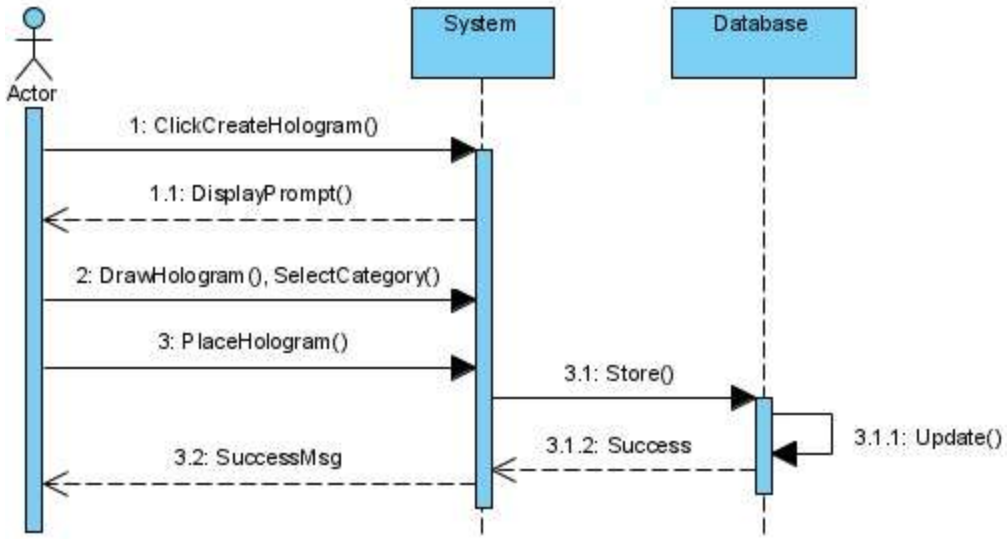


Figure 4. 11 Create Hologram Sequence Diagrams

1.5.4.7. Create Avatar

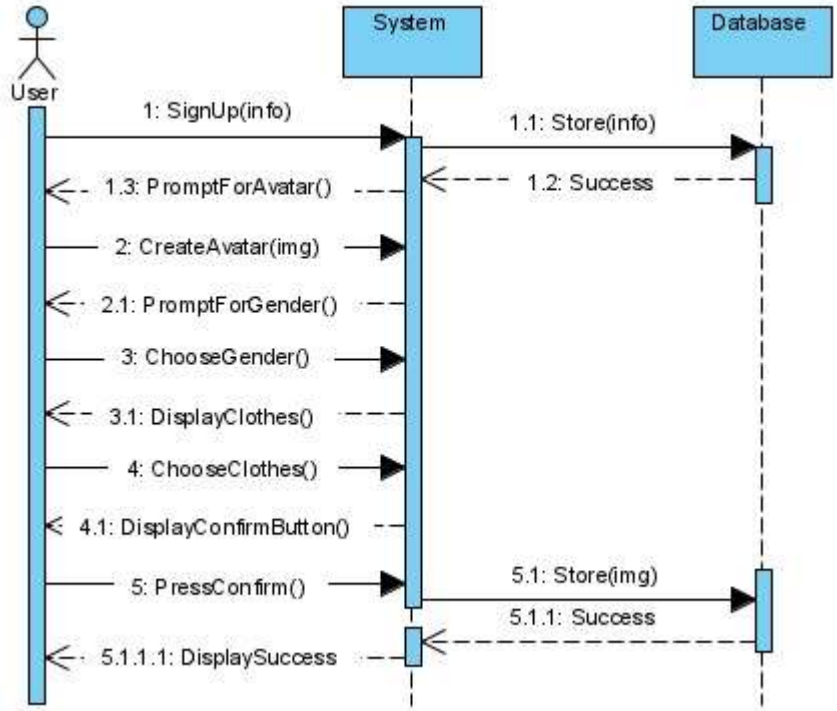


Figure 4. 12 Create Avatar Sequence Diagrams

1.5.4.8. Update Avatar

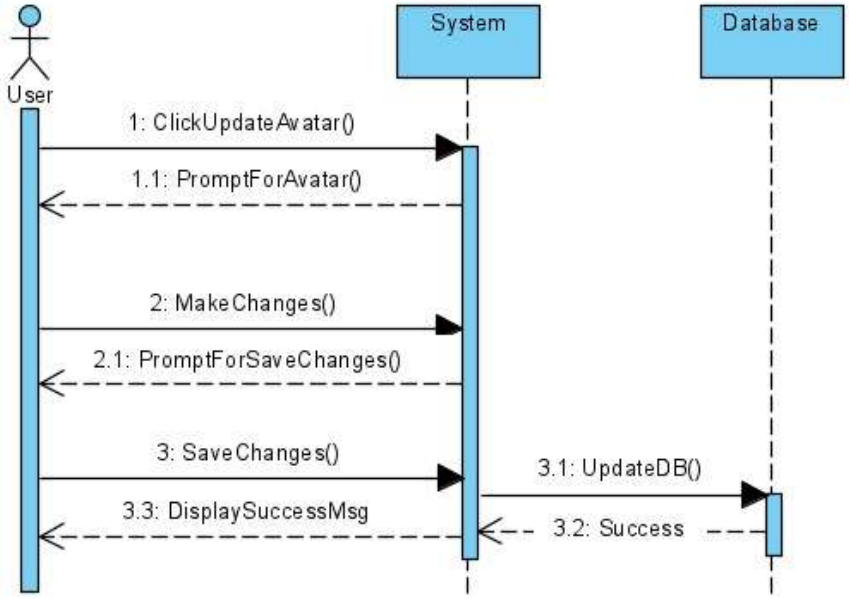


Figure 4. 13 Update Avatar Sequence Diagrams

1.5.4.9. Add Post

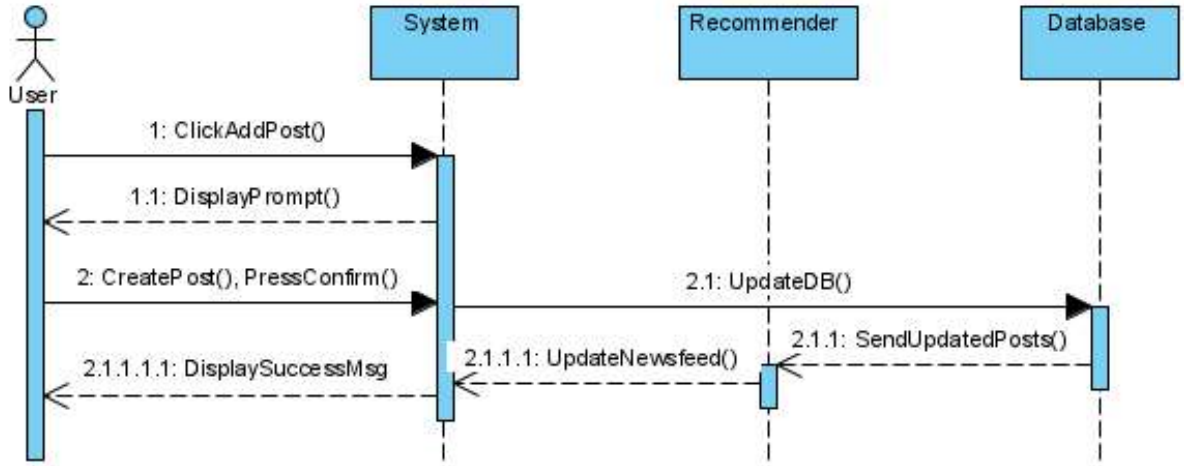


Figure 4. 14 Add Post Sequence Diagrams

1.5.4.10. Delete Post

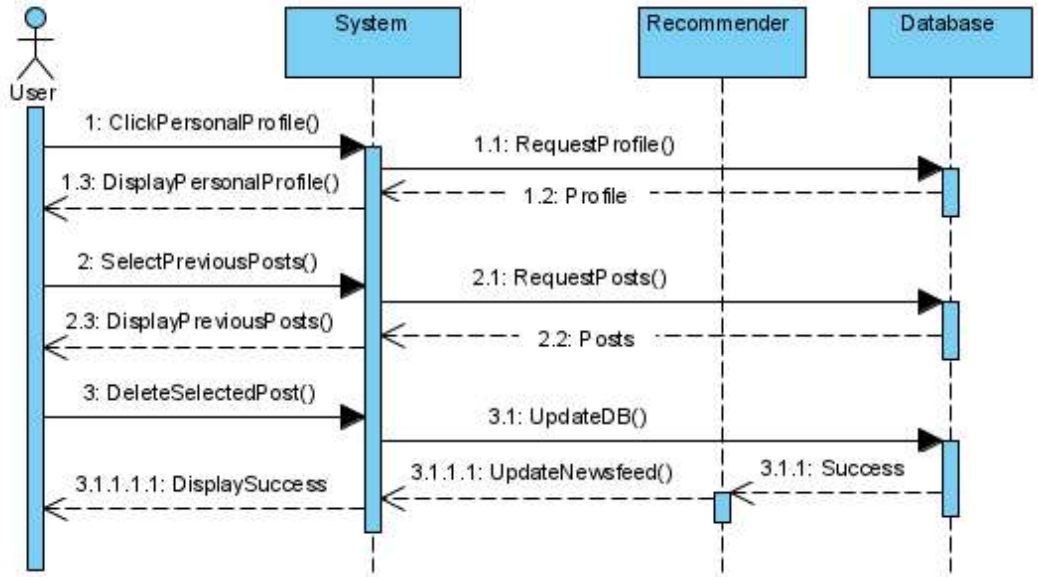


Figure 4. 15 Delete Post Sequence Diagrams

1.5.4.11. Delete Hologram Anchor

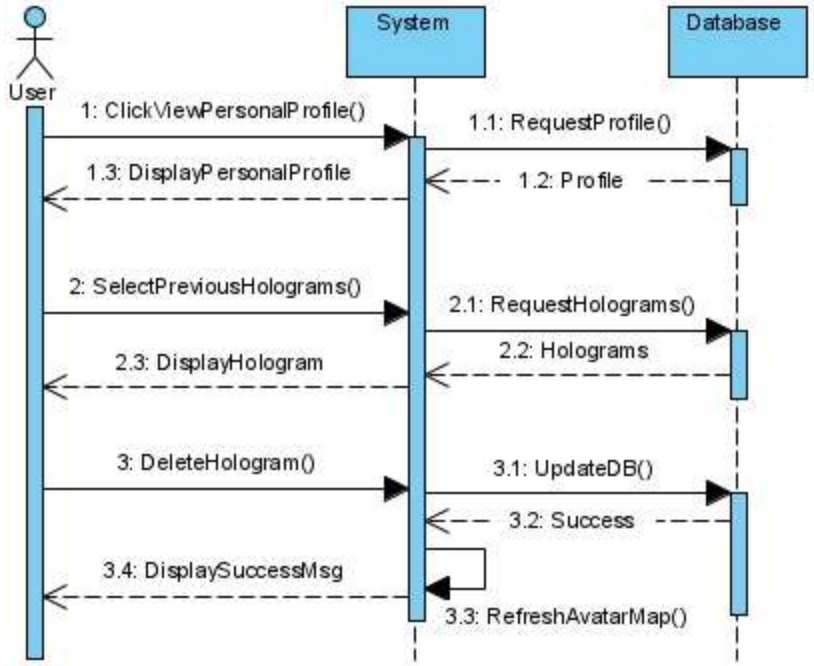


Figure 4. 16 Delete Hologram Anchor Sequence Diagrams

1.5.4.12. View Personal Profile

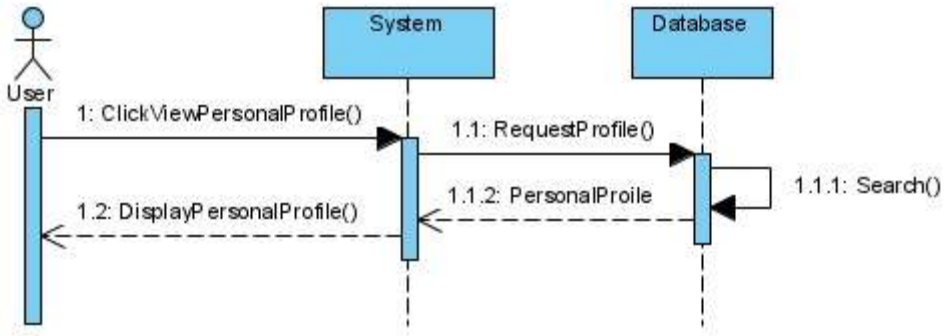


Figure 4. 17 View Personal Profile Sequence Diagrams

1.5.4.13. Newsfeed

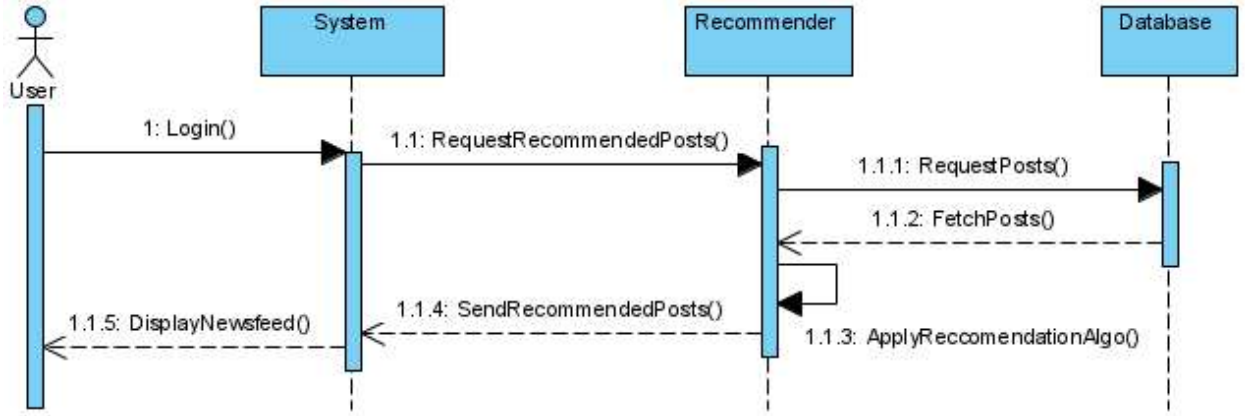


Figure 4. 18 Newsfeed Sequence Diagrams

1.5.4.14. Chat

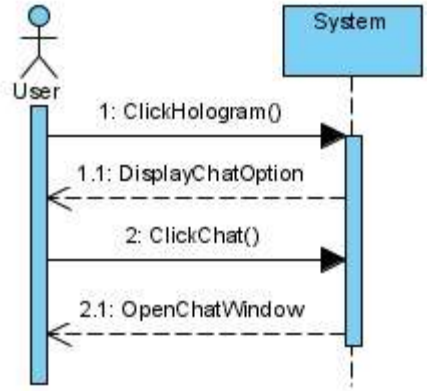


Figure 4. 19 Chat Sequence Diagrams

1.5.4.15. Avatar Map

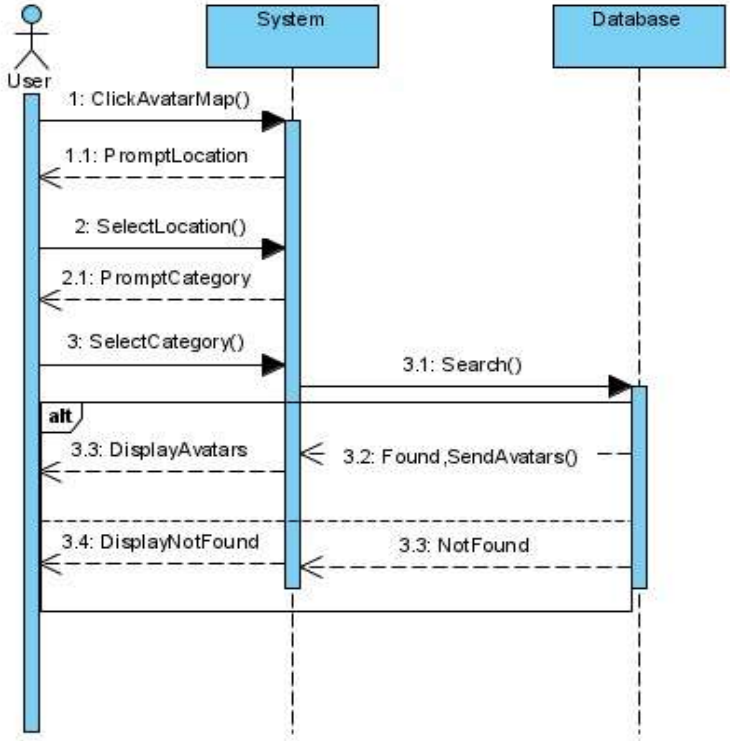


Figure 4. 20 Avatar Map Sequence Diagrams

1.5.4.16. Scan Location

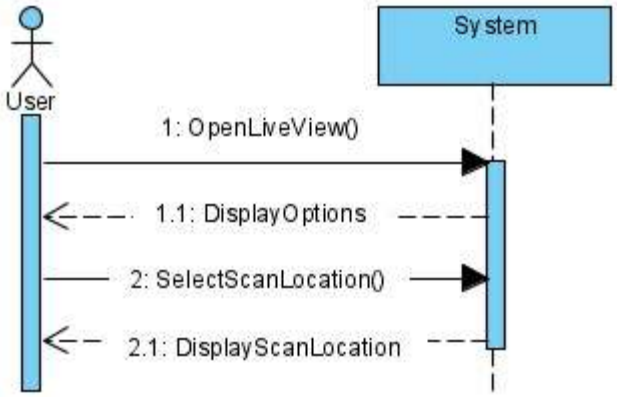


Figure 4. 21 Scan Location Sequence Diagrams

1.5.4.17. Sign out

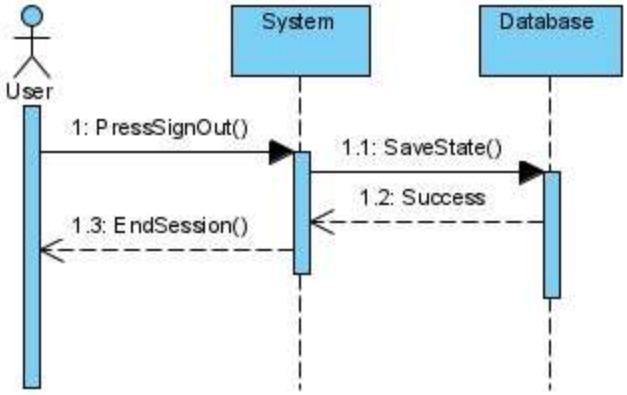


Figure 4. 22 Sign out Sequence Diagram

1.5.4.18. Admin

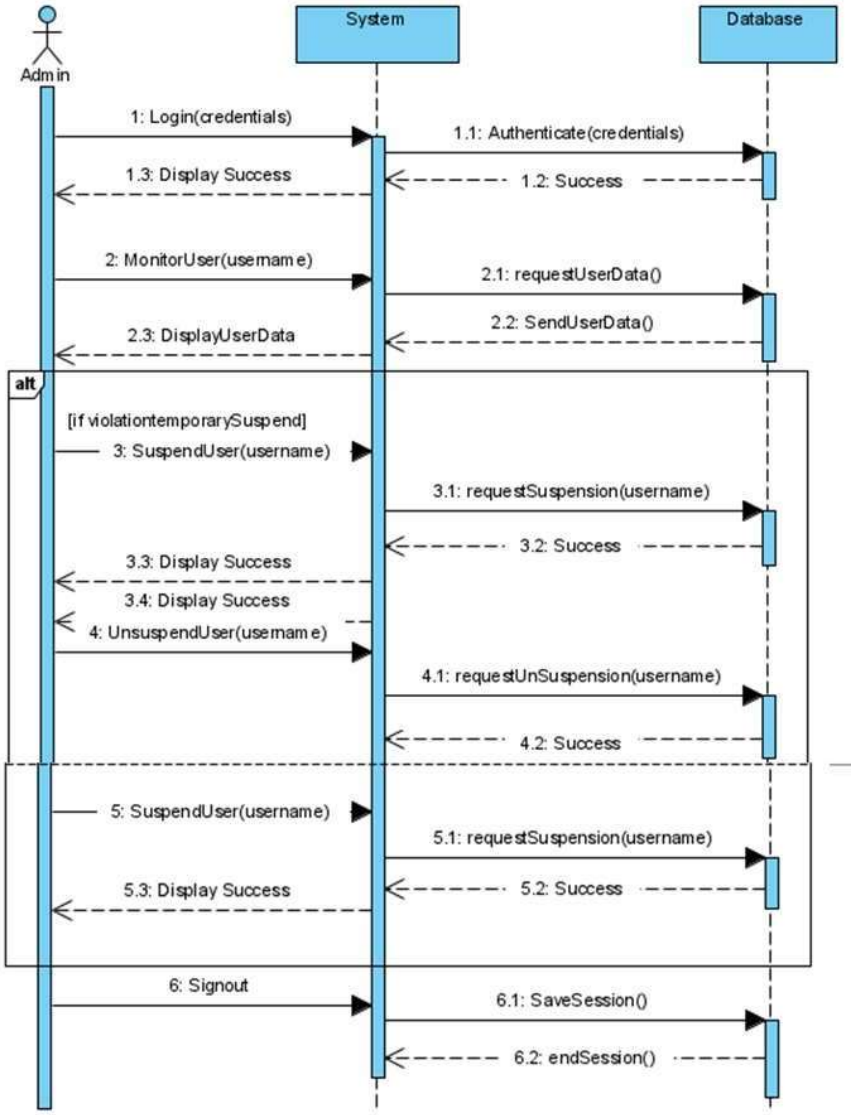


Figure 4. 23 Admin Sequence Diagram

1.5.5. Implementation View (Class Diagram)

The class diagram represents the static view of the application describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among objects.

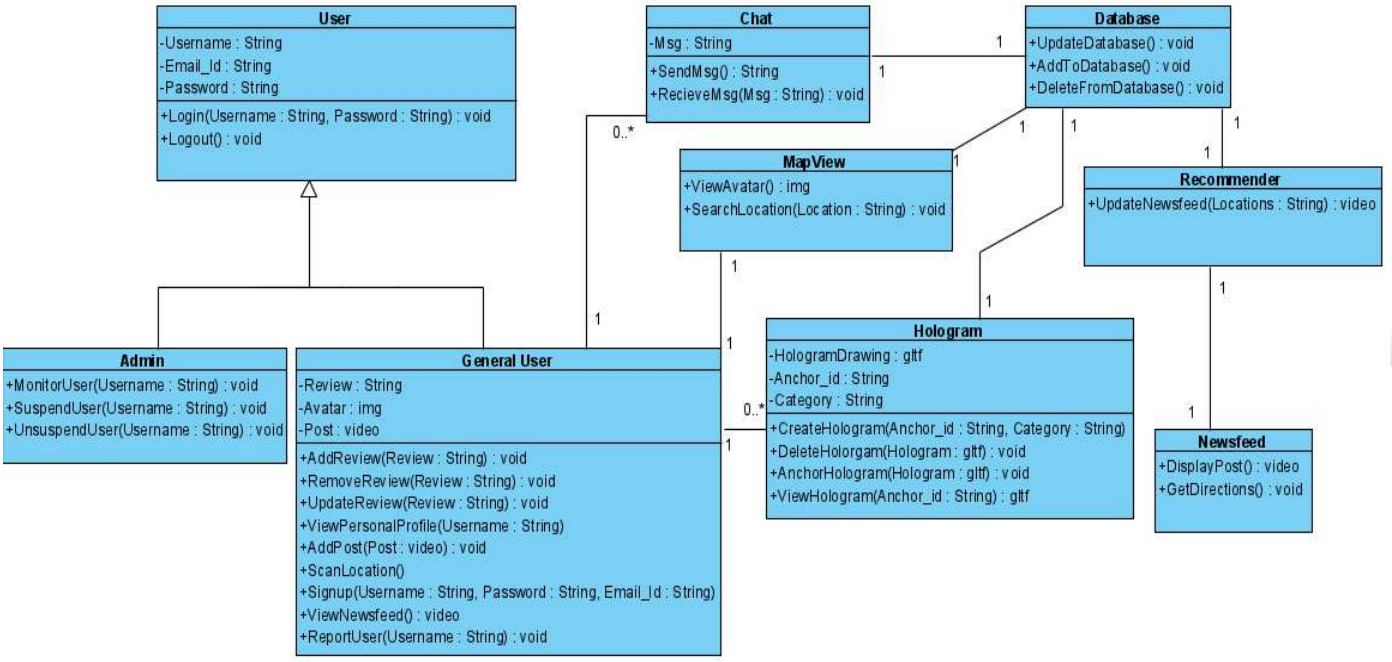


Figure 4. 24 Class Diagram

Table 4. 1 Classes Description

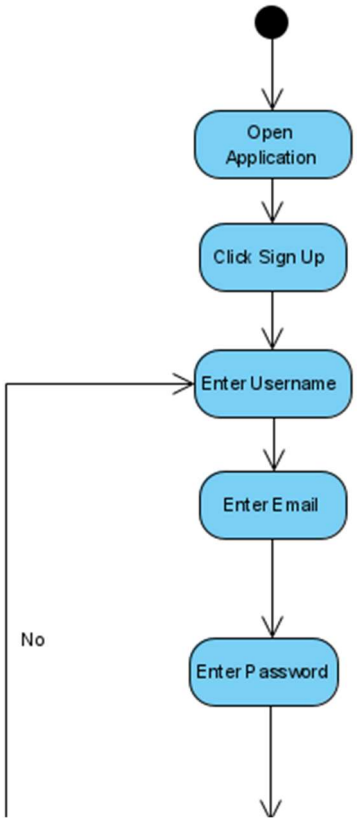
Classes	Description
User	This is the main class of the application. It will be executed first in the program. The user interacts with the application through login. Each user has an email, username, and password.
Admin	Admin can monitor, suspend, and unsuspend user.
General User	The general user can sign up, view his/her profile, create an avatar, create a hologram, and anchor it to a specific location, scan location and view avatar map, add, update, delete review/post and report a user.
Newsfeed	It displays recommendations for places based on user's previously visited locations. and by selecting the specific posts of interest it directs to a specific direction.
Chat	This class allows the general user to interact with another person by messaging them.
Hologram	User can create or leave their hologram at any place they like so others can interact with them, add review for a place where they have left their hologram, delete previously added reviews. User can also delete the hologram anchored to a specific place.
Map View	User can look at avatars anchored on different locations through the avatar map and search the locations.

Database	It stores the holograms, Location of the hologram and the application and user data. Data is updated, added, or deleted as per users need and choice.
Recommender	It sorts recommendations based on users' interests.

1.5.6. Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.

1.5.6.1. Sign Up



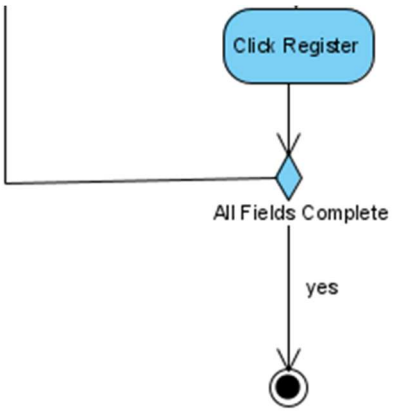


Figure 4. 25 Sign up Activity Diagram

1.5.6.2. Login

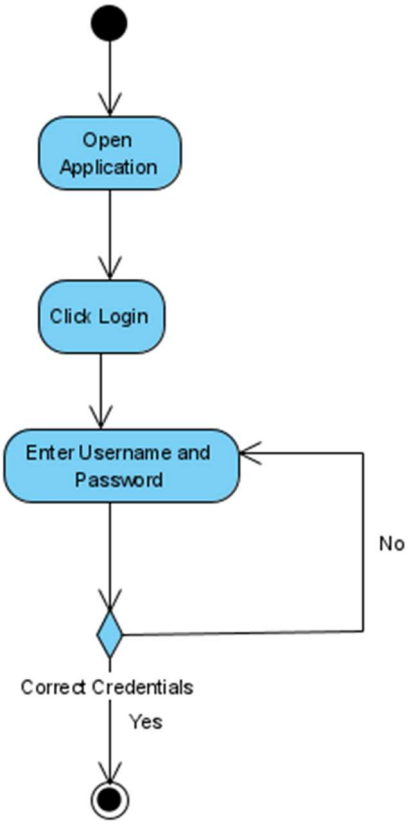


Figure 4. 26 Login Activity Diagram

1.5.6.3. Update Password

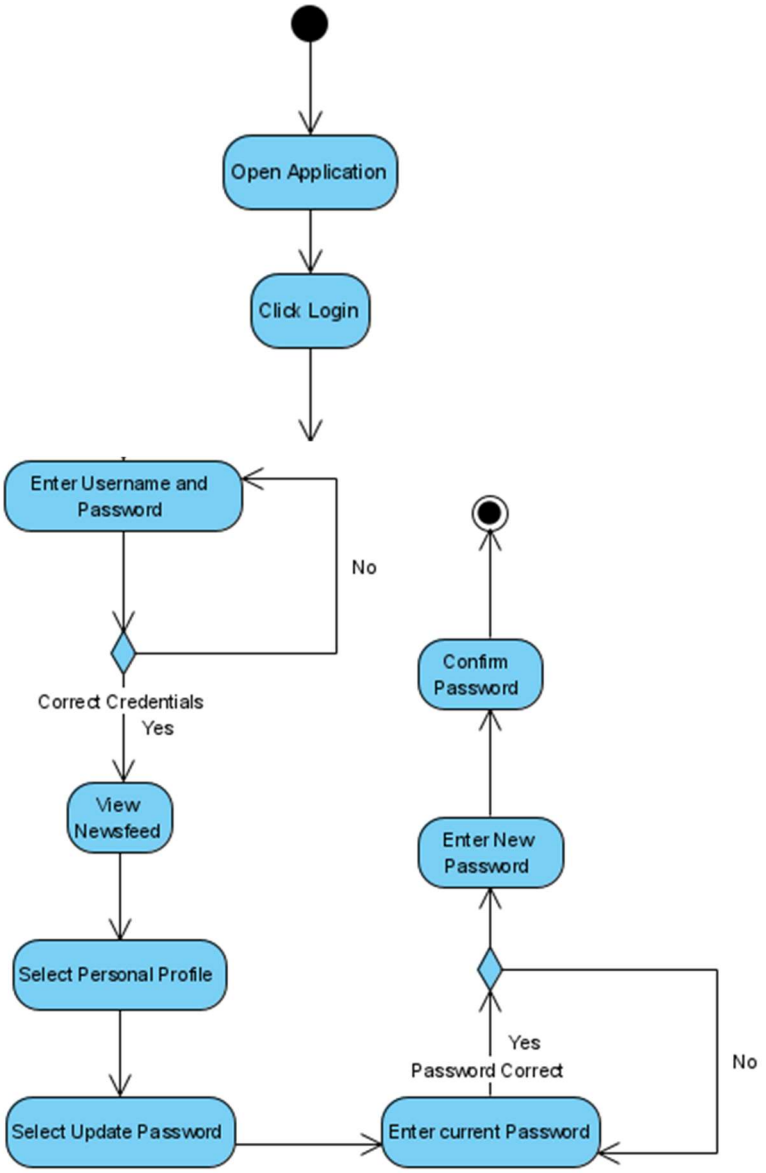


Figure 4. 27 Update Password Activity Diagram

1.5.6.4. Add Review

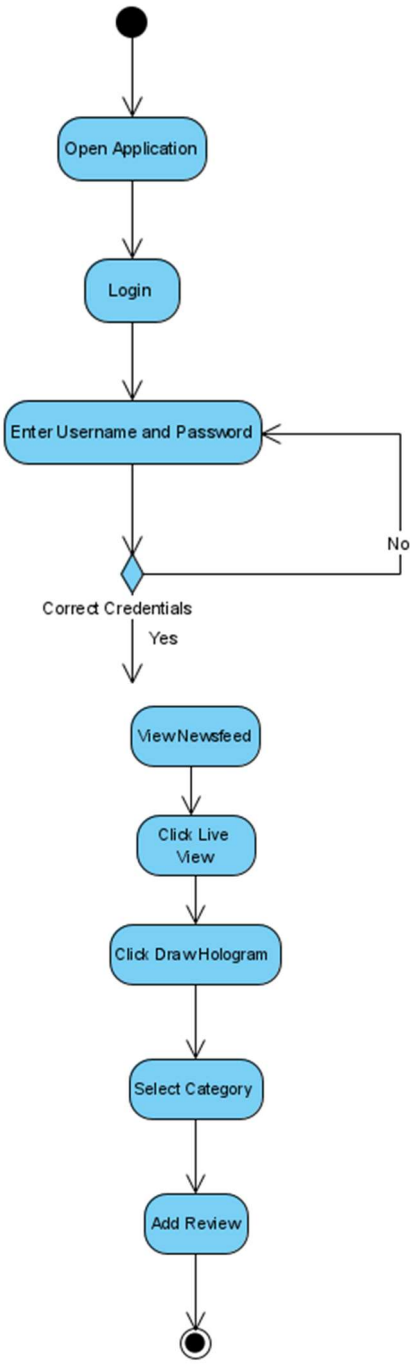
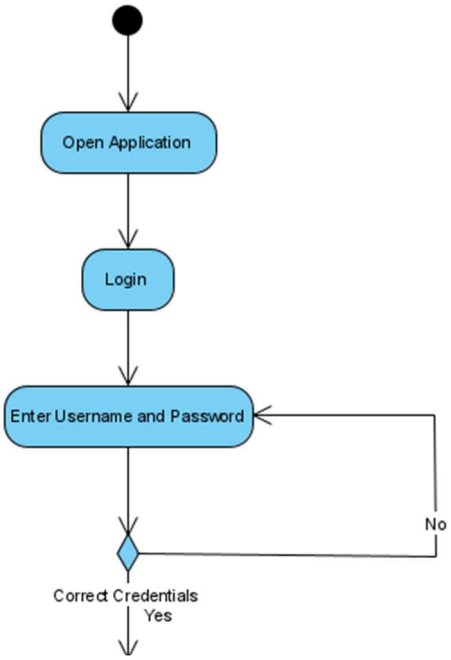


Figure 4. 28 Add Review Activity Diagram

1.5.6.5. Delete Review


Open Application
Login
Enter Username and Password
No
Correct Credentials
Yes

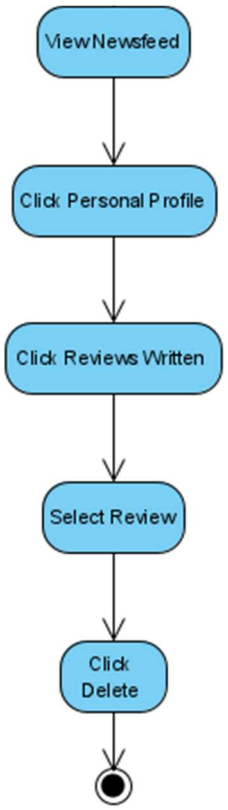


Figure 4. 29 Delete Review Activity Diagram

1.5.6.6. Create Hologram

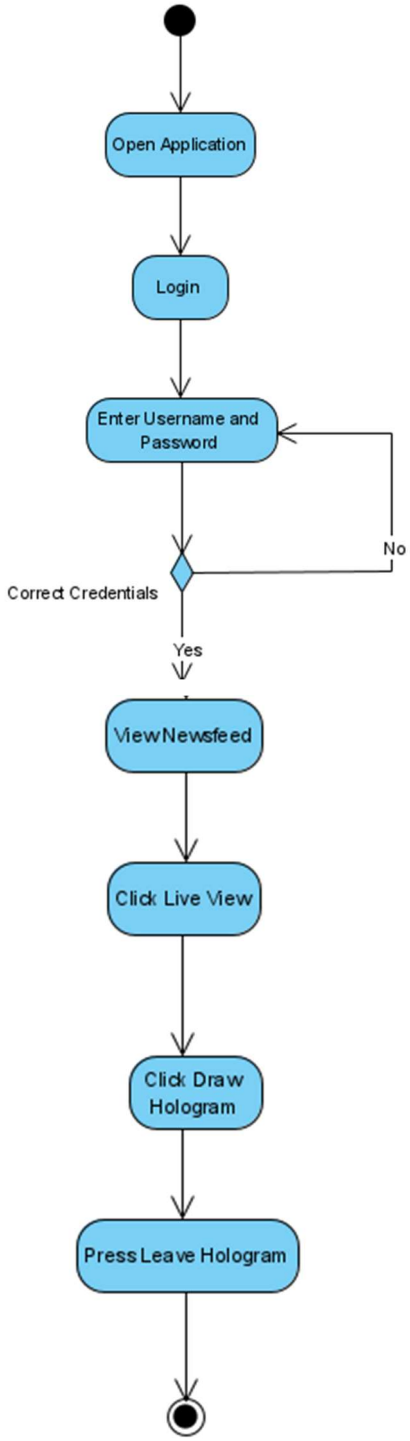


Figure 4. 30 Create Hologram Activity Diagram

1.5.6.7. Create Avatar

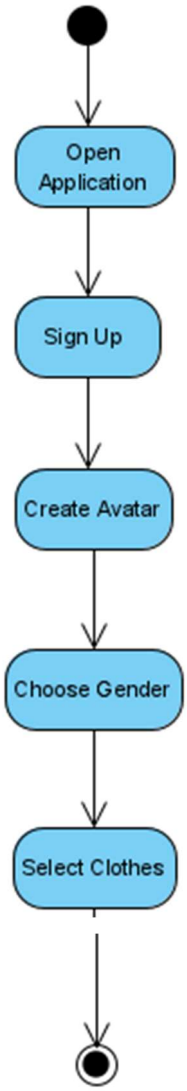


Figure 4. 31 Create Avatar Activity Diagram

1.5.6.8. Update Avatar

Figure 4. 32 Create Avatar Activity Diagram

1.5.6.9. Add Post

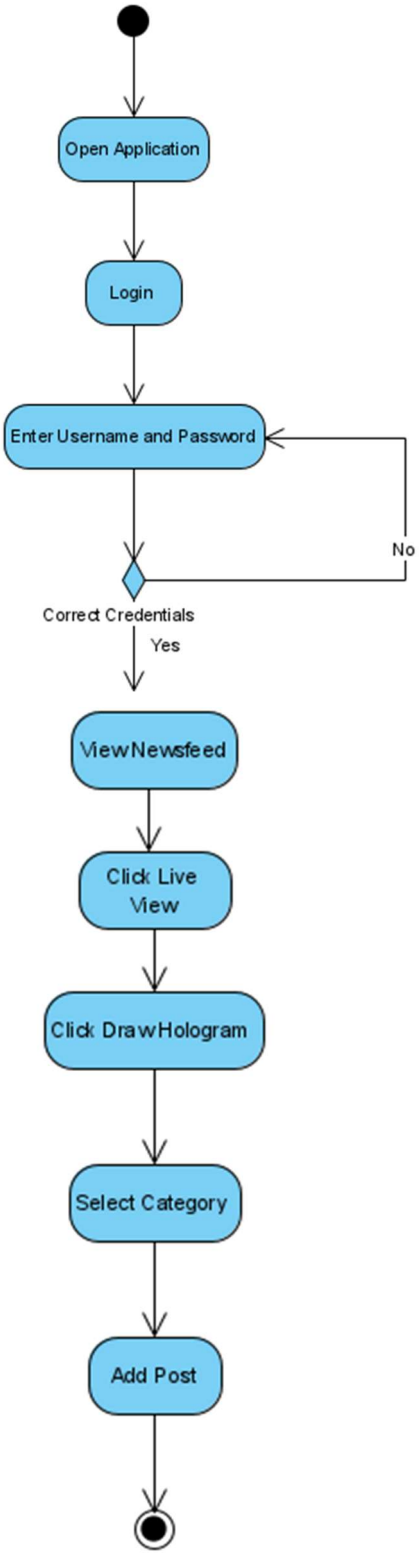


Figure 4. 33 Add Post Activity Diagram

1.5.6.10. Delete Post

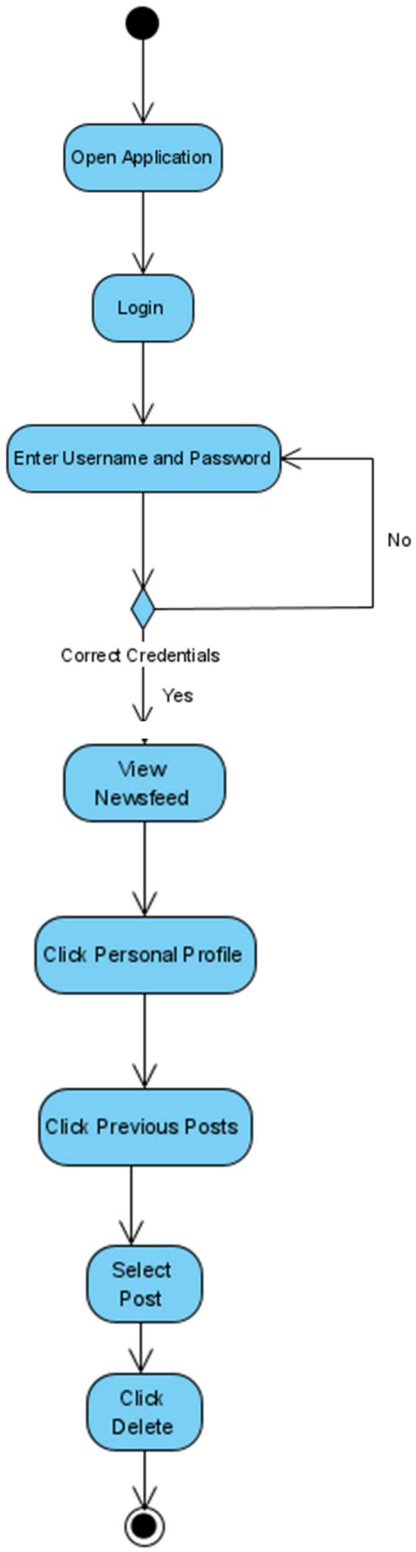


Figure 4. 34 Add Post Activity Diagram

1.5.6.11. View Personal Profile

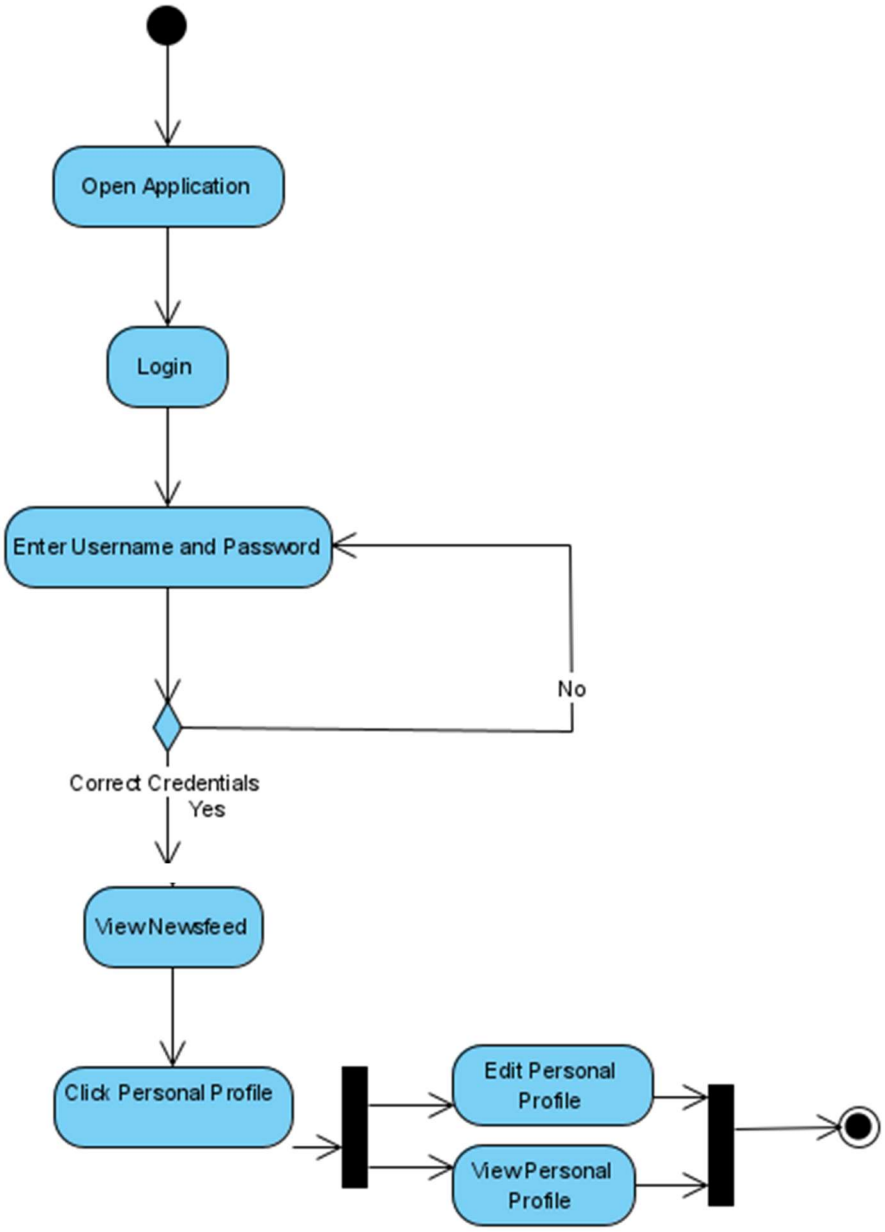


Figure 4. 35 View Personal Profile Activity Diagram

1.5.6.12. Delete Hologram Anchor

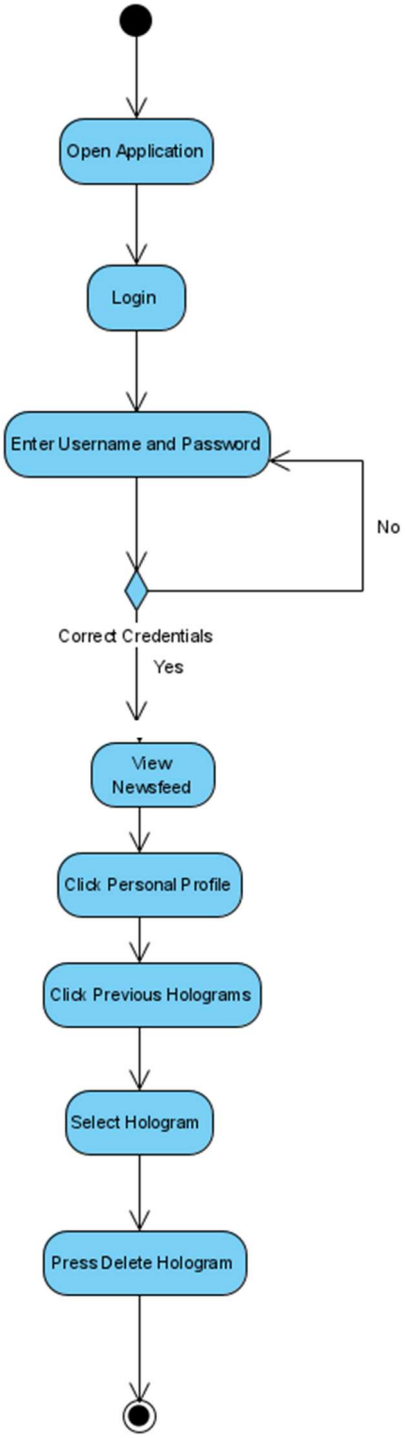


Figure 4. 36 Delete Hologram Anchor Activity Diagram

1.5.6.13. View Anchored hologram

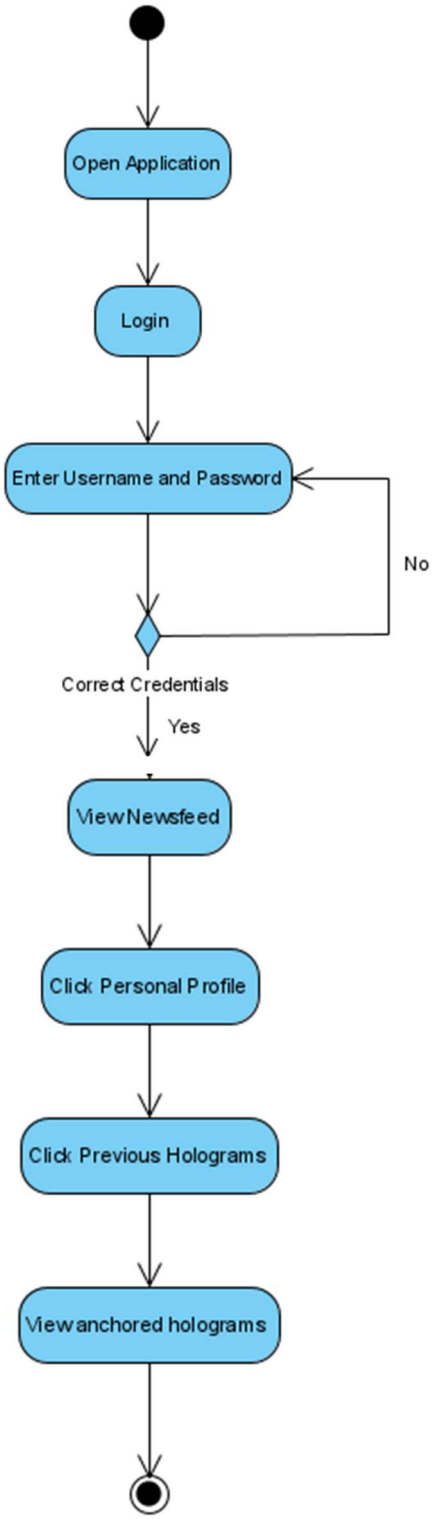


Figure 4. 37 View Anchored Hologram Activity Diagram

1.5.6.14. Newsfeed

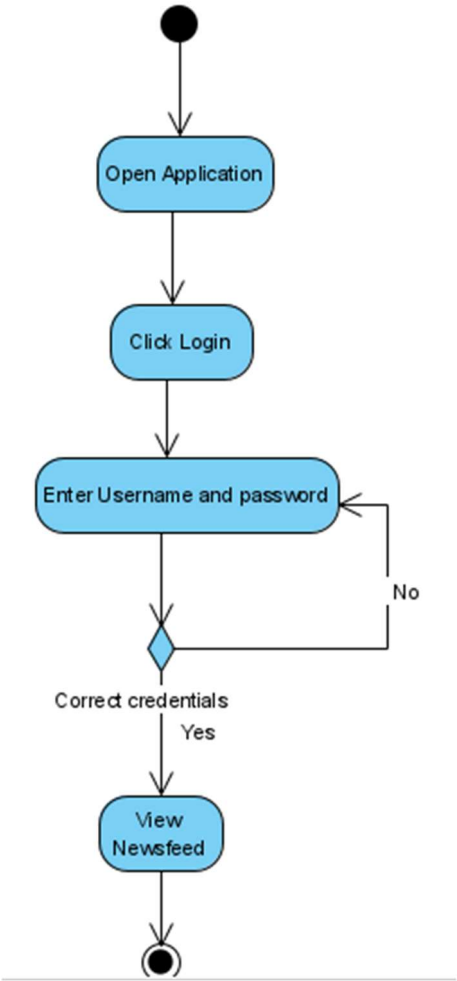
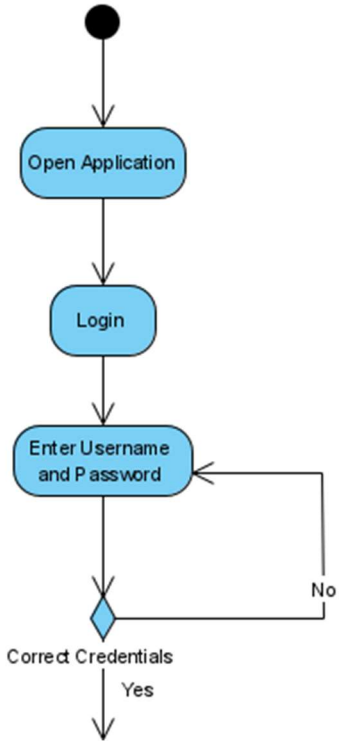


Figure 4. 38 Newsfeed Activity Diagram

1.5.6.15. Chat


Open Application
Login
Enter Username and Password
No
Correct Credentials
Yes

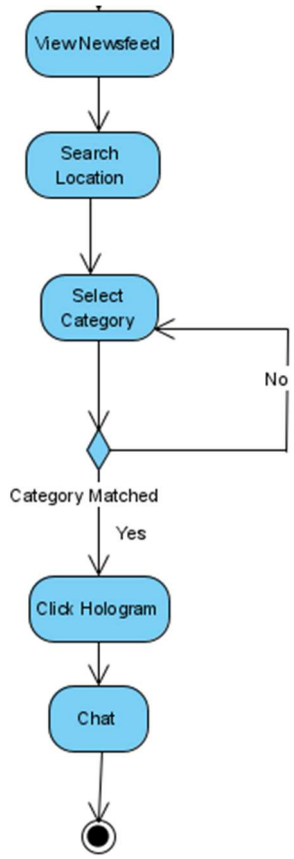
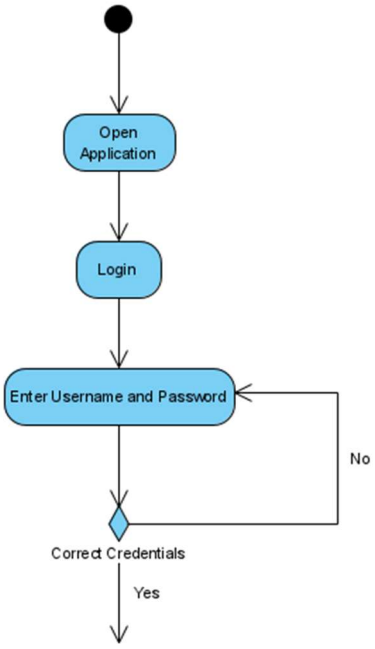


Figure 4. 39 Newsfeed Activity Diagram

1.5.6.16. Avatar Map



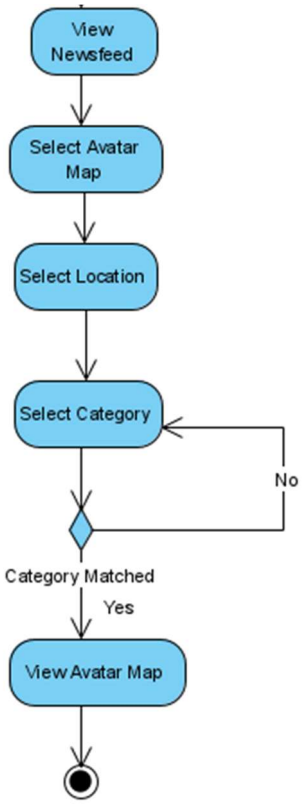
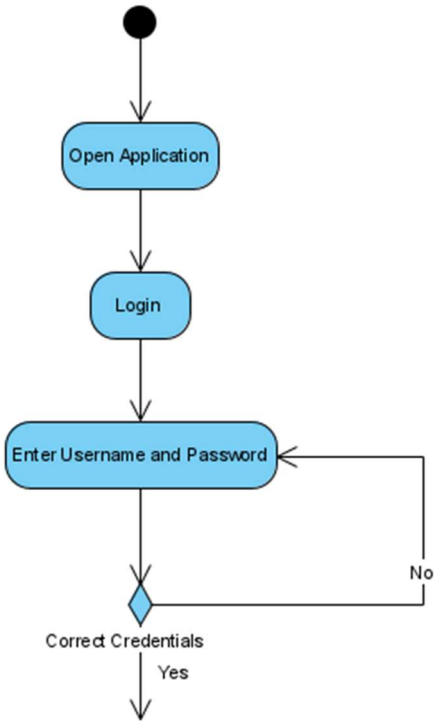


Figure 4. 40 Avatar Map Activity Diagram

1.5.6.17. Scan Location



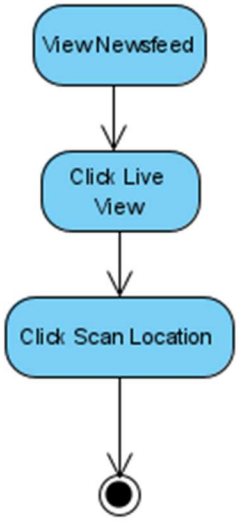


Figure 4. 41 Scan Location Activity Diagram

1.5.6.18. Sign out

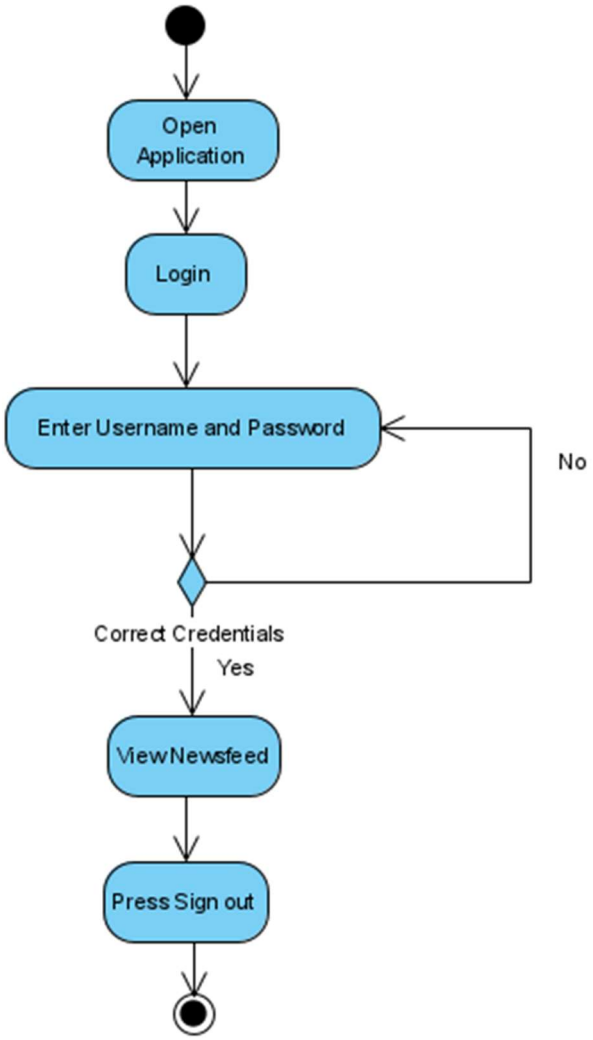


Figure 4. 42 Sign out Activity Diagram

1.5.6.19. Admin

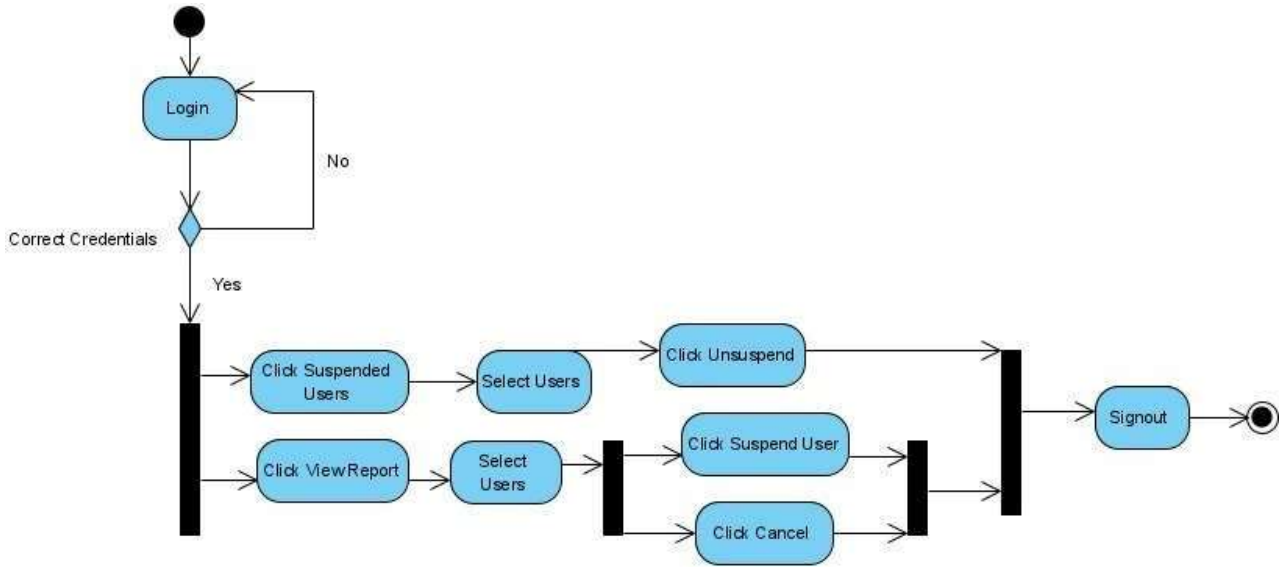


Figure 4. 43 Admin Activity Diagram

4.6. User Interface

The user Interface of the Flamingo – A Smart Guide is as follows:

4.6.1. Sign up

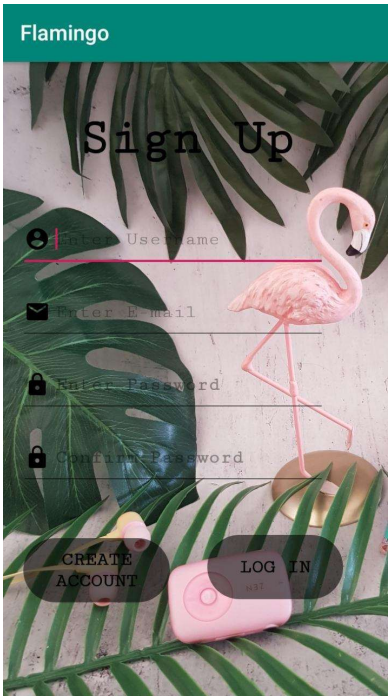


Figure 4. 44 Sign up Screen

4.6.2. Login

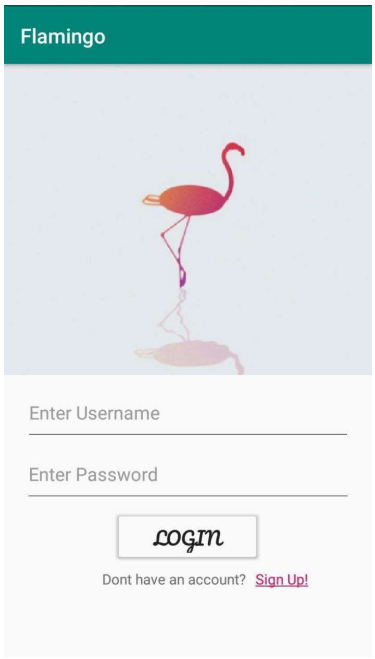


Figure 4. 45 Login Screen

4.6.3. Newsfeed

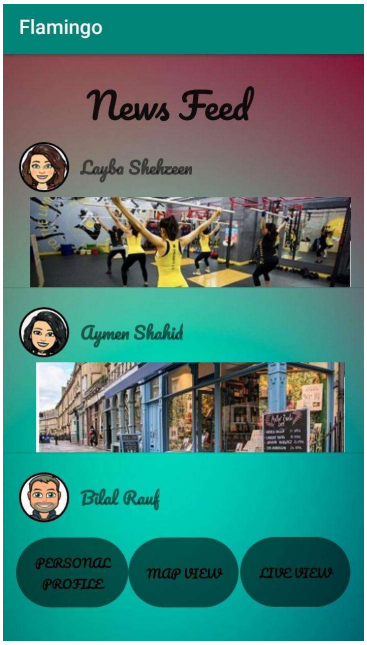


Figure 4. 46 Newsfeed Screen

4.6.4. Personal Profile

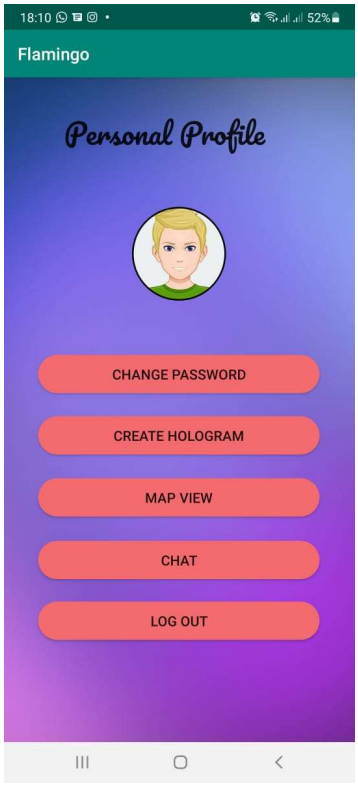


Figure 4. 47 Personal Profile Screen

4.6.5. Update Password

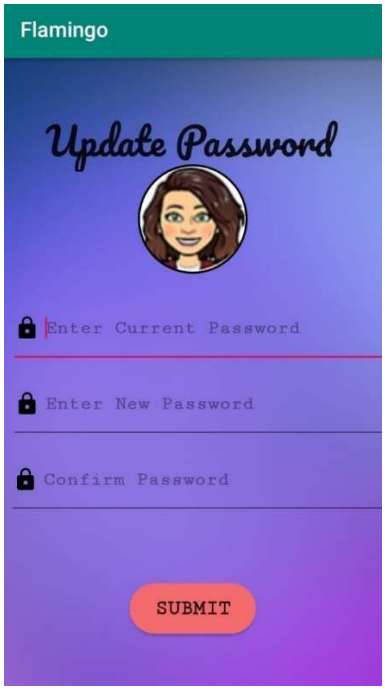


Figure 4. 48 Update Password Screen

4.6.6. Map View

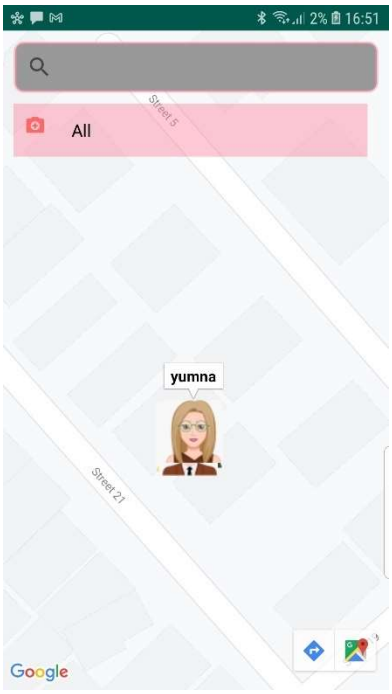


Figure 4. 49 Map View Screen

4.6.7. Live View

Figure 4. 50 Live View Screen

4.6.7.1. Review

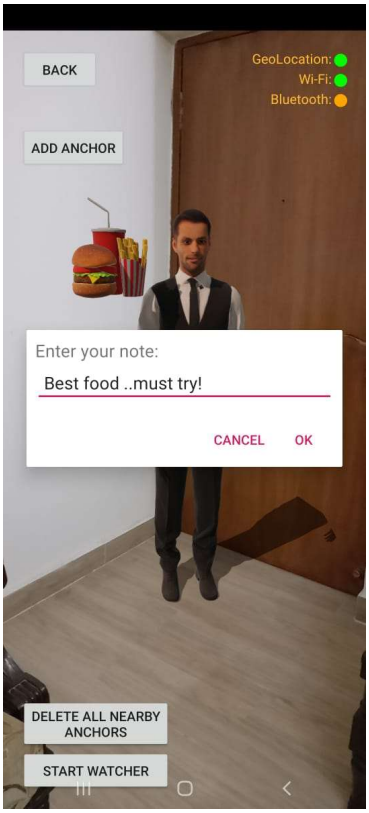


Figure 4. 51 Review Screen

4.6.8. Pick A Category

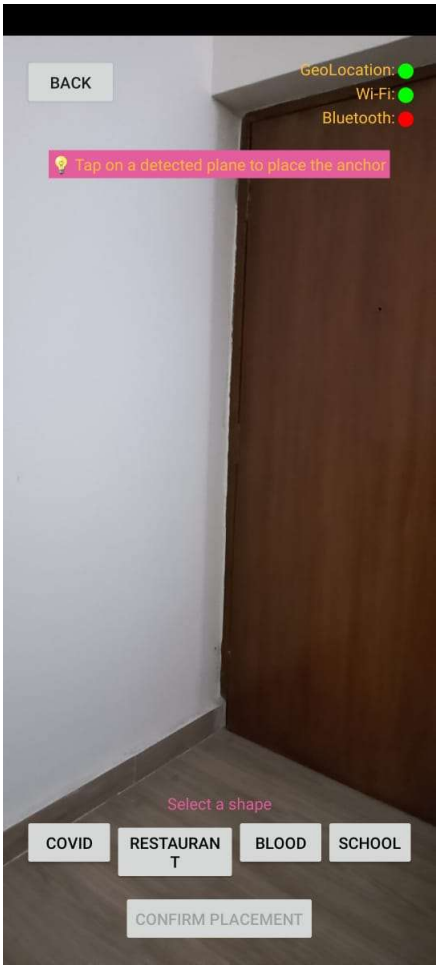


Figure 4. 52 Category Screen

4.6.9. Chat

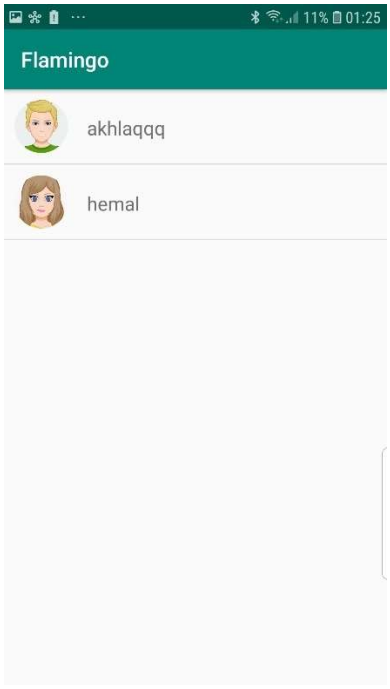


Figure 4. 53 Chat Screen

4.6.10. Admin Panel

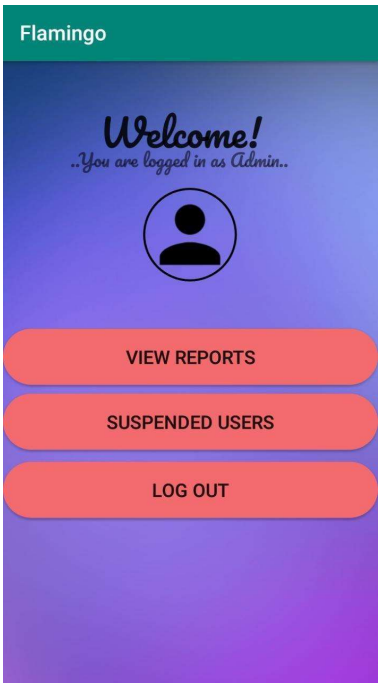


Figure 4. 54 Admin Panel Screen

4.7. System Architecture

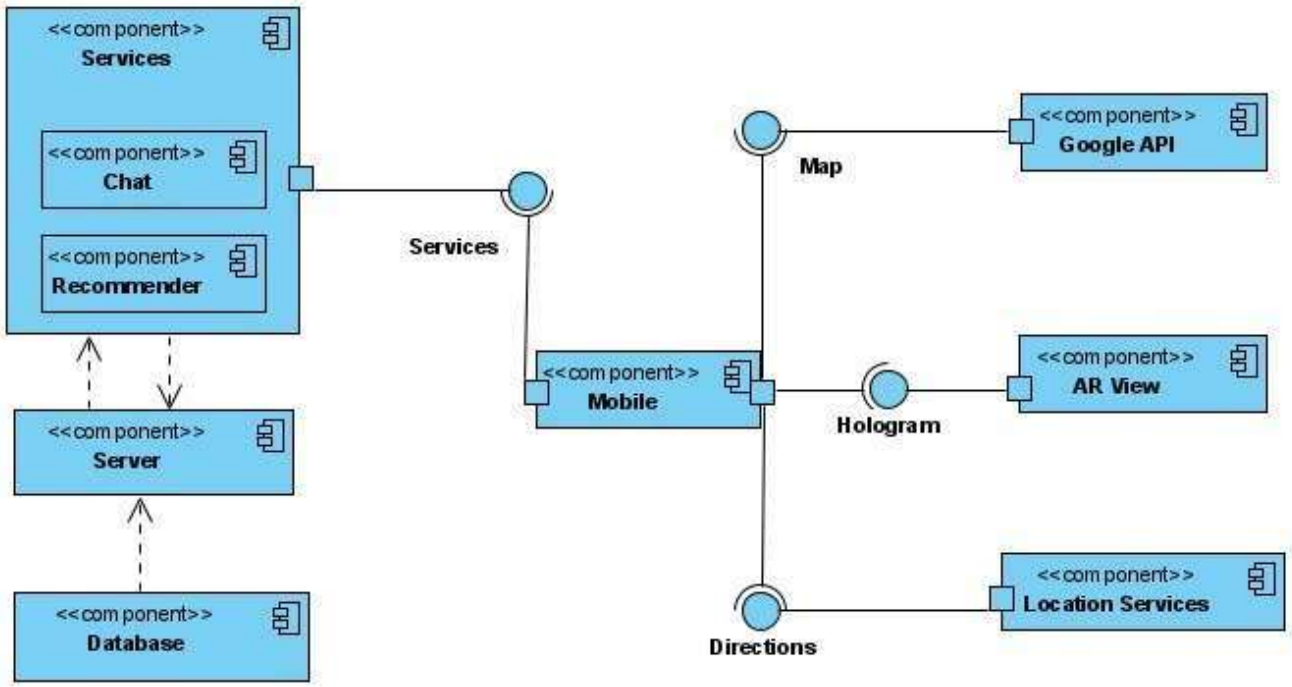


Figure 4. 55 Component Diagram

4.7.1. Architectural Design

The architectural design of Flamingo is Model-View-Controller Architecture. Model-View-Controller divides the system into the following modules to achieve the complete functionality.

1) **User interaction system-View**

   The user interacts with the user interface system through signup or login.

2) **AI based recommendation system-Controller**

   The user will get recommendations based on previous visited locations.

3) Database- Model

It stores the holograms, Location of the hologram and the application and user data.

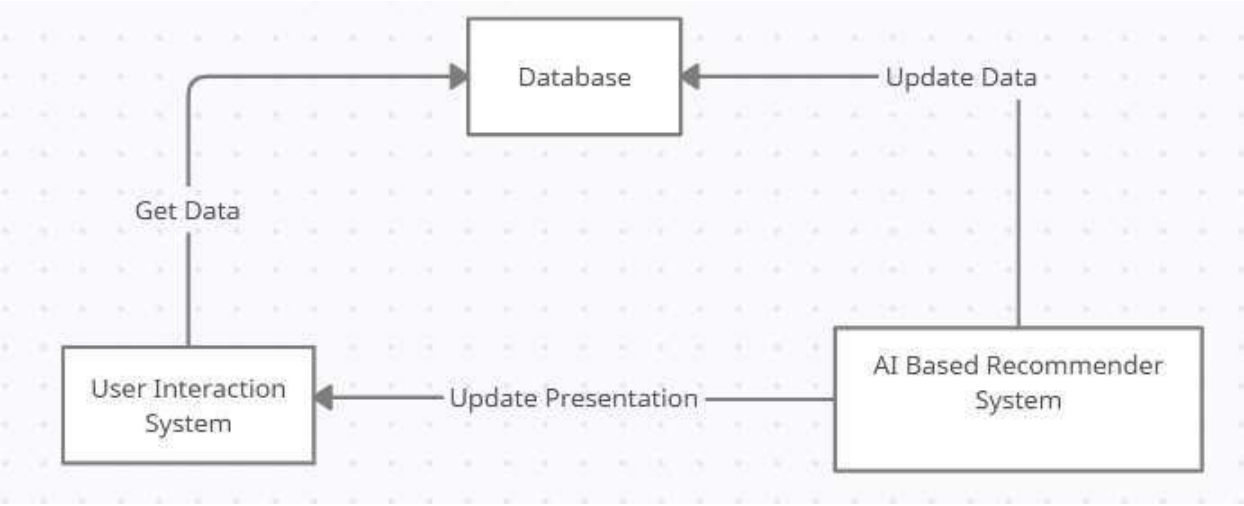


Figure 4. 56 MVC Architecture

CHAPTER: 5

SYSTEM IMPLEMENTATION

2. SYSTEM IMPLEMENTATION

Flamingo is a mobile application that requires the user login or sign up(for new users) to the application when first opened. For existing users, Login requires their Username and Password. For new users Sign Up requires a Username, email, password and then the confirmed password. The user then sets their avatar. It then takes the user to the Newsfeed interface where the user can view activity of other users, interact with them by chatting to get to know more about things of their interest. They can also choose to create their own hologram and review different places. They may also choose to go to the Map View option where they can see the avatars of other people on map and tap on them see the reviews left by those people. User can also select Personal Profile option to see the holograms and reviews they have left.

2.1. Pseudo code

2.1.1. Sing up

User opens the application.

Application displays a login and sign up button.

User clicks the sign-up button.

A prompt is displayed to input sign up information.

User enters information and click sign up button.

User is signed up to application and information is stored in database server. The avatar screen is displayed.

2.1.2. Login

User opens the application.

Application displays a login and sign up button.

User clicks the login button.

5.1.3. Update Password

User logs into the application.

Application displays newsfeed page with settings button.

User clicks the settings button

Options are displayed including change password.

User clicks change password button.

5.1.4. Add Review

User anchors the hologram to a location.

User is prompted to add review of the location.

User writes review and clicks submit button.

System records review in database and displays live view page.

5.1.5. Delete Review

User logs in and clicks on personal profile.

Options including 'reviews written' are displayed.

User clicks on reviews written option.

System displays all review written by the user.

User selects a review.

A delete or update option is displayed.

User clicks delete button.

5.1.6. Create Hologram

User logs in and clicks on live view.

Options including 'Draw Hologram' are displayed.

User clicks on 'Draw hologram' option.

System displays a screen for user to draw hologram.

User draws the hologram on screen.

A leave hologram button is displayed along with try again button.

User clicks on leave hologram button.

System stores the location and shape of hologram in the respective databases and redirects user to live view page.

5.1.7. Create Avatar

User Signups

System displays a screen for user to create avatar.

User creates avatar by choosing multiple options on screen.

5.1.8. Update Avatar

User logs in and clicks on personal profile.

Options including 'Update Avatar' are displayed.

User clicks on 'Update Avatar' option.

User makes the wanted changes and presses 'Confirm' button.
System updates the appearance of avatar in the respective databases and redirects user to personal profile page.

5.1.9. Add Post

User logs in and presses the add post button.

System opens a camera window and displays post media button.

User takes picture or video of the location and clicks the post media button.

The post is saved in database, displayed to user in personal profile and displayed to other users.

5.1.10. Delete Post

User logs in and clicks on personal profile.

Options including 'previous posts' are displayed.

User clicks on 'previous posts' option.

System displays all posts written by the user.

User selects a post.

A delete or update option is displayed.

User clicks delete button.

The post is deleted from the database and user is redirected to personal profile.

5.1.11. View Personal Profile

User logs in and clicks on personal profile button.

Personal profile is displayed.

5.1.12. Delete Hologram Anchor

User logs in and clicks on personal profile.

Options including 'previous holograms' are displayed.

User clicks on 'previous holograms' option.

System displays all holograms anchored by the user.

User selects a hologram to remove.

A delete option is displayed.

User clicks delete button.

The hologram is deleted from the database and user is redirected to personal profile.

5.1.13. View Anchored Hologram

User logs in and clicks on personal profile.

Options including 'previous holograms' are displayed.

User clicks on 'previous holograms' option.

System displays all holograms anchored by the user.

5.1.14. Newsfeed

User logs in.

Newsfeed is displayed if user is not a new one. The new user is shown newsfeed based on their personal profile.

5.1.15. Chat

User clicks on a hologram on a location.

System sends a message request to the hologram owner. If the owner accepts the request, system opens a chat window, else a rejected message is displayed.

User can type a message to another user.

The reply message from the user is displayed.

5.1.16. Avatar Map

User logs in and clicks on view map.

Avatars of other users are displayed on various locations of the map.

5.1.17. Scan Location

User logs in and clicks on live view.

Anchored holograms of other users are displayed on various locations of area.

5.1.18. Sign out

User clicks on sign out button.

User is redirected to login and signup page and session is ended.

5.1.19. Admin

User Logins.

User selects one of the options from view report, suspended users and logout.

If view report is selected, user further selects the users to suspend and cancel by clicking the suspend and cancel button.

If suspended users is selected, user further selects the users and click unsuspend.

User Sign out.

CHAPTER: 6

ANALYSIS AND EVALUATION

3. Analysis and Evaluation

3.1. Introduction

This test plan document describes the appropriate strategies, process and methodologies used to plan, execute and manage testing of “Flamingo – a Smart Guide”. The test plan will ensure that Flamingo meets the customer requirements at an accredited level. Manual Testing will be followed which includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. Each Unit will be tested separately and then will be integrated with other units; therefore, Unit Testing and Integration testing will be followed. For each unit, Black box Testing is done and for combined units Acceptance Testing is done. The test scope includes the Testing of all functional, application performance and use cases requirements listed in the requirement document. Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. This document includes the plan, scope, approach and procedure the testing of our system. The pass/fail criteria of the test items are also defined. The document tracks the necessary information required to effectively define the approach to be used in the testing of the product.

3.2. Approach

Acceptance test will be executed based on this acceptance test plan. And after all test cases are executed, a test report will be summarized to show the quality of Flamingo. Following test approaches will be used in test execution:

1. **Unit test**. Developers are responsible for unit testing. The implementation of each module and individual component will be verified separately.

2. **Integration test**. After the unit test is passed above the defined quality threshold, testers will execute the integration test cases. After all the modules are integrated, it is crucial to test the product as a black-box.

3. **Positive and negative testing design technique**. This approach will be combined with unit test and integration test. Test cases are designed in obvious scenarios, which ensure that all functional requirements are satisfied. What's more, different test cases will also be covered to show how the system reacts with invalid operations.

3.3. Features to be tested

Following Features are tested:

1. User can create an account by adding their personal information.
2. User can login to their account anytime once the account is created.
3. User can change their password as desired.
4. User can add review for a particular place where they have left their hologram .
5. User can create hologram at a place of their choice.
6. System can display recommendations for places based on user's previously visited locations.
7. User can interact with another person by messaging them.
8. User can look at avatars anchored on different locations through the avatar map.
9. User can view holograms anchored at a location by scanning their surroundings.
10. System can sort recommendations based on users' interests.

11. User can end their session by signing out.

3.4. Item Pass/Fail Criteria

Details of the test cases are specified in section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

1. Preconditions are met
2. Inputs are carried out as specified
3. The result works as what specified in output => Pass

3.5. Testing tasks

1. Develop test cases.
2. Execute tests based on the developed test cases for the software.
3. Report defects from the executed test cases if any.
4. Record test results.
5. Provide complete test report.
6. Incorporate or manage changes later in the stage of the project development.

3.6. Test Deliverables

1. Test cases
2. Output from tools

3.7. Responsibilities:

All developers of the project are responsible for the completion of all components testing and

integration testing tasks.

3.8. Staffing and Training Needs:

Basics knowledge of testing strategies and techniques is needed for the testing of the project.
Techniques such as Black Box testing, integration testing should be known to developers.

All the developers will be testing each other's work and will be actively participating in the development and testing of the project simultaneously.

3.9. Schedule

3.9.1. Important Dates

1. Unit Testing and integration testing will be finished by the start of 18 April 2021 as will Development process.
2. Acceptance Testing will be performed right after the Development process completes that is in the Mid-June.

3.10. Risks and contingencies

3.10.1. Schedule Risk:

The project might get behind schedule so in order to complete the project in time we will be needing to increase the hours/day that the project is being worked on.

3.10.2. Operational Risks:

Operational risks will be eliminated by Scheduling daily meetings and regular deadlines to meet the goals of the project as well as provide proper communication within the group.

3.10.3. Technical risks:

Technical risks will be eliminated by keeping the once defined requirements constant.

3.10.4. Programmatic Risks:

In case of a programmatic risk the scope of the project will be limited in order to stay inside the constraints of the project.

3.11. Test Cases

3.11.1. Unit and Component level Testing

Table 6. 1 Admin Login

Test Case Number	01
Test Case Name	Admin Login
Description	Admin will be able to login by giving his details and view list reported users and observe their activity and suspend them.
Testing Technique	Component testing, Black Box Testing
Preconditions	• Application should be open. • Stable internet connection.
Input Values	Enter username, password and click "Login"
Steps	1. Open the application. 2. Enter credentials. 3. Click on 'Login' button.
Expected output	Admin should be logged in.
Actual output	Admin is logged in successfully.
Status	Test case passed successfully.

Table 6. 2 Create Account

Test Case Number	02
Test Case Name	Create account
Description	User will enter his/her credentials and then select an avatar to create his account.
Testing Technique	Component testing, Black Box Testing
Preconditions	• Application should be downloaded and open. • Stable internet connection.

Input Values	Enter patient's name, Gender, Contact number, Email and password and click 'Submit form'
Steps	1. Open the application. 2. Click create account. 3. Enter username, password, email and click next. 4. Select gender and avatar. 5. Click on the 'Create Account' button.
Expected output	User should be added on database.
Actual output	User is successfully registered.
Status	Test case passed successfully.

Table 6. 3 User Login

Test Case Number	03
Test Case Name	User Login

Description	User an login into the application and view newsfeed and perform other tasks.
Testing Technique	Component testing, Black Box Testing
Preconditions	User must be registered. Application must be opened. Stable internet connection.
Input Values	Enter User's Email and password and click 'Login'
Steps	1. Open the web application. 2. Click Login. 3. Enter email. 4. Enter password. 5. Click on the 'Login button.
Expected output	User should be logged in.
Actual output	User logged in successfully.
Status	Test case passed successfully.

Table 6. 4 Update Password

Test Case Number	04
Test Case Name	Update Password
Description	Testing the update password. The user clicks update password button in personal profile. Then the user enters old password, new password and confirm password then press confirm button.
Testing Technique	Unit testing, Black Box Testing
Preconditions	User must be registered.
Input Values	Enter old password, new password and click 'Confirm'
Steps	1. Open the application. 2. Open personal Profile. 3. Click Update Password 4. Enter old password.

	5. Enter new password. 6. Click on ‘Confirm’ button.
Expected output	Password should be updated in database.
Actual output	Password updated successfully..
Status	Test case passed successfully.

Table 6. 5 Add Review

Test Case Number	05
Test Case Name	Add Review
Description	When user creates the hologram, a prompt appears, and the user enters the review.
Testing Technique	Component testing, Black Box Testing
Preconditions	User must be logged in and anchor a hologram to a specific location.
Input Values	Enter Review
Steps	1. Open Application. 2. Click hologram icon. 3. Anchor hologram to the location. 4. Enter Review.

Expected output	Review should be saved in database and appear on the note attached with the hologram.
Actual output	Review is saved in database and appears on the note attached with the hologram.
Status	Test case passed successfully.

Table 6. 6 Display Posts

Test Case Number	06
Test Case Name	Display Posts
Description	System can display recommendations for places based on user's previously visited locations.
Testing Technique	Component testing, Black Box Testing
Preconditions	User should be logged in.
Input Values	User should have an account. User should be logged in.
Steps	1. User logs in. 2. Newsfeed is displayed
Expected output	Newsfeed should be displayed.
Actual output	Newsfeed is displayed on screen.
Status	Test case passed successfully.

Table 6. 7 Chat

Test Case Number	07
Test Case Name	Chat
Description	User can interact with another person by messaging them
Testing Technique	Component testing, Black Box Testing
Preconditions	User is logged in.

	User selects another user to chat with them.
Input Values	Click on Hologram or avatar and then on the name of person to chat with them.
Steps	1. User clicks on a hologram on a location. 2. System sends a message request to the hologram owner. If the owner accepts the request, system opens a chat window, else a rejected message is displayed. 3. User can type a message to another user. 4. The reply message from the user is displayed.
Expected output	Chat screen should open.
Actual output	Chat screen is opened.
Status	Test case passed successfully.

Table 6. 8 View Avatar Map

Test Case Number	08
Test Case Name	View Avatar Map
Description	User can look at avatars anchored on different locations through the avatar map.

Testing Technique	Component testing, Black Box Testing
Preconditions	1. User is logged in 2. User has clicked on avatar map opyopm
Input Values	Click on ‘Avatar Map button
Steps	• User logs in and • User clicks on view map. • Avatars of other users are displayed on various locations of the map.
Expected output	Avatar map should be displayed.
Actual output	Avatar map displayed successfully.
Status	Test case passed successfully.

Table 6. 9 Scan Location

Test Case Number	09
Test Case Name	Scan Location
Description	User can view holograms anchored at a location by scanning their surroundings.
Testing Technique	Component testing, Black Box Testing
Preconditions	Location Services should be enabled. Camera access should be given. User should be logged in. Watcher should be on.
Input Values	User clicks on scan location option.
Steps	1. User logs in. 2. User clicks on Scan Location. 3. Anchored holograms of other users are displayed on various locations of area.

Expected output	User should be able to view other holograms.
Actual output	User views other holograms.
Status	Test case passed successfully.

Table 6. 10 Recommendation

Test Case Number	10
Test Case Name	Recommendation
Description	System sorts recommendations based on users' interests.
Testing Technique	Component testing, Black Box Testing
Preconditions	User is logged in.
Input Values	User clicks on Newsfeed.
Steps	1. User logs in. 2. User is redirected to newsfeed. 3. User refreshes the newsfeed. 4. System decides what to recommend to user on newsfeed.
Expected output	Posts should be updated.
Actual output	Newsfeed updated successfully.
Status	Test case passed successfully.

Table 6. 11 Sign Out

Test Case Number	11
Test Case Name	Sign Out
Description	User can end their session by signing out.
Testing Technique	Component testing, Black Box Testing
Preconditions	User should be logged in.
Input Values	Click “Sign out”
Steps	1. User clicks on sign out button. 2. User is redirected to login and signup page and session is ended.
Expected output	User should be logged out..
Actual output	User is logged out successfully.
Status	Test case passed successfully.

Table 6. 12 Create hologram

Test Case Number	12
Test Case Name	Create hologram
Description	User will click the hologram icon and then on add anchor button after that he will select the category and tap on screen to place the hologram. After that user clicks on confirm placement button and create anchor button to upload anchor.
Testing Technique	Component testing, Black Box Testing
Preconditions	• Application should be open, and user should already be logged in. • Stable internet connection. • GPS connection.

Input Values	Selects the category.
Steps	1. Open the application. 2. Sign in. 3. Click on add anchor. 4. Selects category. 5. Tap on screen. 6. Click Confirm placement 7. Click create anchor..
Expected output	Anchor should be added to the database.
Actual output	Anchor is successfully uploaded.
Status	Test case passed successfully.

CHAPTER: 7

FUTURE WORK

7. FUTURE WORK

A system of this magnitude always needs continuous work to evolve. There are a lot of possible changes and additions that can be done to the system to improve its performance and functionalities. The system has been made in a modular fashion which enables integrating new features very easy.

7.1. Extended Scope

Create follow option:

One user can send follow request to another user and this will create a more private experience if user wants.

AI Recommendation:

after enough data is collected, newsfeed can display recommendations based on user's previously travelled location and interest as well as friends' interests with the help of AI.

Hardware:

Smart glasses can also be made which allow a handsfree experience so no mobile phone is needed to view objects in AR.

Expand Categories:

Other categories can also be created apart from the ones already set.

CHAPTER: 8

CONCLUSION

8. CONCLUSION

8.1. Overview

Direct communication and knowledge of firsthand experience lack in many cases. When in a new place most people get to face many barriers due to language and lack of knowledge about the experiences of previous tourists. In case of blood arrangements, most people always know someone or would have experienced the late supply of blood or lack of knowledge about blood donors which nearly costs them lives of their loved ones. While opting for school most people face difficulty specially of special children so that their child does not face any difficulty in education or with the environment.

Google AR Core platform gives developers an SDK to build Augmented Reality applications so that the user can get an occlusion effect with correct light estimation and creating an immersion effect which makes user feel like the virtual objects are really placed in the world. Augmented Reality increases the level of interactivity with user which is more appealing than other social media applications.

Flamingo is a social media application that uses AR to connects people around the world. People interact by anchoring the holograms and attaching a review or posting a video with it. Be it reviews for a restaurant, theaters, cinemas, or a blood donor at a specific blood bank all you need to do is scan your phones' camera to look for holograms or see on maps while sitting at home and you can connect to anyone with one single tap. Even while scrolling your newsfeed if you come across an intriguing post user can check the place it is anchored to and get the distance and directions to that place.

8.2. Objectives Achieved

The Project helped to achieve the objectives of learning software development cycle, Augmented Reality, Azure Spatial Anchors, App Development, handling Exceptions and database Development and Design. It also helped us understand what problems are faced when developing a project in the industry, in exploring domains and getting insights on the trending technologies of the future.

CHAPTER: 9

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] https://developer.android.com/studio

[2] https://developers.google.com/ar

[3] https://azure.microsoft.com/en-us/services/spatial-anchors/

[4] https://firebase.google.com/docs

[5] https://developers.google.com/sceneform/develop/import-assets

APPENDIX A

USER MANUAL

Table of Contents

General Information

1 GENERAL INFORMATION

This section explains in general terms the system **Flamingo-A Smart Guide** and the purpose for which it is intended.

1.1 System Overview:

Flamingo is a social media application for anyone who wishes to connect to people from around the world. It will help users in various ways such as for arrangement of blood, for getting reviews of a restaurant, for deciding schools for children and for arranging plasma for covid. Every user can create holograms and click on other users' holograms to chat with them to attain their purpose. If user is at home, they can view holograms as avatars in the avatar map and chat with them from the comfort of their home. Newsfeed will display posts of users who create holograms and make videos of those holograms.

1.2 Organization of the manual:

The user's manual consists of five sections: General Information, System Summary, Getting Started, Using the System.

1.2.1 **General Information** section explains in general terms the system and the purpose for which it is intended.

1.2.2 **System Summary** section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

1.2.3 **Getting Started** section explains how to setup the system and configure it for the first time. The section presents briefly system's settings.

1.2.4 **Using the System** section provides a detailed description of system functions.

System Summary

2 SYSTEM SUMMARY

System Summary section provides a general overview of the system. The summary outlines the uses of the system's software requirements, system’s configuration, user access levels and system's behavior in case of any contingencies.

2.1 System Configuration:

Flamingo requires an Android phone that supports AR Core and has a camera, location services and Wifi/Mobile data enabled. Storage permission is also required.

2.2 User Access Levels:

Flamingo will be installed on android phones from the Google Play Store.

2.3 Contingencies:

In case of any errors or system crashes, the database will not be affected and user records will remain safe.

Getting Started

3 GETTING STARTED

Getting Started section explains how to configure the system and install it for the first-time use. The section also briefly presents the system's menu.

3.1 Installation:

The application can be installed from Google Play Store.

1. After installation, System has to be checked for network availability and AR compatibility.
2. It should be connected to the network.
3. Location Services should be enabled.
4. Camera permissions should be granted.

3.2 System Interface:

The first screen of the desktop application is Login

3.2.1 It will allow the patient to **Login** himself by entering his email and password.

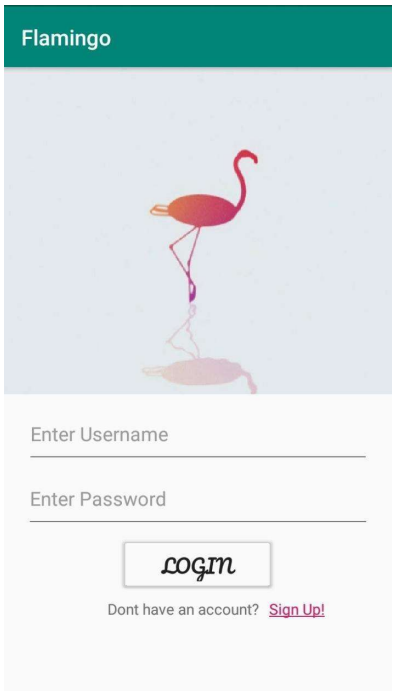


Figure 11. 1 Login Screen

3.2.1 Newsfeed

Figure 11. 2 Newsfeed Screen

3.2.2 Personal Profile

After the patient has selected personal profile the following screen will open.

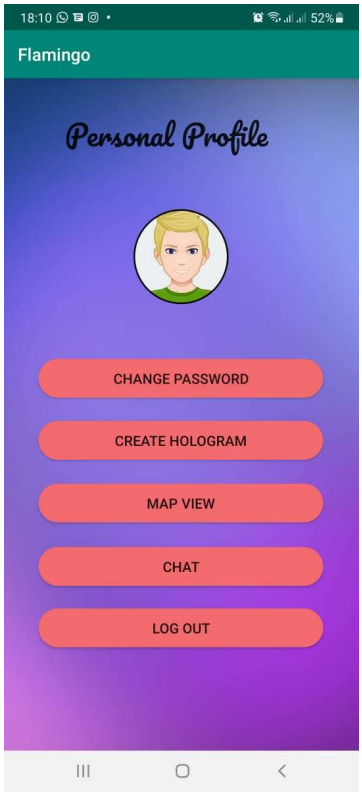


Figure 11. 3 Personal Profile Screen

3.2.3 Pick A Category

It will allow users to pick a category for the hologram.

Figure 11. 4 Pick A Category Screen

3.2.4 Create Hologram and Add Review

The user anchors hologram to a specific location interest and add a review.

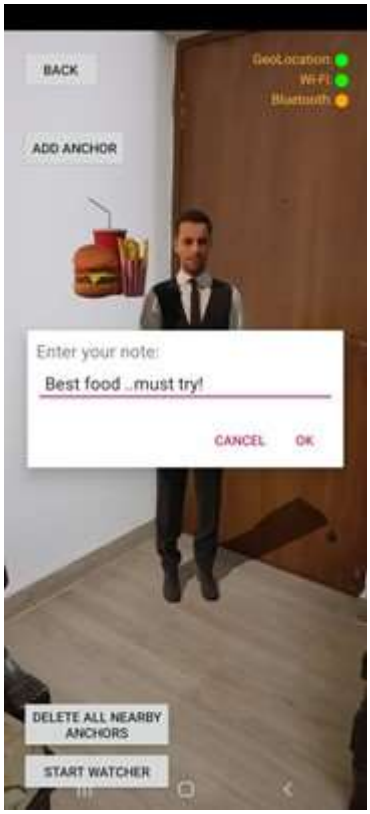


Figure 11. 5 Create Hologram and Add Review Screen

3.2.5 Scan Location

The user views the holograms anchored by others.

Figure 11. 6 Scan Location Screen

3.2.6 Avatar Map

As the user clicks the avatar map, he/she views this screen.

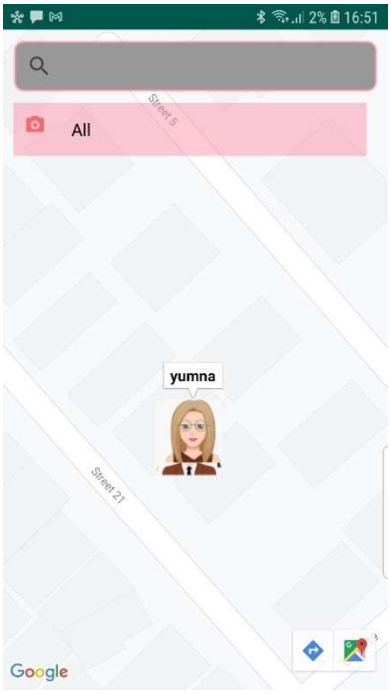


Figure 11. 7 Avatar Map Screen

3.2.7 Chat

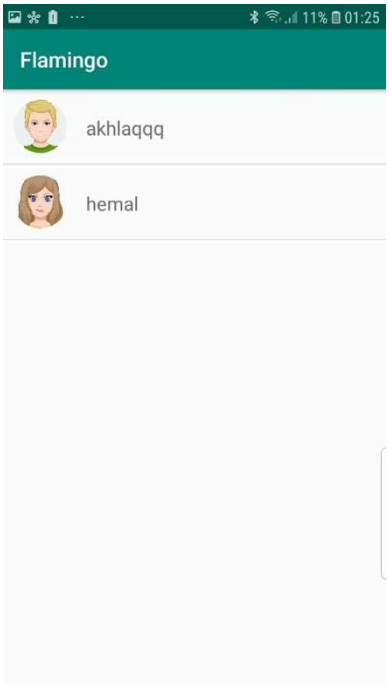


Figure 11. 8 Chat Screen

Using the System

4 USING THE SYSTEM

This section provides a description of system functions and features.

4.2 Sign up:

1. User opens the application.
2. Application displays a login and sign up button.
3. User clicks the sign-up button.
4. A prompt is displayed to input sign up information.
5. User enters information and click sign up button.
6. User is signed up to application and information is stored in database server. The avatar screen is displayed.

4.3 Login:

1. User opens the application.
2. Application displays a login and sign-up button.
3. User clicks the login button.
4. A prompt is displayed to input login credentials.
5. User enters username and password and clicks login button.
6. If credentials are correct, newsfeed page will be rendered, else error message is displayed.

4.4 Update Password:

1. User logs into the application.
2. Application displays newsfeed page with settings button.
3. User clicks the personal profile button.
4. Options are displayed including change password.
5. User clicks change password button.
6. User is prompted to add current password twice and new password once.

7. User enters the desired credentials and clicks update password button.
8. Success message is displayed, and database is updates.

4.5 Add Review

1. User anchors the hologram to a location.
2. User is prompted to add review of the location.
3. User writes review and clicks submit button.
4. System records review in database and displays live view page.

4.6 Delete Review

1. User logs in and clicks on personal profile.
2. Options including ‘reviews written’ are displayed.
3. User clicks on reviews written option.
4. System displays all review written by the user.
5. User selects a review.
6. A delete or update option is displayed.
7. User clicks delete button.
8. The review is deleted from the database and user is redirected to personal profile.

4.7 Create Hologram

1. User logs in and clicks on live view.
2. Options including ‘Draw Hologram’ are displayed.
3. User clicks on ‘Draw hologram’ option.
4. System displays a screen for user to draw hologram.
5. User draws the hologram on screen.
6. A leave hologram button is displayed along with try again button.
7. User clicks on leave hologram button.
8. System stores the location and shape of hologram in the respective databases and redirects user to live view page.

4.8 Create Avatar

1. User logs in and clicks on live view.
2. Options including ‘Draw Hologram’ are displayed.
3. User clicks on ‘Draw hologram’ option.
4. System displays a screen for user to draw hologram.
5. User draws the hologram on screen.
6. A leave hologram button is displayed along with try again button.
7. User clicks on leave hologram button.
8. System stores the location and shape of hologram in the respective databases and redirects user to live view page.

4.9 Update Avatar

1. User logs in and clicks on personal profile.
2. Options including ‘View Avatar’ are displayed.
3. User clicks on ‘View Avatar’ option.
4. Buttons including ‘Update Avatar’ are displayed.
5. User clicks ‘Update Avatar’ button makes the wanted changes and presses ‘Confirm update’ button.
6. System updates the appearance of avatar in the respective databases and redirects user to personal profile page.

4.10 Add Post

1. User logs in and presses the add post button.
2. System opens a camera window and displays post media button.
3. User takes picture or video of the location and clicks the post media button.
4. The post is saved in database, displayed to user in personal profile and displayed to other users.

4.11 Delete Post

1. User logs in and clicks on personal profile.
2. Options including ‘previous posts’ are displayed.
3. User clicks on ‘previous posts’ option.
4. System displays all posts written by the user.
5. User selects a post.
6. A delete or update option is displayed.
7. User clicks delete button.
8. The post is deleted from the database and user is redirected to personal profile.

4.12 View Personal Profile

1. User logs in and clicks on personal profile button.
2. Personal profile is displayed.

4.13 Delete Hologram Anchor

1. User logs in and clicks on personal profile.
2. Options including ‘previous holograms’ are displayed.
3. User clicks on ‘previous holograms’ option.
4. System displays all holograms anchored by the user.
5. User selects a hologram to remove.
6. A delete option is displayed.
7. User clicks delete button.
8. The hologram is deleted from the database and user is redirected to personal profile.

4.14 View Anchored Hologram

1. User logs in and clicks on personal profile.
2. Options including ‘previous holograms’ are displayed.
3. User clicks on ‘previous holograms’ option.
4. System displays all holograms anchored by the user.

4.15 Newsfeed

1. User logs in.
2. Newsfeed is displayed on the basis of AI recommendation system if user is not a new one. The new user is shown newsfeed on the basis of their personal profile.

4.16 Chat

1. User clicks on a hologram on a location.
2. System sends a message request to the hologram owner. If the owner accepts the request, system opens a chat window, else a rejected message is displayed.
3. User can type a message to another user.
4. The reply message from the user is displayed.

4.17 Avatar Map

1. User logs in and clicks on view map.
2. Avatars of other users are displayed on various locations of the map.

4.18 Scan Location

1. User logs in and clicks on live view.
2. Anchored holograms of other users are displayed on various locations of area.

4.19 Sign out

1. User clicks on sign out button.
2. User is redirected to login and signup page and session is ended.