FoodBeast(Application)

Final Year Project

Session 2017-2021

A project submitted in partial fulfilment of the
COMSATS University Degree
of
BS in Software Engineering (CUI)

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

28 December 2020

Project Detail

Type (Nature of project)	[*] **D**evelopment [ ] **R**esearch [ ] **R**&**D**			
Area of specialization				
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	SP17-BSE-040	Hafiz Hamza Ahmed	sp17-bse-040@cuilahore.edu.pk	
(ii)	SP17-BSE-022	Saad Tariq Saadi	sp17-bse-022@cuilahore.edu.pk	
(iii)	SP17-BSE-153	Mohammad Abdullah Zafar	sp17-bse-153@cuilahore.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am **Hafiz Hamza Ahmed** S/o **Ghulam Mustafa** group leader of FYP under registration no CIIT/SP17-BSE-040/LHR at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is **13%** that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: **28-December-2020** Name of Group Leader: **Hafiz Hamza Ahmed** Signature: _____

Name of Supervisor: **Amaid Zia** Co-Supervisor (if any): **Mohsin Mehdi**

Designation: Lecturer Designation: Lecturer

Signature: _____ Signature: _____

HoD: _____ Signature: _____

Abstract

Welcome to FoodBeast! A mobile application that gives you a live feed of the kitchen where the food is being prepared, a social media platform for foodies where they can post reviews, place orders by choosing from a vast range of their favourite cuisines. FoodBeast enables their users to add their food buddies to their profile and post whatever food related thoughts or media they have on their feeds. Are you really crazy about food? Then write reviews and post food related media. It allows foodbuddies to chat via built in chat.

Table of Contents

List of Tables

List of Figures

1. Chapter 1: Introduction

1.1 Introduction

Normal users can sign up on the application. They can see all the posts on the feed related to food. He can add food buddies and can check-in to the different restaurants. The user will get the feature of posting videos on feed which can result in influencing the users. The users can place an order through our application. Users can rate the Restaurants from which they have ordered food. Users will be able to view livestream of restaurants.

Restaurant will have the same login page as customers but they can't log in until approved by the admin. Restaurants can do all the things that a Normal User can do with the exception of giving reviews to other restaurants and is not able to check-in. The added feature that restaurants has is of hosting their menu on the app. Restaurants can modify their menu as they see fit.

Admin will log in through the admin web portal interface, the admin will have the power to add or remove any Restaurant or Normal User. Admins will be able to add other admins. Admin can delete posts made by any user or Restaurant.

1.2 Objectives

- **Convenience**
    - One of the main goals of our application is to provide the user with convenience, this is done by providing a multitude of cuisines on one platform from which the user can choose any and place an order through our Application. It also aims to provide convenience in the way that the user will be able to track his/her order through use of maps.
- **Security**
    - Our application will be completely secure, this will be done by following various security protocols and measures. We aim to provide our users with peace of mind that their data will be secure.

- **Performance**
    - We will ensure the smooth functioning of our application on a multitude of hardware and hardware combinations. Our app will aim to consume as little hardware resources as possible.

- **Unique Experience**
    - One of our main goals is to transform the ordinary online food ordering experience into a much more interesting ordeal. We will ensure that the user has fun and is immersed in the experience by providing food feed, food buddy options, posting of videos and we will keep our users engaged by Gamification.

1.3 Problem Statement

There is a big gap in the country to have a social platform which revolves around the category which is food. There is no local player right now who is effectively working into restaurant reviews and order system along with a social platform which will aim in assisting of establishing a food community in the country. The lack of data available regarding reviews of the restaurant services creates problem for the customers to short list restaurant with the best services in the business. People can't see the kitchen in which the food is being prepared and can't play games while waiting for their order.

FoodBeast is the one stop solution to provide user friendly experience to every foodie out there with our multiple features to save their time and energy. With few clicks, the users can easily make their decision of going out to the place of their food interest. You can find a buddy to discuss mutual food interest and let them know of your recent food experience by writing reviews and sharing posts to the community feed. You can see the kitchens on the live stream functionality.

1.4 Assumptions and Constraints

1.4.1 Assumptions

The assumptions for the project are as follows:

- Availability of Internet to use the application.
- User must know how to use a smart phone.
- User must know English Language to follow instructions.
- User knows how to interact with the application.
- User knows how to provide information regarding their interest.

1.4.2 Constraints

The constraints for the project are as follows:

- Availability of the internet is necessary.
- User access is limited to only it's profile customization and feed activity
- Application is based on English language.
- User can't post anything going against community guidelines.

1.5 Project Scope

The urge to learn new technologies and solve the problem we see every day in this era of technological advancements gives us immense power to make this project efficient and successful. We as a team, eager to learn new advancement and have skills to complete anything comes in our way. The main work ethic of our team work is to provide the food fanatics a user friendly interface with every possible feature they are looking for before going out or ordering food. The review feature will lead them to understand what the restaurant is providing and how does the experience of other people went with them. We are simply saving time of the people in this hectic routine. The teamwork and skills needed develop all these new feature into the market with beautiful design and easy going interface is our top priority. We will allocate our time and resources to this project and complete it in 7 months.

In the future work, our plan is to make the application adaptable to the user preferences. The users will get the suggestion according to their interest and recent visits.

2 Chapter 2: Requirement Analysis

2.1 Literature Review

1: Yelp

Yelp is a crowd-sourced review application about different businesses started with restaurant reviews. They offer restaurant reservation, delivery and other service based products in their application for people to make their choice for dine-in or delivery based on available reviews. They have millions of reviews available on their website and application to help people make choices before eating or going out. We will provide the users to add their food buddies and share their experience on the feed about different experiences with various restaurants[1]

2: OpenTable

This app provide the reservation feature to the restaurants with their menu and reviews by the users on the application. The live feed and food buddies is our additional feature to let the users to interact with other users who have similar interests in food[2]

3: Zomato

It is an app with providing menu, reviews of restaurant and on time delivery to it's customer. It also has a feature of zomato kitchen. The live feed makes us different from them and this is like a social media platform where users can add each other and become food buddies.[3]

4: Foodpanda

The delivery app to provide the food on demand to the customers. We are different because we have food related games to play and live feed which make sure food fanatics share their food moments as featured posts in their subscriber's feeds[4]

2.2 Stakeholders list

The following are types of users that are identifiable in the context of the system:

- Users
- Developer
- Admin

The following table describes effect of user characteristics on the system's functionality.

Actor	Level of Computer Knowledge	Level of Business Knowledge	Frequency of use
Developer	Good knowledge of development of Cross platform compatibility and integration of different components	Good understanding of upcoming and advanced innovation in the language.	When required
Users	Basic use of Android application and IOS application. Have the ability to read English Basic knowledge what does application symbols depict. Must have an interest to write reviews	They may or may not know the usage very well. Can follow the steps of the usage given by admin	Daily Basis
Admin	Good knowledge of both Web and Mobile application. Good knowledge to handle database queries Have the ability to understand user	They should know the importance to handle user problems. They should be able to solve every query faced by the user.	Daily Basis

	difficulties during the usage of applications.		

Table 1 - Stakeholders

2.3 Requirements Elicitation

2.3.1 Functional Requirement

Requirement ID	Requirement Definition
FR01:	**Sign Up**
FR01-01	The application will allow the user to enter their first name, last name, email, new password, confirm password and address.
FR01-02	Application will allow the user to enter unique email.
FR01-03	The application will send the verification link to the user's unique email

Table 2-FR01

Requirement ID	Requirement Definition
FR02:	**Login**
FR02-01	The application will allow the user to send login request.
FR02-02	The application will allow the user to provide a username and a password.
FR02-03	Application will check the email and password in the database.
FR02-04	The application will prompt the user on the login page if the email and password are not correct.
FR02-05	The application will only allow those users to login who are registered on the application.
FR02-06	The application will provide the user an option of 'Forgot Password' if the user forgets their password.
FR02-07	The application will allow the user to reset their password through email.

Table 3-FR02

Requirement ID	Requirement Definition
FR03	**Customize Profile**
FR03-01	The application will allow the user to edit their profile information, bio.
FR03-02	The user will be allowed to edit their profile only if they have registered and logged in.

Table 4-FR03

Requirement ID	Requirement Definition
FR04:	**Post Video**
FR04-01	The application will allow the user to post a video on the community feed.
FR04-02	The user will be allowed to edit the caption of the video.
FR04-03	The user will be allowed to upload the video from the gallery.

Table 5-FR04

Requirement ID	Requirement Definition
FR05:	**Post Photo**
FR05-01	The application will allow the user to post the picture inside the community feed.

FR05-02	The application will allow the user to edit caption of the picture.
FR05-03	The user will be allowed to upload the photo from the gallery.

Table 6-FR05

Requirement ID	**Requirement Definition**
FR06	**Like Post**
FR06-01	The application will allow the user to like post available on their news feed
FR06-02	The user can like the post if and only if they are logged in with their account.

Table 7-FR06

Requirement ID	**Requirement Definition**
FR07:	**Access Feed**
FR07-01	The Application will allow the user to access feed if and only if the user is logged in.
FR07-02	The user can post something to the feed
FR07-03	The application will allow the user to like public post on the feed.

Table 8-FR07

Requirement ID	Requirement Definition
FR08:	**Post Review**
FR08-01	The application will allow the user to post review about some restaurant or food on the restaurant's page.
FR08-02	User will be allowed to give star ratings to restaurant or food

Table 9-FR08

Requirement ID	Requirement Definition
FR09:	**Add FoodBuddy**
FR09-01	The application will allow the user to follow other users based on their interest.
FR09-02	The user will be allowed to send messages their buddy
FR09-03	The user will be allowed to unfriend their buddy at any point of time.

Table 10-FR09

Requirement ID	Requirement Definition
FR10:	**Post Check-in**
FR10-01	The application will allow the user to select the restaurant to post check-in

Table 11-FR10

Requirement ID	Requirement Definition
FR11:	**Play Game**
FR11-01	The application will allow the user to play game.
FR11-02	The user will be allowed to select the option to continue the story in the game
FR11-03	The application will calculate the points to determine the ending.

Table 12-FR11

Requirement ID	Requirement Definition
FR12:	**View Menu**
FR12-01	The application will allow the user to select a menu to view through browsing or searching the restaurant whose menu the user wants to see.
FR12-02	The application will not allow the user to see the menu if the restaurant have not uploaded their menu.

Table 13-FR12

Requirement ID	Requirement Definition
FR13:	**Order**
FR13-01	The user will be allowed to place an order from restaurant's menu section
FR13-02	The user will select the desired food dishes from the menu and add them into the cart.
FR13-03	The user will be allowed to delete the dishes from the cart or change the quantity
FR13-04	The user will be allowed to confirm if and only if the user enter the address.
FR13-05	The user will enter the details again if not specified before to be able to place order.

Table 14-FR13

Requirement ID	Requirement Definition
FR14:	**Approve Restaurant**
FR14-01	The application will allow the admin to view the request of different restaurant
FR14-02	The admin will be allowed to view the details and verify the credentials

FR14-03	The application will allow the admin to click approve button to list the restaurant if and only if the credentials are approved
FR14-04	The application will allow the admin to click reject button if and only if credentials are declined.

Table 15-FR14

Requirement ID	**Requirement Definition**
FR15:	**Remove Profile**
FR15-01	The application will allow the admin to view all profile list.
FR15-02	The application will allow the user to remove the profile by clicking confirm if and only if the user is admin

Table 16-FR15

Requirement ID	**Requirement Definition**
FR16:	**Accept Order**
FR16-01	The application will allow the restaurant to view the order requests

FR16-02	The restaurant will be allowed to accept the order by clicking on the accept order button.

Table 17-FR16

Requirement ID	Requirement Definition
FR17:	**Cancel Order**
FR17-01	The application will allow the restaurant to select pending orders tab.
FR17-02	The restaurant will be allowed to select cancel order button by clicking on the confirm option button

Table 18-FR17

Requirement ID	Requirement Definition
FR18:	**Add Menu**
FR18-01	The application will allow the restaurant to add menu to their profile
FR18-02	The restaurant will allowed to add each item with price, name, description and picture.
FR18-03	The restaurant will be allowed to add item if and only if they provide price, name and picture of each item by clicking on add item button.

FR18-04	The application will allow the restaurant to save menu to make it live for the users.

Table 19-FR18

Requirement ID	Requirement Definition
FR19:	**Edit Menu**
FR19-01	The application will allow the restaurant to edit the menu
FR19-02	The restaurant will be prompted by an option button to edit item into the menu
FR19-03	The restaurant will be allowed to select edit item button with change option of each attribute.

Table 20-FR19

Requirement ID	Requirement Definition
FR20:	**Pending Order**
FR20-01	The application will allow the restaurant to select the pending orders tab
FR20-02	The restaurant will be able to order time and details by selecting pending order tab

Table 21-FR20

Requirement ID	Requirement Definition
FR21:	**Dispatch Order**
FR21-01	The application will allow the restaurant to select the pending orders tab
FR21-02	The restaurant will be allowed to select the order from pending orders tab.
FR21-03	The restaurant will select the dispatch order button if and only if the order is ready to be dispatched

Table 22-FR21

Requirement ID	Requirement Definition
FR22:	**View Analytics**
FR22-01	The application will allow the admin or restaurant to select analytics tab on the dashboard if and only if logged in with desired user account
FR22-02	The application will allow the restaurant to view order analytics by clicking on view analytics button if and only if restaurant have logged in.
FR22-03	The application will allow the admin to view analytics by clicking view analytics button if and only if admin have logged in.

Table 23-FR22

Requirement ID	Requirement Definition
FR23:	**Manage Stream**
FR23-01	The application will allow the user to change camera by select manage stream tab
FR23-02	The admin will be able to assign the restaurant the stream link.
FR23-03	The admin will be able to edit the stream link for the restaurant.

Table 24-FR23

Requirement ID	Requirement Definition
FR24:	**View LiveStream**
FR24-01	The application will allow the user to view LiveStream of the restaurant.

Table 25-FR24

2.3.2 Non Functional Requirments

Requirement ID:	Requirement Definition
NFR01:	**Cross Compatibility**
NFR01-01	The application would be available with native features on both android and ios
NFR01-02	The admin panel will be available on web application for every admin users

Table 26-NFR01

Requirement ID:	Requirement Definition
NFR02:	**System Performance**
NFR02-01	Average load time of the starting Android Application of the system must be less than 6 seconds.
NFR02-02	Average load time of the starting IOS Application of the system must be less than 7 seconds.
NFR02-03	Average load time of the starting WEB Application of the system must be less than 9 seconds.
NFR02-04	Average response time of any query should be less than 5 seconds within the application.
NFR02-05	Average response time of each web application should be less than 7 seconds

Table 27-NFR02

Requirement ID	Requirement Definition
NFR03:	**Security and Precautions**
NFR03-01	Every user's data is secured as every user is required to login using their own details.
NFR03-02	Unique email is assigned to each user which makes the application more secure.
NFR03-03	If someone tries to change the password, he/she will be asked to enter code sent on their email or contact number.

Table 28-NFR03

Requirement ID:	Requirement Definition
NFR04:	**Backup and Recovery**
NFR04-01	The data will remain secure in the database if the application crashes due to some reason
NFR04-02	All the data will be saved in the database in case of any problem it will be secure and can be recovered if needed
NFR04-03	Both Application and web application will be integrated in such a way with database to make it secure.

*Table 29-***NFR04**

2.3.3 Requirement Traceability Matrix

Requirements Traceability Matrix						
Project Name:		FoodBeast				
Project Description:		The project aims to develop an application that would allow users to communicate like any social media app would while simultaneously acting as an online food ordering system.				
Requirement Id	**Requirements Description**	**WBS Deliverables**	**Activity Diagram**	**Sequence Diagram**	**Use Case**	**Test Case**
FR-01	Allow the user to sign up the application.	2	AC 01	SQ 01	UC 01	2
FR-02	Allows User to login to the application	1	AC 02	SQ 02	UC 02	1
FR-03	Allows user to customize their profile	3.2.1	AC 03	SQ 03	UC 13	3
FR-04	Allows the user to post video	3.1.1.2	AC 04	SQ 04	UC 04	6
FR-05	Allows user to post photo	3.1.1.1	AC 05	SQ 05	UC 03	6
FR-06	Allows user to like a post	3.1.1.3	AC 06	SQ 06	UC 19	
FR-07	Allows user to access the feed	3.1	AC 07	SQ 07	UC 05	

FR-08	Allows the user to Post Review of a restaurant	3.1.1.4	AC 08	SQ 08	UC 06	7
FR-09	Allows user to add a foodbuddy	3.1.2	AC 09	SQ 09	UC 07	
FR-10	Allows the user to post check-in	3.1.1.5	AC 10	SQ 10	UC 11	7
FR-11	Allow the user to play a quiz game	3.6	AC 11	SQ 11	UC 10	
FR-12	Allows user to view menu	5.3	AC 12	SQ 12	UC 12	
FR-13	Allows user to order from a restaurant	4.1	AC 13	SQ 13	UC 23	4
FR-14	Allows the admin to approve restaurant	6.1	AC 14	SQ 14	UC 21	
FR-15	Allows the admin to remove a profile	6.3	AC 15	SQ 15	UC 22	
FR-16	Allows the restaurant to accept order		AC 16	SQ 16	UC 14	5
FR-17	Allows the restaurant to cancel order		AC 17	SQ 17	UC 15	5

FR-18	Allows the restaurant to add menu		AC 18	SQ 18	UC 16	
FR-19	Allows the restaurant to edit menu		AC 19	SQ 19	UC 17	10
FR-20	Allows the restaurant to view pending order		AC 20	SQ 20	UC 18	
FR-21	Allows the restaurant to dispatch order		AC 21	SQ 21	UC 20	8
FR-22	Allows the restaurant/ admin to view analytics	6.2	AC 22	SQ 22	UC 09	
FR-23	Allows the restaurant to manage Live Stream		AC 23	SQ 23	UC 08	9

Table 30-TraceabilityMatrix

2.4 Use Case Description

Use Case ID:	UC-01
Use Case Name:	Sign Up
Actors:	Restaurant and Normal User
Description:	New users will be able to register on our application.
Trigger:	
Preconditions:	2. Anyone registering should have a valid email address. 3. They should not be already registered on the app with the same email address.
Post conditions:	1. Users successfully registered and have to verify the email through link sent to provided email.
Normal Flow:	1. User launches the app. 2. User enters email. 3. User creates a password 4. User provides information like name, address, mobile number etc. 5. User checks the box for Restaurant if it is a restaurant profile. 6. User Clicks Register. 7. User is registered and Home screen is showed.
Variations: **[Alternative Flow 1 – Not in Network]**	
Exceptions:	4a. Invalid Email enter 1. Application shows error message prompting user to renter valid email. 2. User enters email again (Use case resumes on step 3 of Normal flow) 4b. Email is already registered with an account. 1. Application show message that email is already in use and prompts user to login or recover password. 2. Use case ends.

Table 31 - UC 1

Use Case ID:	UC-02
Use Case Name:	Login
Actors:	Restaurant and Normal User
Description:	Existing users will be able to login to their profiles and access the application.
Trigger:	
Preconditions:	1. User must have signed up for the application and verified the email
Post conditions:	1. User will be able to login and access the application
Normal Flow:	1. User Launches the application. 2. User presses the login button. 3. User enters email and password and presses confirm 4. User is taken to the home screen of the application.
Variations: **[Alternative Flow 1 – Not in Network]**	
Exceptions:	User can choose to close the app anytime they wish. 3a. Invalid Password is entered. 1. Application shows invalid credentials message and user is prompted to try again. 2. User tries again (Use case resumes from step 3 of Normal Flow) or he presses forget password and is prompted to enter email. 3. User enters email 4. System sends user a mail with reset password link. 3b. Invalid email is entered 1. Application prompts user to enter valid email. 2. User enters email again (Use case resumes from step 3 of the normal flow) 3c. Non-Registered email is entered 1. Application prompts user to Sign Up first by showing sign up button. 2. User enters email again (Use Case Resumes from step 3 of normal flow) or User chooses to sign up.

	3. Use Case ends. 3d. Email not verified

Table 32 - UC02

Use Case ID:	UC-03
Use Case Name:	Post Photo
Actors:	Normal User
Description:	Allows the user to post photos on his/her feed
Trigger:	
Preconditions:	1. User is logged in
Post conditions:	1. Picture is successfully posted
Normal Flow:	1. User Launches the app. 2. User accesses his feed (Refer to access feed Use Case) 3. User Clicks on the post picture button. 4. User selects picture to upload from gallery. 5. User writes a caption for the picture. 6. User presses the post now button 7. Picture is posted
Variations: **[Alternative Flow 1 – Not in Network]**	
Exceptions:	6a. User chooses to cancel posting 1. Application prompts the user to confirm if he/she is sure about canceling the post by showing a message 2. User confirms 3. Use case ends 4b. Size of media is too large 1. User is asked to choose another file

	2. Use case resumes from Step 4 of normal flow.

Table 33- UC03

Use Case ID:	UC-04
Use Case Name:	Post Video
Actors:	Normal User
Description:	Allows the user to post videos on his/her feed
Trigger:	
Preconditions:	1. User is logged in 2. User has Verified Status.
Post conditions:	1. Video is successfully posted
Normal Flow:	1. User Launches the app. 2. User accesses his feed (Refer to access feed Use Case) 3. User Clicks on the post video button. 4. User selects video to upload from gallery. 5. User writes a caption for the video. 6. User presses the post now button 7. Video is posted
Variations: **[Alternative Flow 1 – Not in Network]**	
Exceptions:	6a. User chooses to cancel posting 1. Application prompts the user to confirm if he/she is sure about canceling by showing a message 2. User confirms 3. Use case ends 4b. Size of media is too large 3. User is asked to choose another file 4. Use case resumes from Step 4 of normal flow.

Table 34-UC04

Use Case ID:	UC-05
Use Case Name:	Access feed
Actors:	Restaurant or Normal User
Description:	Enables user to access feed of the app
Trigger:	
Preconditions:	1. User is logged in
Post conditions:	1. Feed is displayed to the user
Normal Flow:	1. User launches app 2. Fresh Launch of the app will take user to home screen 3. Press the access feed button 4. Feed is shown to the user
Variations: **[Alternative Flow 1 – Not in Network]**	2a. User is not at the home screen 1. User will navigate to home screen 2. Use case resumes from step 3 of normal flow
Exceptions:	

Table 35-UC05

Use Case ID:	UC-06
Use Case Name:	Post Review
Actors:	Normal User
Description:	Allows user to post review of a restaurant
Trigger:	
Preconditions:	1. User is logged in

Post conditions:	1. Review is successfully posted
Normal Flow:	1. User launches the app 2. User goes to the restaurant profile 3. User press review button 4. User writes the review 5. User gives his/her rating out of 10 6. User confirms to post the review 7. Review is posted
Variations: **[Alternative Flow 1 – Not in Network]**	4a. User wants to just give a rating out of 10 1. Use Case step 3 is bypassed 2. Use case continues from step 5 of the normal flow
Exceptions:	4a. User exceeds word limit 1. Application shows message to user informing him/her to stay within the word limit 2. Use case continues from step 4 of the normal flow 6a. User cancels the review 1. Application prompts the user to confirm if he/she is sure about canceling by showing a message 2. User confirms 3. Use case ends

Table 36-UC06

Use Case ID:	UC-07
Use Case Name:	Add foodbuddy
Actors:	Normal User /Restaurant
Description:	Works like adding friend option on Facebook, it adds a foodbuddy to your account
Trigger:	
Preconditions:	1. User is logged in

Post conditions:	1. The user and selected profile which user was adding as foodbuddy will be made foodbuddies
Normal Flow:	1. User Launches app 2. User searches for a profile 3. User selects the profile he/she wants to be foodbuddies with 4. User presses the Add FoodBuddy button. 5. FoodBuddy is added.
Variations: **[Alternative Flow 1 – Not in Network]**	
Exceptions:	4a. Selected profile has disabled add foodbuddy option 1. Use case ends 3a. Selected profile is a Restaurant 1. Application shows a message asking user to select a user profile 2. Use case continues from step 2 of the normal flow User can choose not to add foodbuddy and just surf the app at any step

Table 37-UC07

Use Case ID:	UC-08
Use Case Name:	Manage Live stream
Actors:	Admin
Description:	Assign OR Edit stream link to/of restaurant
Trigger:	
Preconditions:	1. Admin is logged in 2. Restaurant is Approved
Post conditions:	1. Will assign/edit stream link to/of restaurant
Normal Flow:	1. Admin login into the admin panel 2. Admin selects to manage live stream 3. Admin selects restaurant to assign /edit stream link 4. Admin assigns or edits the stream link to/of restaurant 5. Admin clicks on save button.

Variations: **[Alternative Flow 1 – Not in Network]**	
Exceptions:	Restaurant is not approved Use case ends

Table 38-UC08

Use Case ID:	UC-09
Use Case Name:	View Analytics
Actors:	Admin and Restaurant
Description:	Will allow the restaurant to see the analytics of their restaurant and will allow the admin to see the analytics of the whole app
Trigger:	
Preconditions:	• Admin is logged in
Post conditions:	• Analytics of sales. will be displayed.
Normal Flow:	1. Admin accesses the Analytics portal 2. Admin decides on the analytic he she wants to view These can be: Yearly sales or monthly sales etc. 3. Analytical information and data about the selected choice is shown
Variations: **[Alternative Flow 1 – Not in Network]**	3a. Admin wants to choose a different analytic 1. Admin presses the back button 2. Use Case resumes from step 2 of the Normal Flow
Exceptions:	At any step of the use case, admin can choose to not view the analytics ending the use case

	4a. No information about the selected choice can be found 1. No information or data found message is shown and admin is prompted to make another choice 2. Use case ends

Table 39-UC09

Use Case ID:	UC-10
Use Case Name:	Play Game
Actors:	Normal User
Description:	The user will be able to play a quiz mini game
Trigger:	
Preconditions:	1. User is Logged In
Post conditions:	1. Game is displayed and is playable
Normal Flow:	1. User Launches the app 2. User presses the Play Game Button 3. A screen open with a question with multiple choice of answers 4. User selects the correct option 5. Use case returns to step 3 and follows normal flow of questions 6. Game ends after completing the story.
Variations: **[Alternative Flow 1 – Not in Network]**	4a. If the user selects wrong option 1. Point is deducted 2. Use case return to step 3 of Normal flow
Exceptions:	At any step of the use case, user can choose to stop playing by pressing leave button which will end the use case

Table 40-UC10

Use Case ID:	UC-11
Use Case Name:	Post Check In
Actors:	Normal User
Description:	Allows the user to post a check in which tells which restaurant he is at.
Trigger:	
Preconditions:	1. User is logged in
Post conditions:	1. The location of the restaurant which the user is visiting has been posted
Normal Flow:	1. User launches app 2. User accesses his/her feed 3. User presses post check-in 4. User selects the restaurant where he/she is 5. User presses Check-in Button 6. Check-in has been posted
Variations: **[Alternative Flow 1 – Not in Network]**	4a. User may choose to use GPS 1. The application will use the location to show only those restaurants which are in the close vicinity of the user 2. Use case continues from Step 5 of Normal Flow
Exceptions:	4a. The user is using GPS but that is unable to determine the location 1. User will be prompted to manually search for the restaurant 2. Use case return to Step 4 of the normal flow 5a. Caption exceeds word limit 1. Application will prompt the user to shorten the caption and stay within word limit 2. Use case returns to step 5 of the normal flow During any step of the use case, user may cancel the check-in ending the use case

Table 41-UC11

Use Case ID:	UC-12
Use Case Name:	View Menu
Actors:	Normal User
Description:	User will be able to view the menu of a Restaurant
Trigger:	
Preconditions:	1. User is logged in.
Post conditions:	1. Menu of the Restaurant is displayed
Normal Flow:	1. User Launches App 2. User navigates to Browse Restaurants section. 3. User Searches for the restaurant. 4. User Selects the restaurant he/she wants 5. User Views the Restaurant Menu.
Variations: **[Alternative Flow 1 – Not in Network]**	5a. User selects a restaurant which hasn't added a menu 1. Application shows message of Menu not found and redirects the user to search again 2. Use case returns to step 3 of the normal flow
Exceptions:	Step 1-5. User may choose to not view any menu and use case will end

Table 42-UC12

Use Case ID:	UC-13
Use Case Name:	Customize Profile
Actors:	Restaurant and Normal User
Description:	Users will be enabled to customize their profile as they want in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.

Normal Flow:	1. User access the home screen. 2. User clicks the customize profile button. 3. User make changes. 4. User clicks Save button. 5. Profile has been customized.
Alternative path	4a. User clicks Cancel button.
Post conditions:	Profile is customized successfully.
Exceptions:	3a. User enters numbers or special characters in the name section.

Table 43-UC13

Use Case ID:	UC-14
Use Case Name:	Accept Order
Actors:	Restaurant
Description:	Restaurants will be enabled to accept orders placed by the normal users in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.
Normal Flow:	1. Restaurant clicks the view order button. 2. Restaurant clicks the pending order button 3. Restaurant selects the order to accept. 4. User clicks Accept order button. 5. Order has been accepted.
Alternative path	
Post conditions:	Order has been accepted successfully.
Exceptions:	

Table 44-UC14

Use Case ID:	UC-15
Use Case Name:	Cancel Order
Actors:	Restaurant
Description:	Restaurants will be enabled to cancel orders placed by the normal users in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.
Normal Flow:	1. Restaurant clicks the view order button. 2. Restaurant clicks the pending orders button. 3. Restaurant selects the order to cancel. 4. Restaurant clicks Confirm cancel order button. 5. Order has been cancelled.
Alternative path	4a. Restaurant clicks Cancel button.
Post conditions:	Order has been cancelled successfully.
Exceptions:	

Table 45-UC15

Use Case ID:	UC-16
Use Case Name:	Add Menu
Actors:	Restaurant
Description:	Restaurants will be enabled to Add menu in their profile in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.
Normal Flow:	1. Restaurant clicks the Add Menu button. 2. Restaurant clicks the Add items button. 3. Restaurant clicks the Add items icon. 4. Restaurant Add item name, item picture, item price and item description. 5. Restaurant can again click the add item icon to add more items to menu. 6. Restaurant clicks Save button.

	7. Menu has been added to profile.
Alternative path	6a. User clicks Cancel button.
Post conditions:	Menu has been added to profile successfully.
Exceptions:	4a. Restaurant enters numbers or special characters in the name section. 4b. Restaurant enters Alphabets or special characters in price section.

Table 46-UC16

Use Case ID:	UC-17
Use Case Name:	Edit Menu
Actors:	Restaurant
Description:	Restaurants will be enabled to Edit menu in their profile in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.
Normal Flow:	1. Restaurant clicks the Edit Menu button. 2. Restaurant clicks the Add items button. 3. Restaurant Add item name, item picture, item price and item description. 4. Restaurant clicks Save button. 5. Restaurant clicks the Customize items button. 6. Restaurant make changes. 7. Restaurant clicks Save button. 8. Restaurant clicks the Delete items button. 9. Restaurant selects items to delete. 10. Restaurant clicks confirm button. 11. Restaurant clicks Save button. 12. Menu has been updated.
Alternative path	4a. User clicks Cancel button. 7a. User clicks Cancel button. 10a. User clicks Cancel button.
Post conditions:	Menu has been updated successfully.
Exceptions:	3a. Restaurant enters numbers or special characters in the name section. 3b. Restaurant enters Alphabets or special characters in price section.

Table 47-UC17

Use Case ID:	UC-18
Use Case Name:	View Pending Orders
Actors:	Restaurant
Description:	Restaurants will be enabled to view pending orders placed by the normal users in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.
Normal Flow:	1. Restaurant clicks the view order button. 2. Restaurant clicks the View pending orders button
Alternative path	
Post conditions:	View pending orders successfully.

Table 48-UC18

Use Case ID:	UC-19
Use Case Name:	Like Posts
Actors:	Restaurant and Normal User
Description:	Restaurant and Normal Users will be enabled to like posts in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.
Normal Flow:	1. Users clicks the Feed button. 2. Users view post. 3. Users click Like button
Alternative path	
Post conditions:	Post liked successfully.

Exceptions:	

Table 49-UC19

Use Case ID:	UC-20
Use Case Name:	Dispatch Order
Actors:	Restaurant
Description:	Restaurants will be enabled to dispatch orders placed by the normal users in the system.
Preconditions:	1. They should have a profile. 2. They must be logged in in the system.
Normal Flow:	1. Restaurant clicks the view order button. 2. Restaurant clicks the Accepted orders button. 3. Restaurant selects the order to dispatch. 4. Restaurant clicks Confirm dispatch order button. 5. Order has been dispatched.
Alternative path	4a. Restaurant clicks go back button.
Post conditions:	Order has been dispatched successfully.
Exceptions:	

Table 50-UC20

Use Case ID:	UC-21
Use Case Name:	Approve Restaurant
Actors:	Admin
Description:	Admin will be enabled to approve restaurants in the system.
Preconditions:	1. User should be an admin. 2. They must be logged in in the system.

Normal Flow:	1. Admin clicks the view request button. 2. Admin selects the restaurant. 3. Admin verifies the credentials. 4. Admin clicks approved if verified. 5. Admin clicks reject if not verified.
Alternative path	
Post conditions:	Restaurant has been approved successfully.
Exceptions:	

Table 51-UC21

Use Case ID:	UC-22
Use Case Name:	Remove Profile
Actors:	Admin
Description:	Admin will be enabled to remove profiles in the system.
Preconditions:	3. User should be an admin. 4. They must be logged in in the system.
Normal Flow:	6. Admin clicks the Account removal button. 7. Admin selects the profile. 8. Admin clicks remove profile button. 9. Admin clicks confirm.
Alternative path	
Post conditions:	Profile has been removed successfully.
Exceptions:	

Table 52-UC22

Use Case ID:	UC-23
Use Case Name:	Order

Actors:	Normal User
Description:	User will be able to order food from a restaurant.
Trigger:	
Preconditions:	1. User is logged in 2. User is viewing menu (View Menu Use Case, ID is UC-12)
Post conditions:	1. Food from restaurant is successfully ordered.
Normal Flow:	1. User selects all the items he/she wants to order. 2. User Proceeds to Checkout screen 3. User confirms order
Variations: **[Alternative Flow 1 – Not in Network]**	2a. User has not specified address in his/her profile 1. A pop up box appears which allows the user to enter address 2. Use Case resumes from step 6 of the normal flow.
Exceptions:	User has not provided the address.

Table 53-UC23

2.5 Use Case Designs

2.5.1 Whole System Use Case Design (UCD-01)

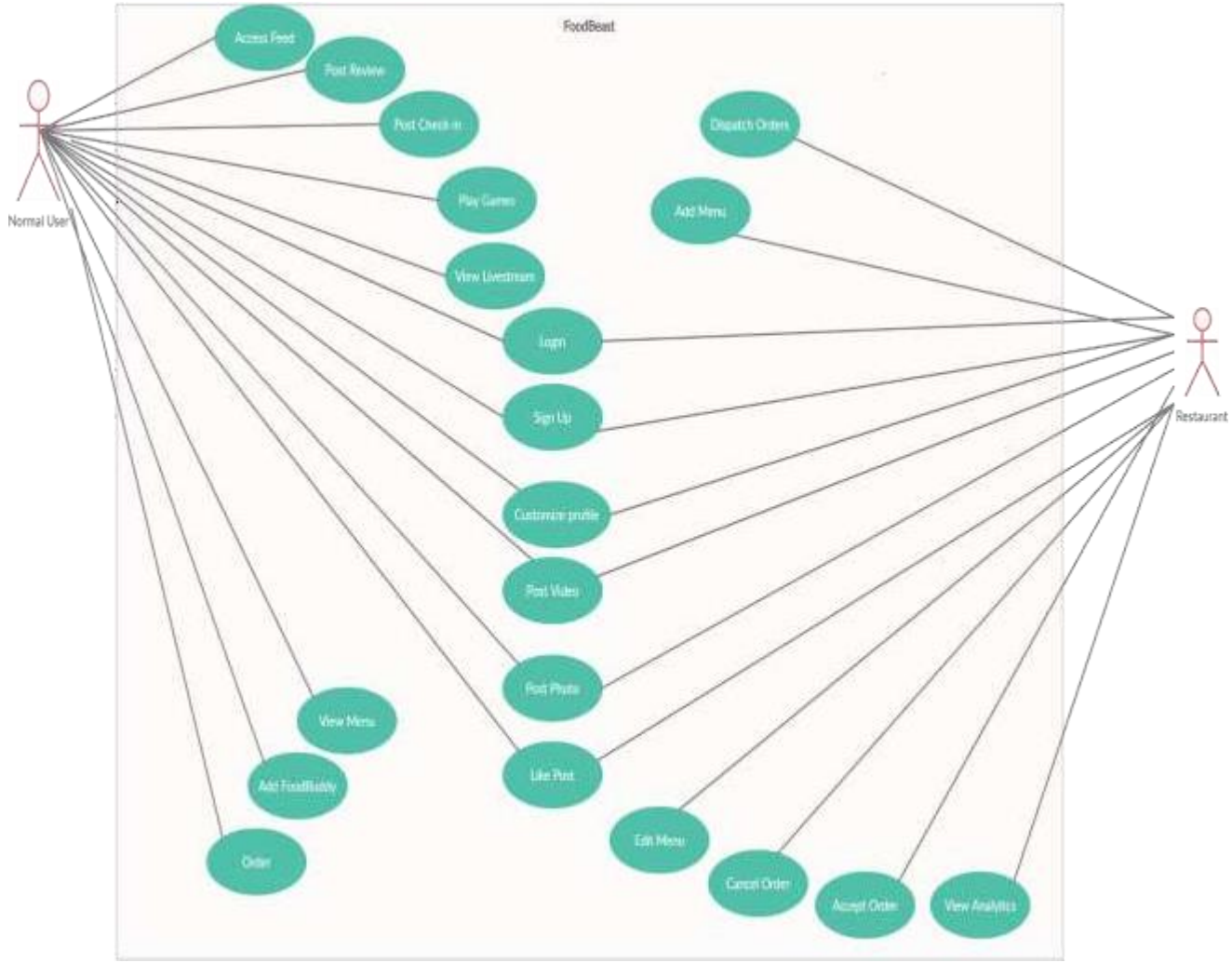

Figure 1-UCD 1

2.5.2 Normal User Use Case Design (UCD-02)

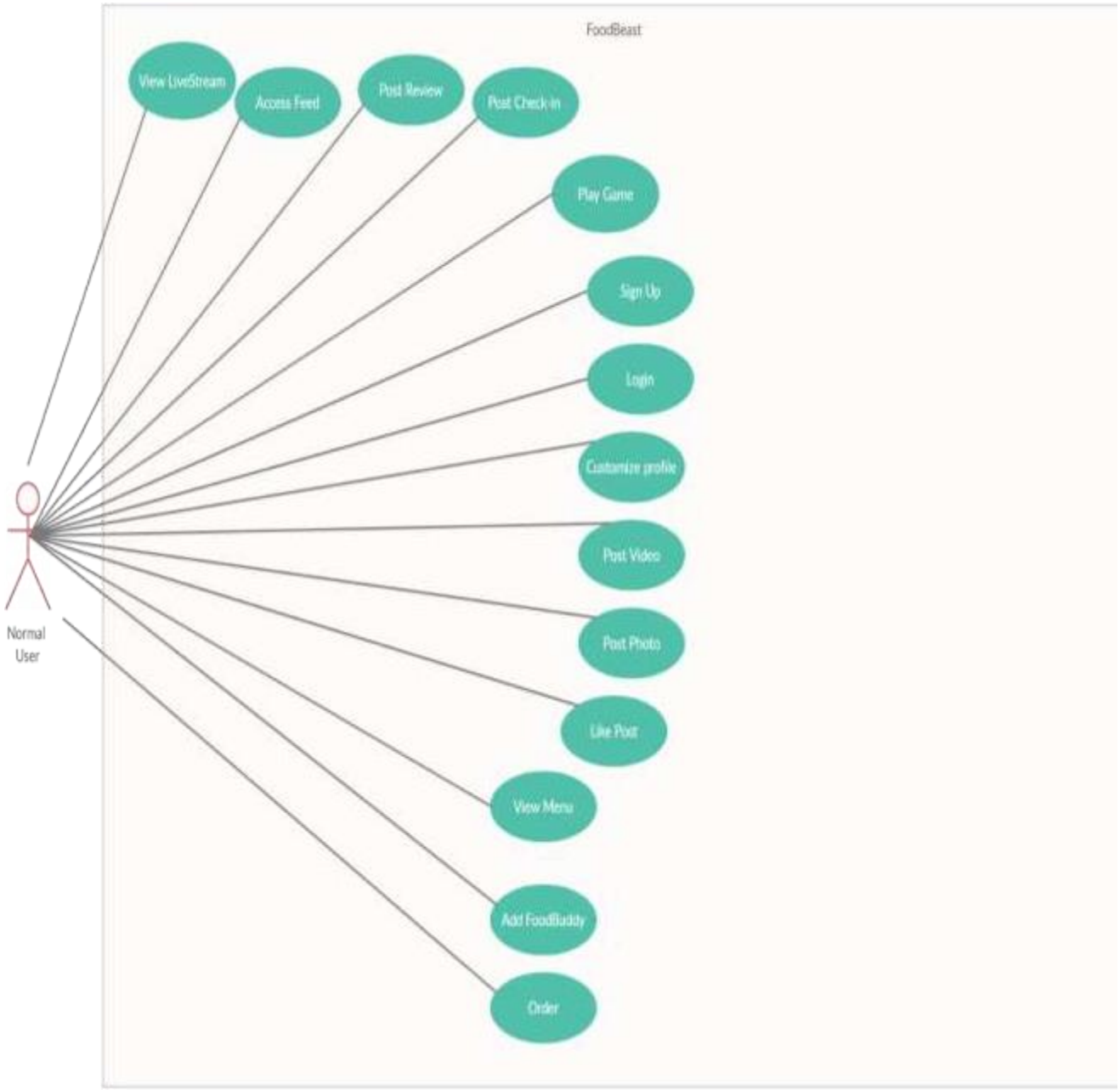

Figure 2-UCD 2

2.5.3 Restaurant User Use Case Design (UCD-03)

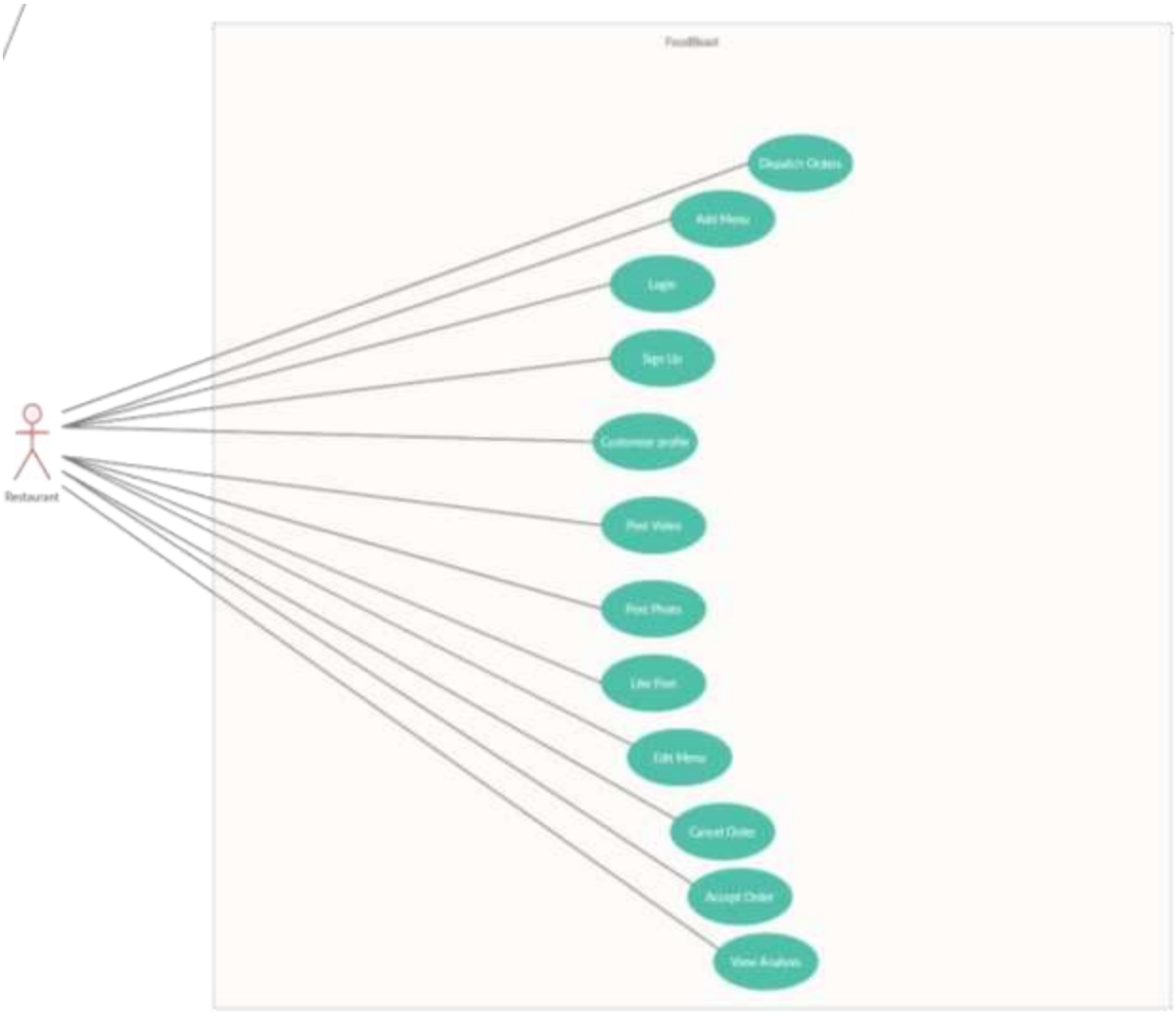

Figure 3-UCD 3

2.5.4 Admin Use Case Design (UCD-04)

Figure 4-UCD 4

2.6 Software Development Life Cycle

Agile is a software development methodology and an iterative process in which the project requirements and solutions evolve through collaboration between the project stakeholders. It is one the most disciplined project management process that encourages frequent inspection and adaptation. At each iteration in agile model the product is tested. These short releases let the developers understand application in a broader way and makes debugging process easy. For example, instead of spending hours or days building out a detailed project plan that may or may not match the project's execution, an Agile team would begin with a small portion of work, assess progress, then continue down the path once enough feedback has been gathered to indicate they are on the right path.

Advantages of Agile model:

- Small pieces developed with bug fixing will be integrated in the end will make everyone happy.

- Flexibility.
- Work load is decreased which results in better development.
- Finding bugs is easier with each development being tested.
- Customer satisfaction with every module development.

Why not any other model?

- The other models encourages to be completed first and then to be tested.
- Iterative process are not acknowledged in other models.
- Short releases of applications are not encouraged in other models.
- Bugs are easily detected and removed with iterative process.
- We can say that Agile is an evolved version of all the models.
- Each iterations make sure, if there is a space to add or to remove anything.

3 Chapter 3: System Design

3.1 Work Breakdown Structure (WBS)

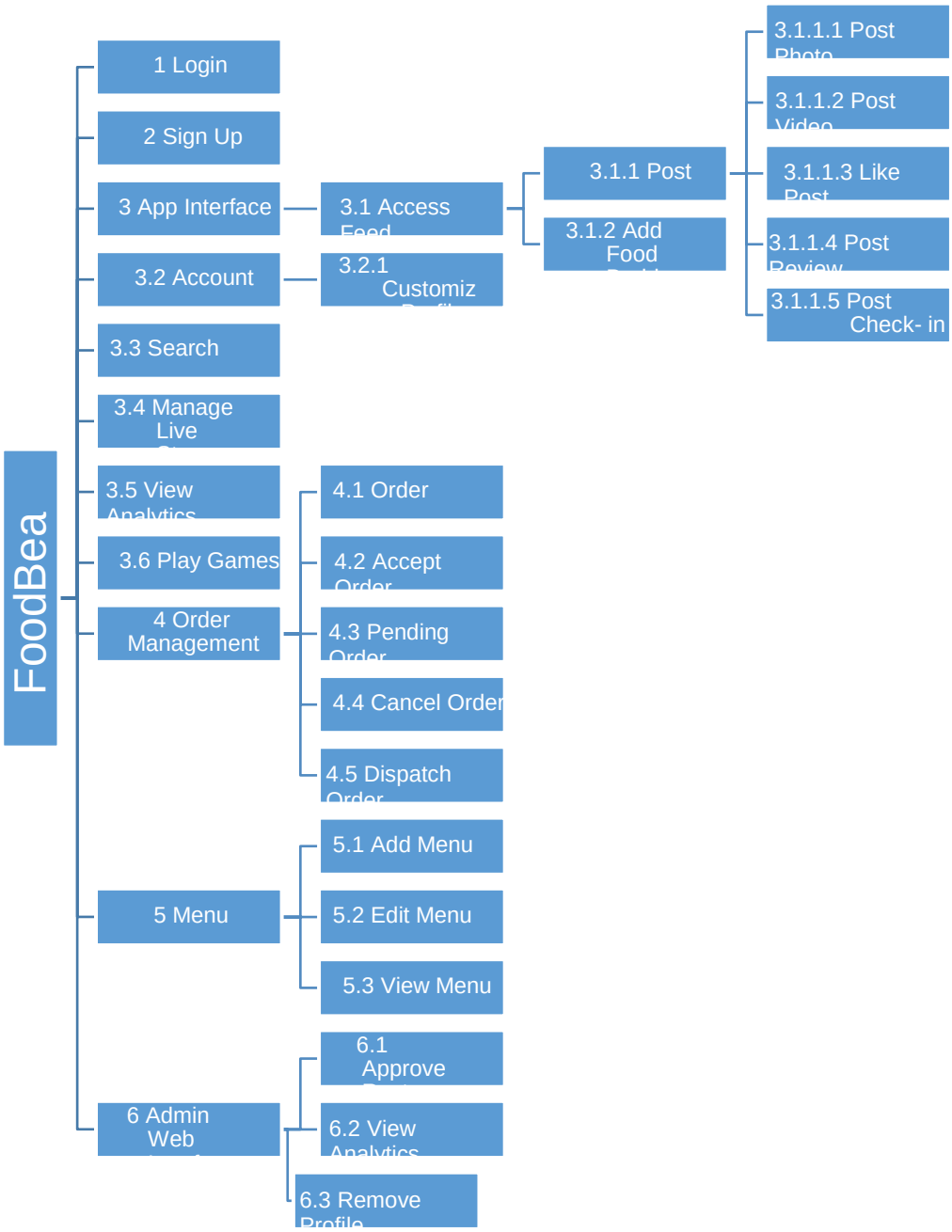

Figure 5- WBS

3.2 Activity Diagram

3.2.1 Sign Up AC01

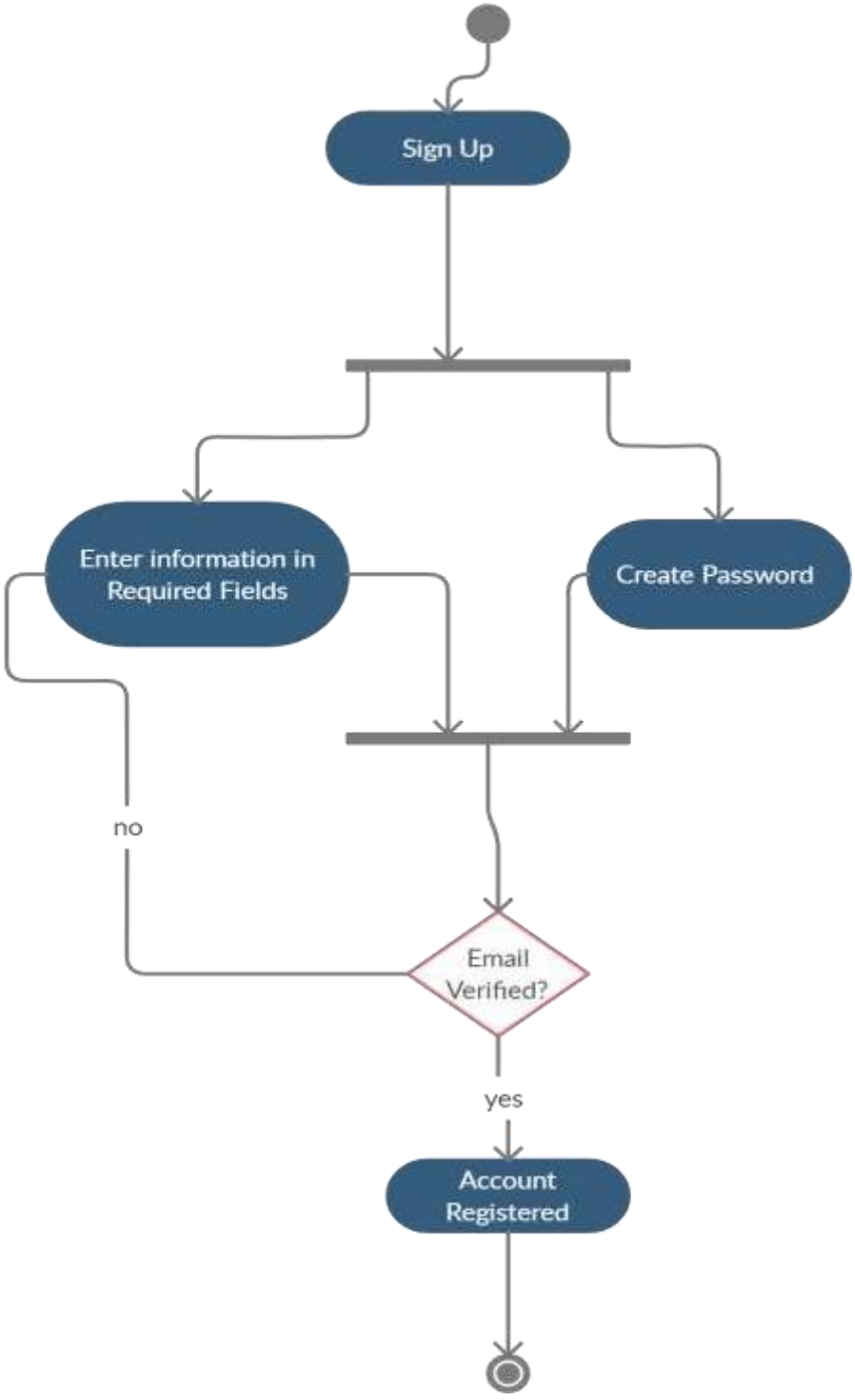

Figure 6- AC 1

3.2.2 Login AC02

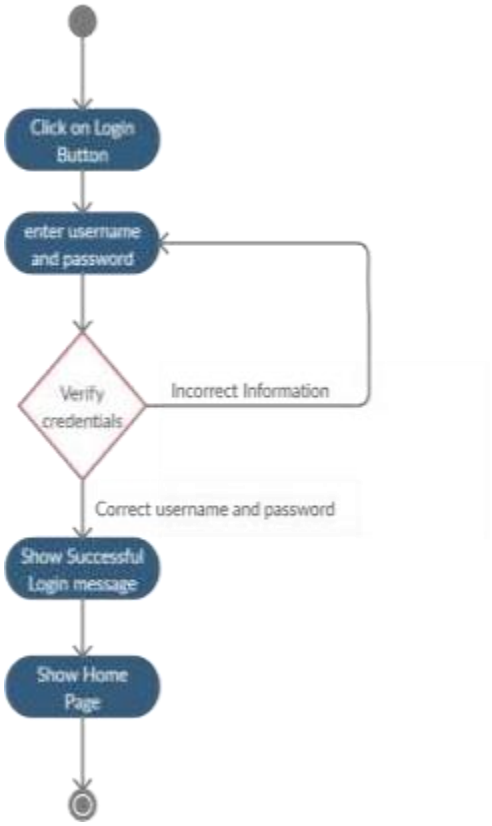

Figure 7- AC 2

3.2.3 Customize Profile AC03

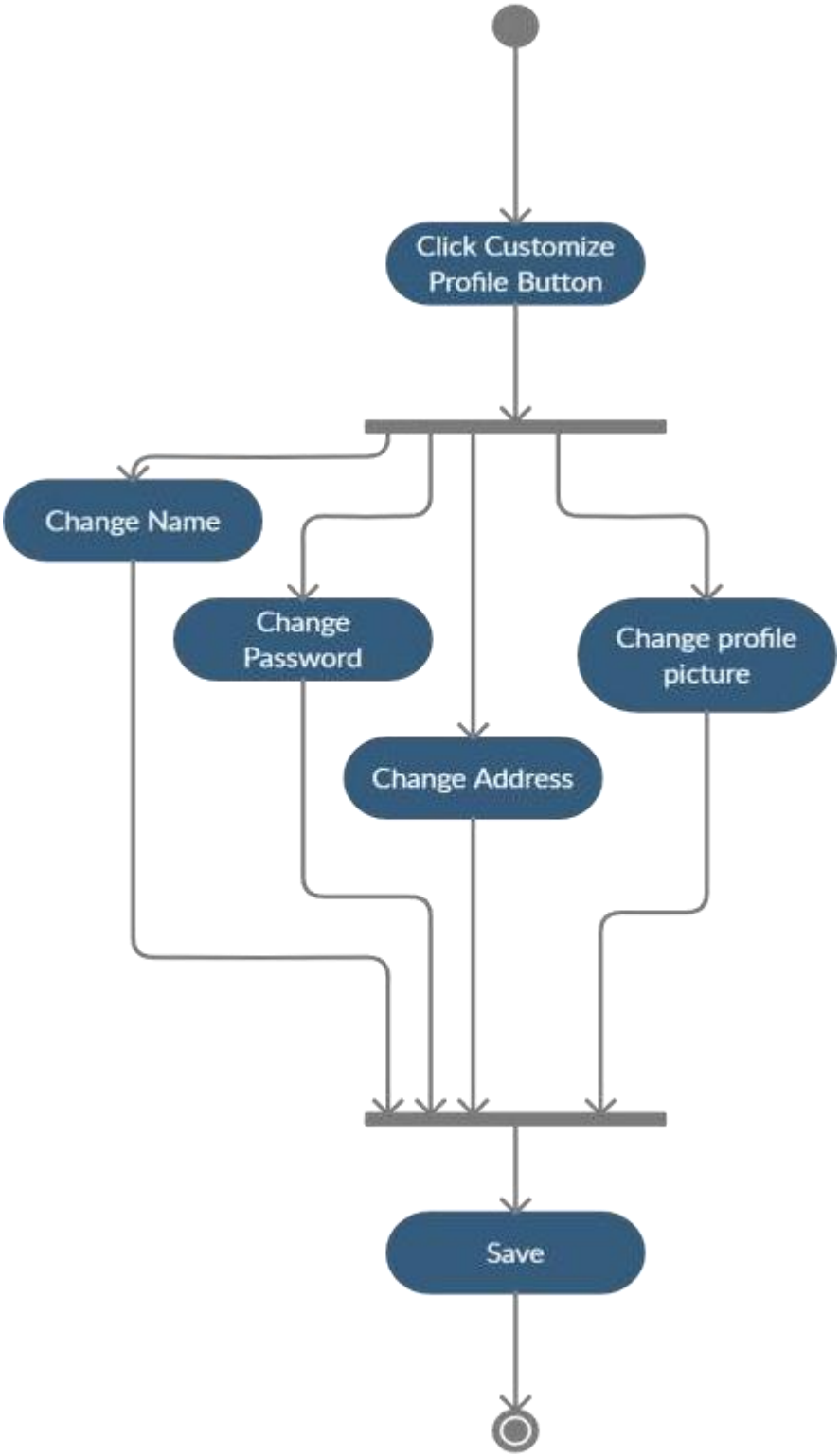

Figure 8- AC 3

3.2.4 Post Video AC04

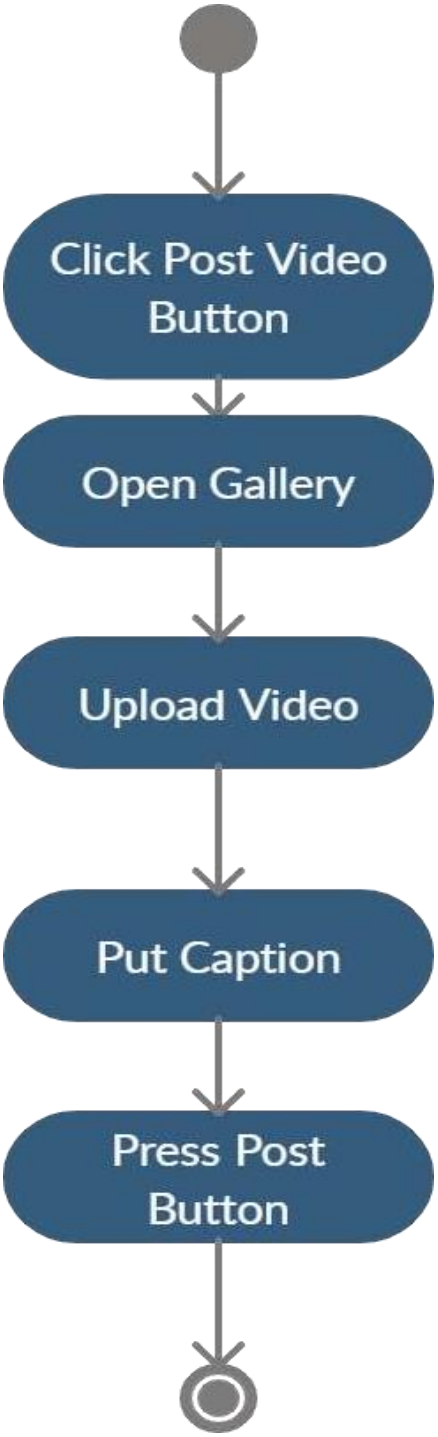

Figure 9- AC 4

3.2.5 Post Photo AC05

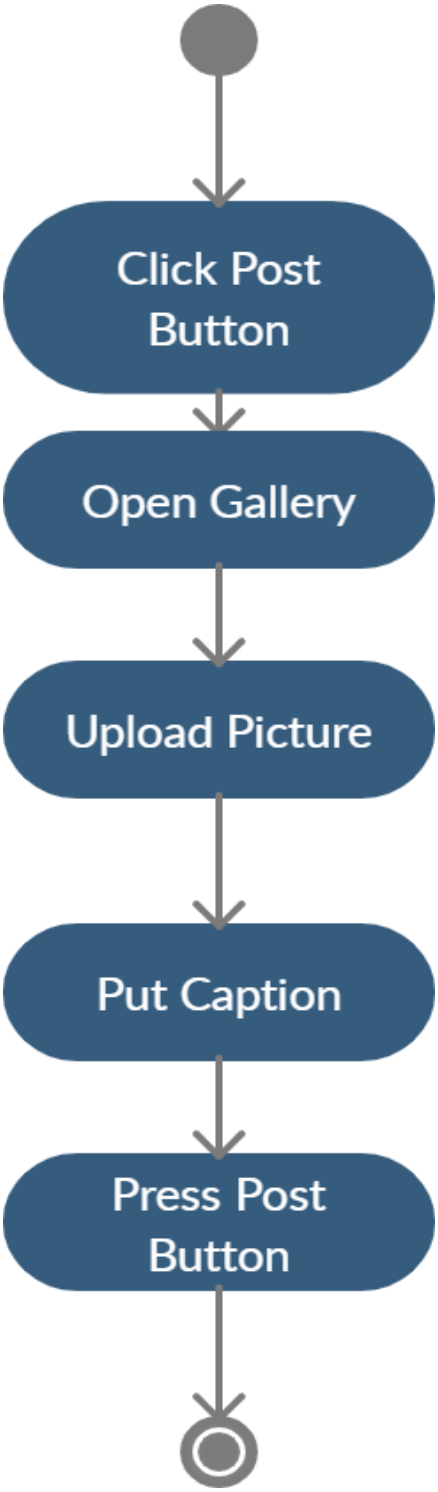

Figure 10- AC 5

3.2.6 Like Post AC06

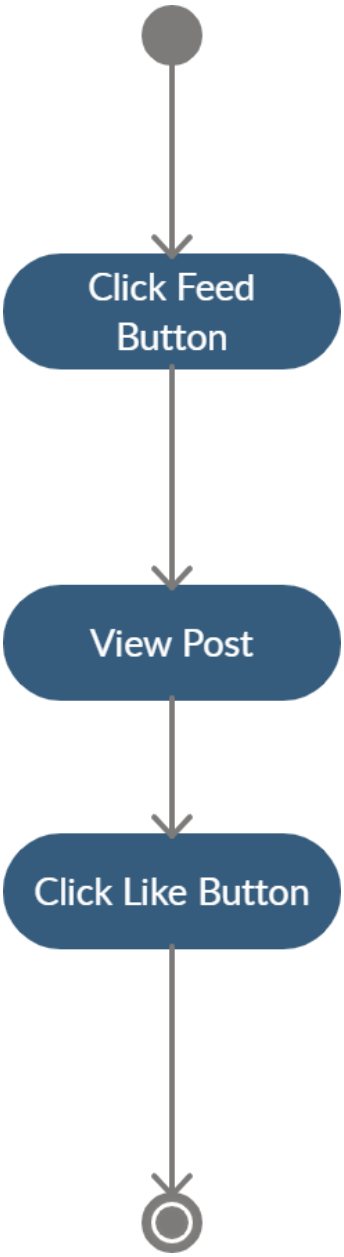

Figure 11- AC 6

3.2.7 Access Feed AC07

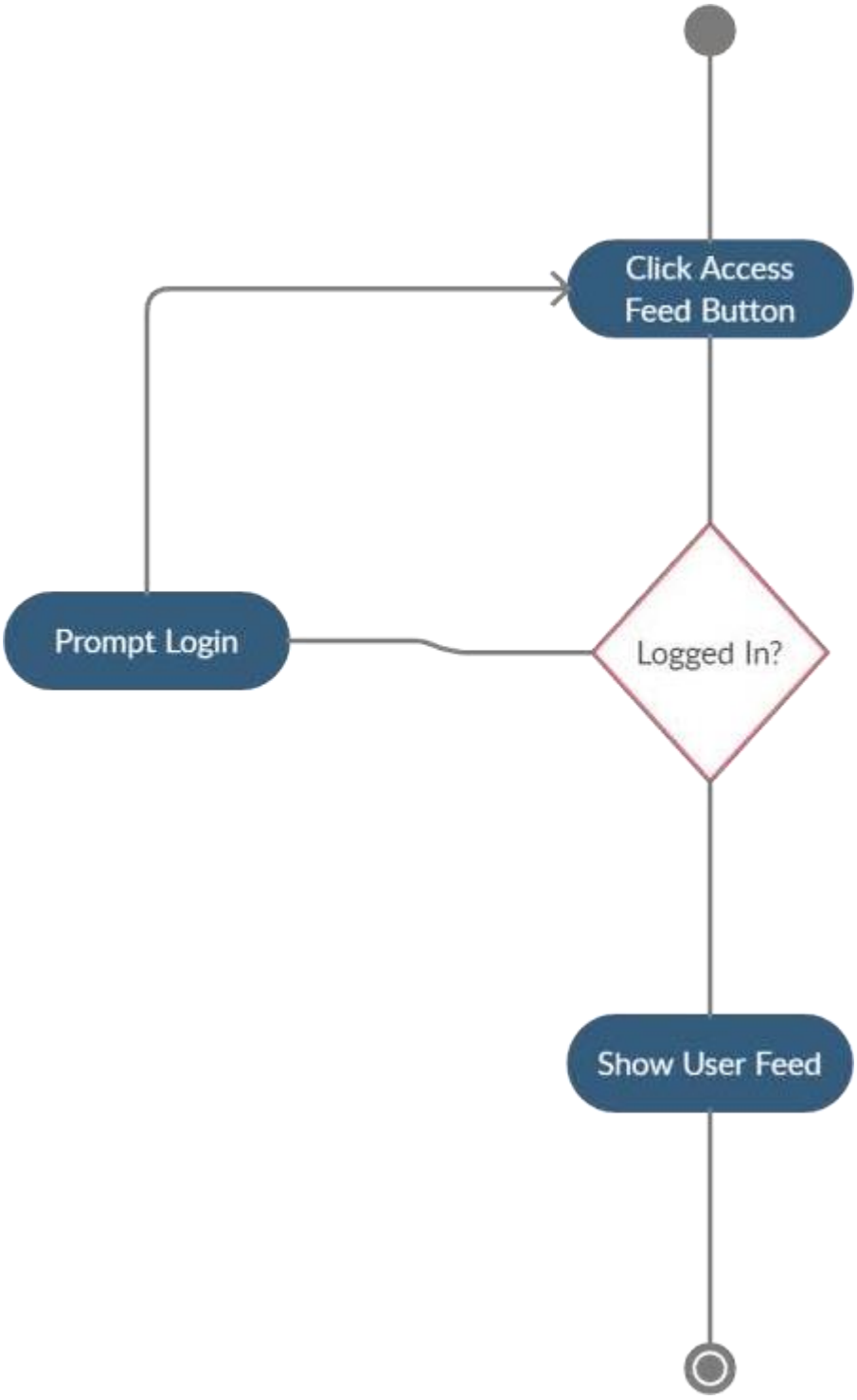

Figure 12- AC 7

3.2.8 Post Review AC08

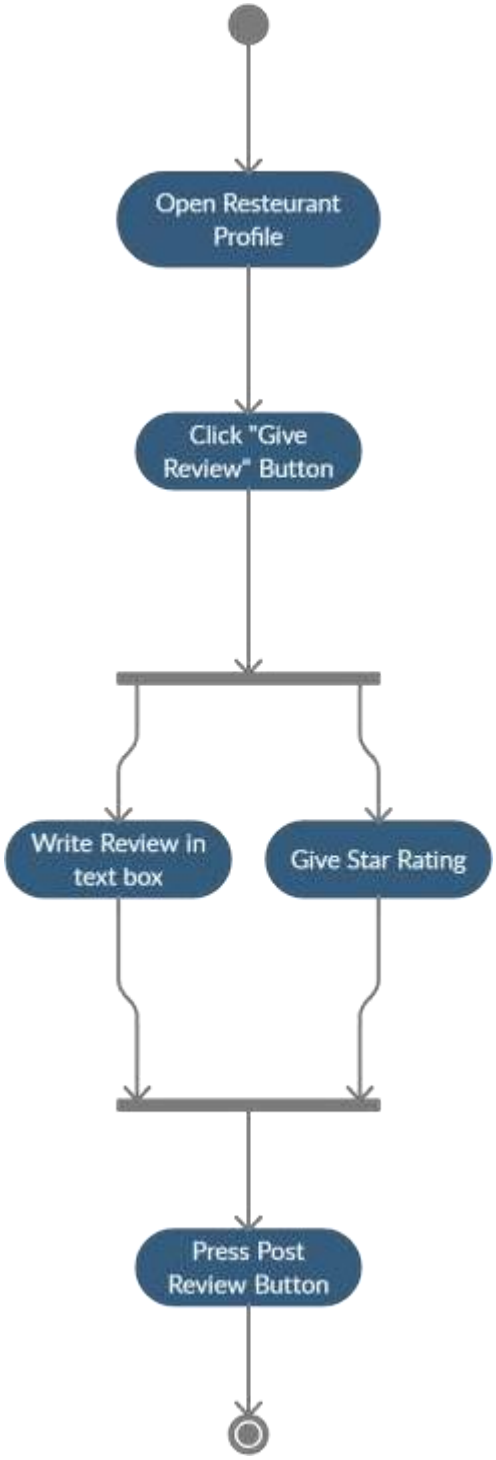

Figure 13- AC 8

3.2.9 Add FoodBuddy AC09

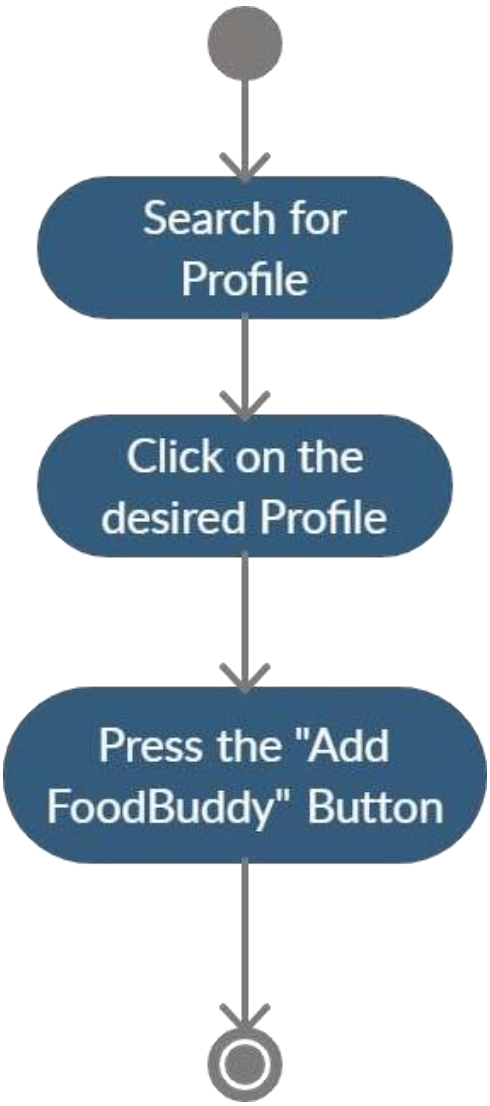

Figure 14- AC 9

3.2.10 Post Check-in AC10

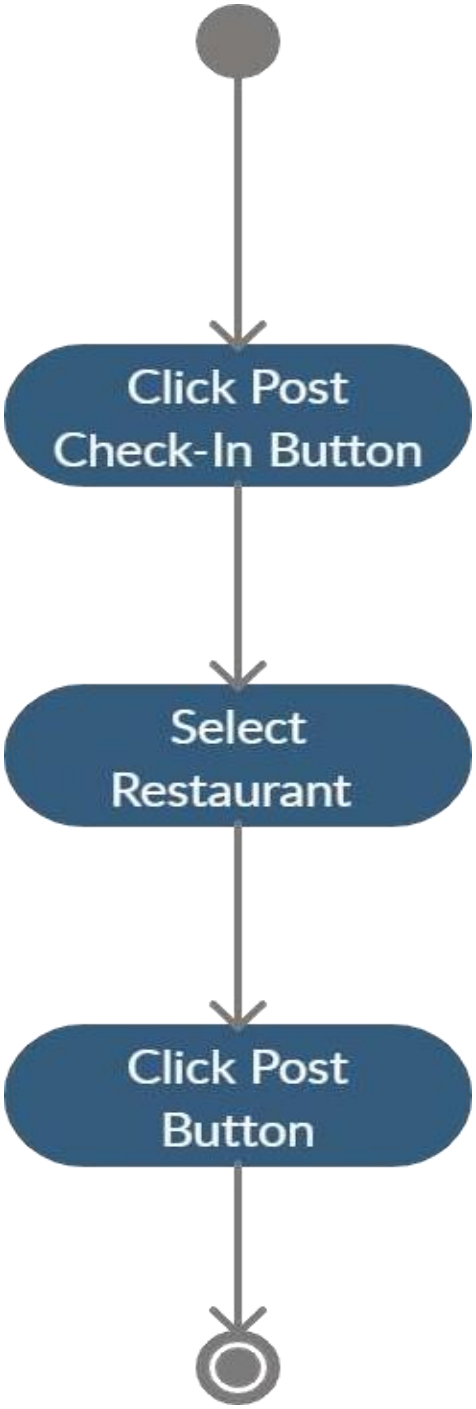

Figure 15- AC 10

3.2.11 Play Game AC11

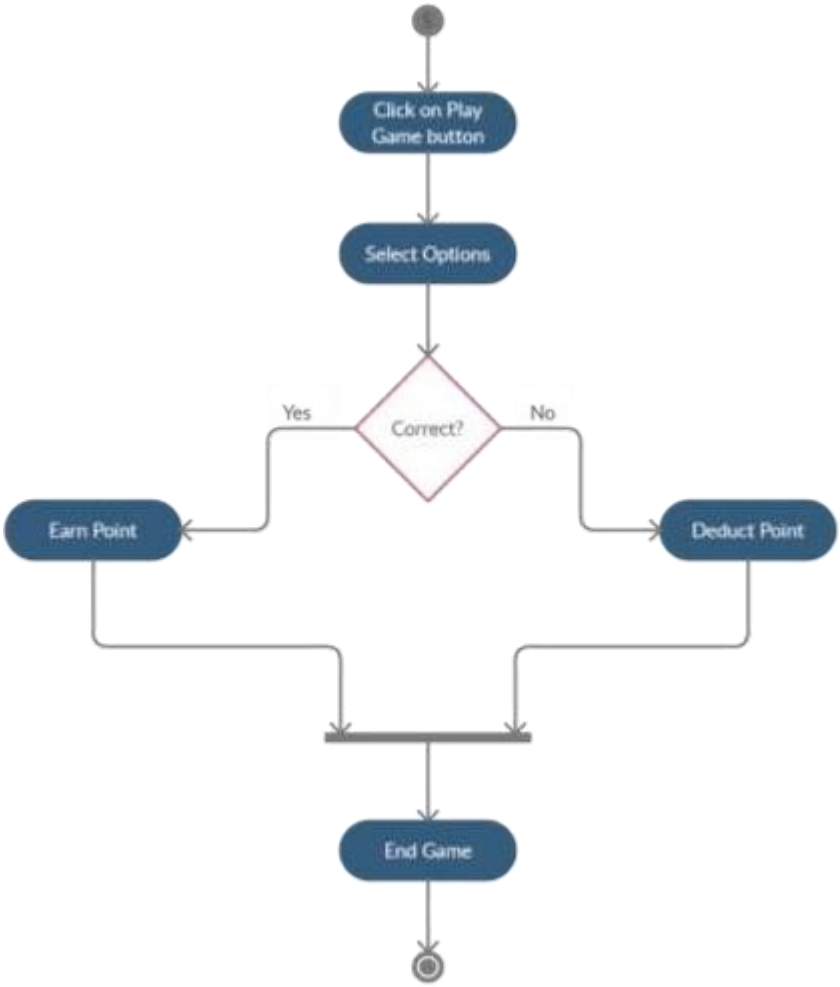

Figure 16- AC 11

3.2.12 View Menu AC12

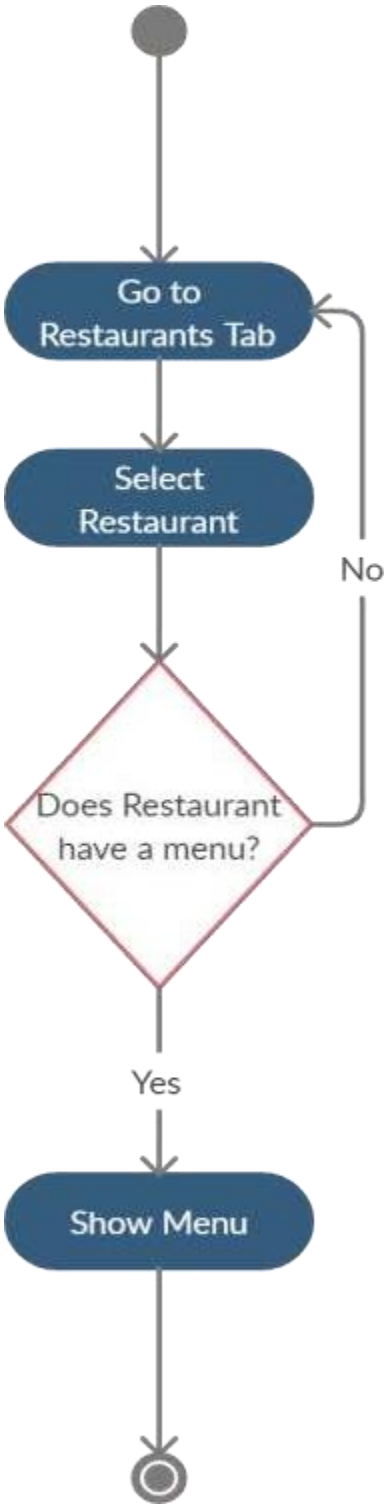

Figure 17- AC 12

3.2.13 Order AC13

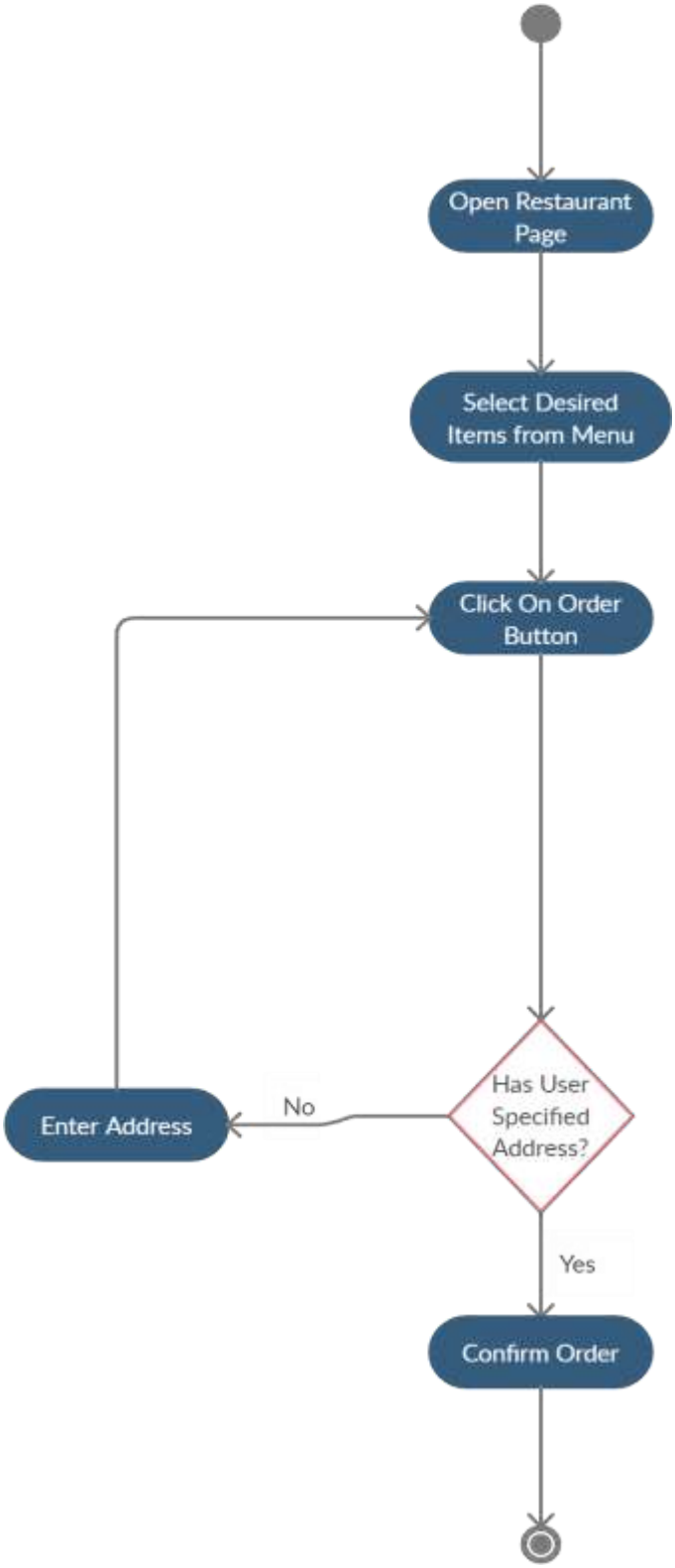

Figure 18- AC 13

3.2.14 Approve Restaurant AC14

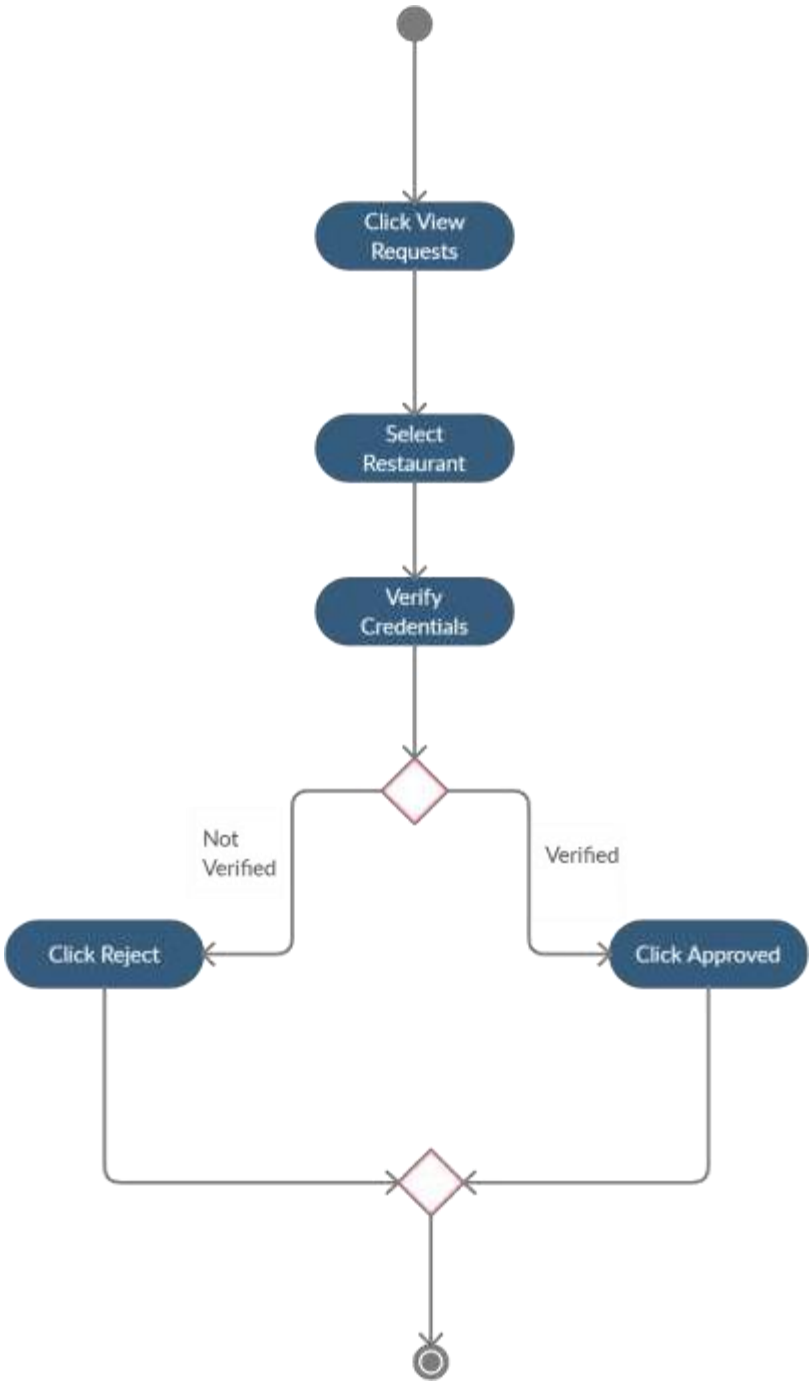

Figure 19- AC 14

3.2.15 Remove Profile AC15

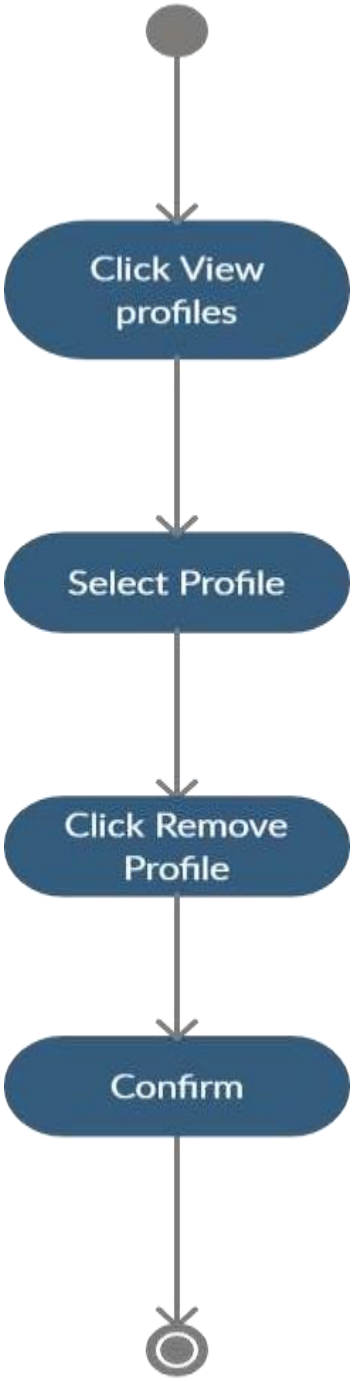

Figure 20- AC 15

3.2.16 Accept Order AC16

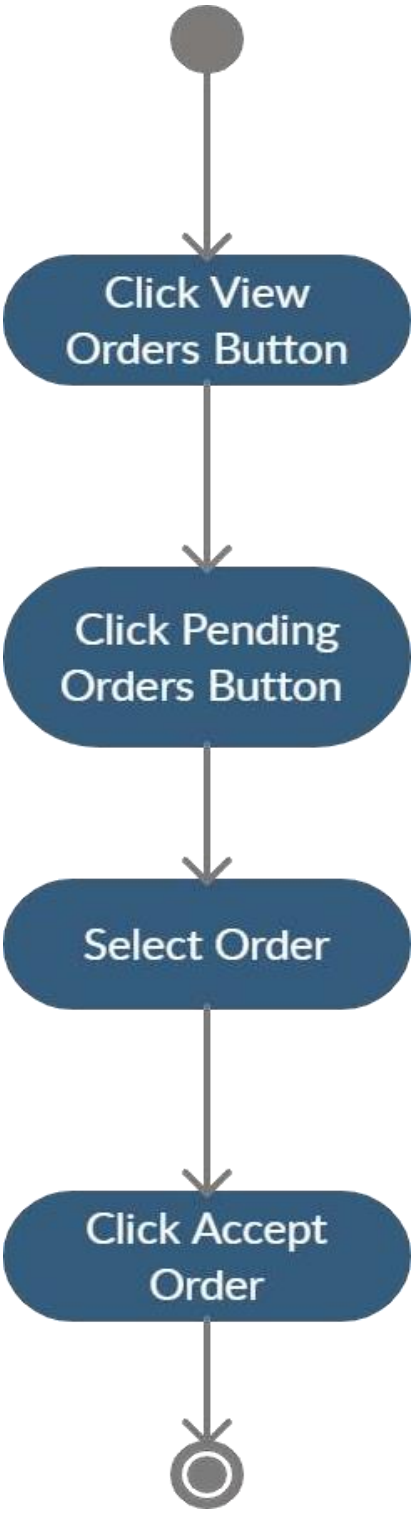

Figure 21- AC 16

3.2.17 Cancel Order AC17

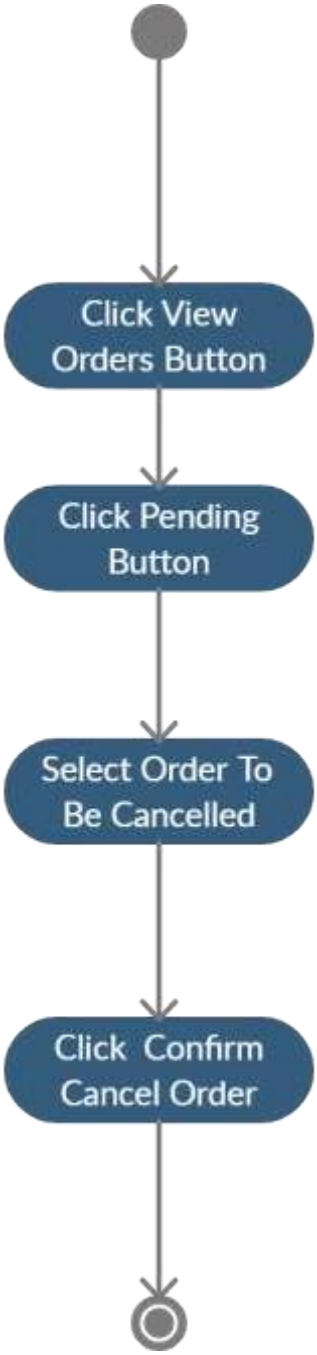

Figure 22- AC 17

3.2.18 Add Menu AC18

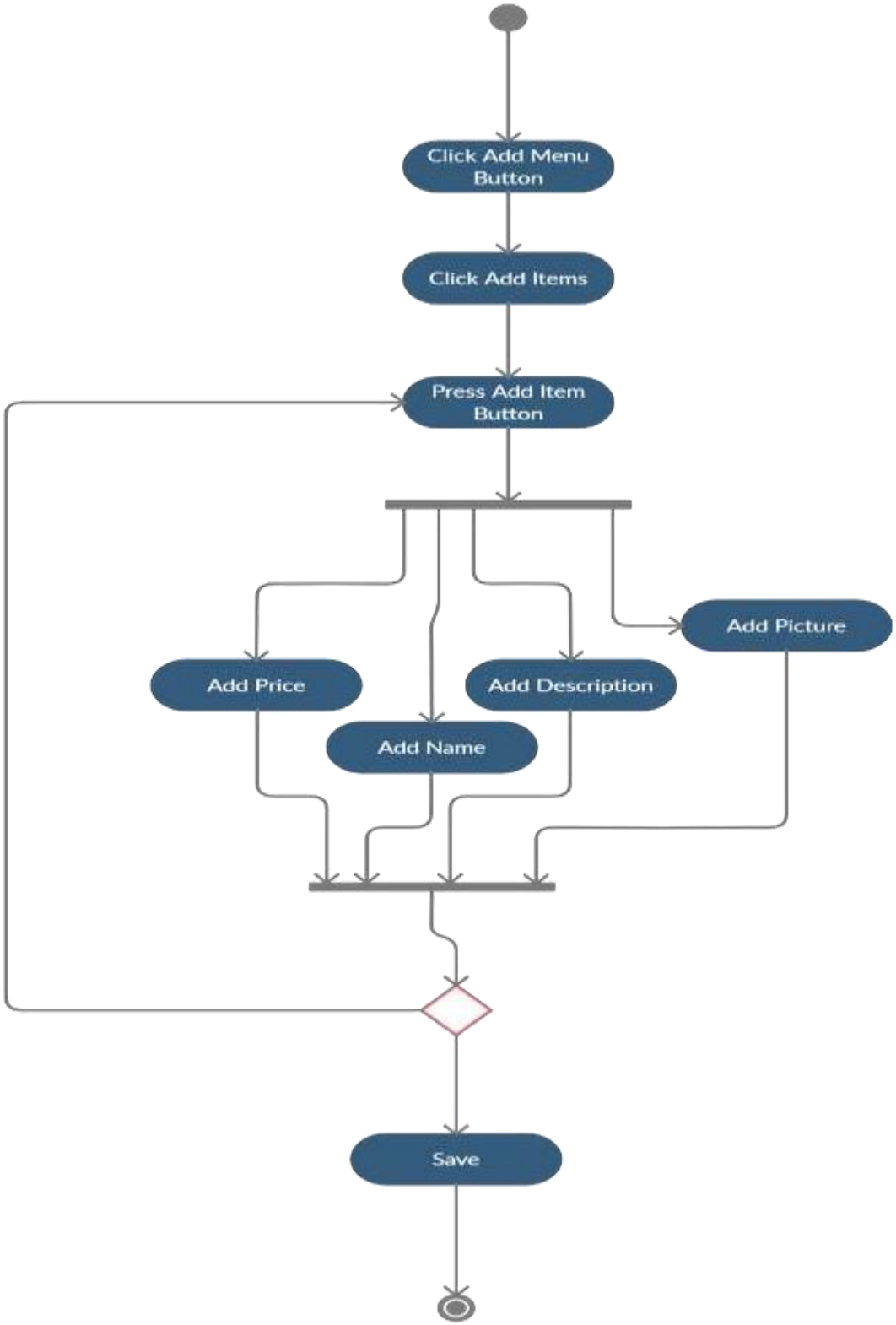

Figure 23- AC 18

3.2.19 Edit Menu AC19

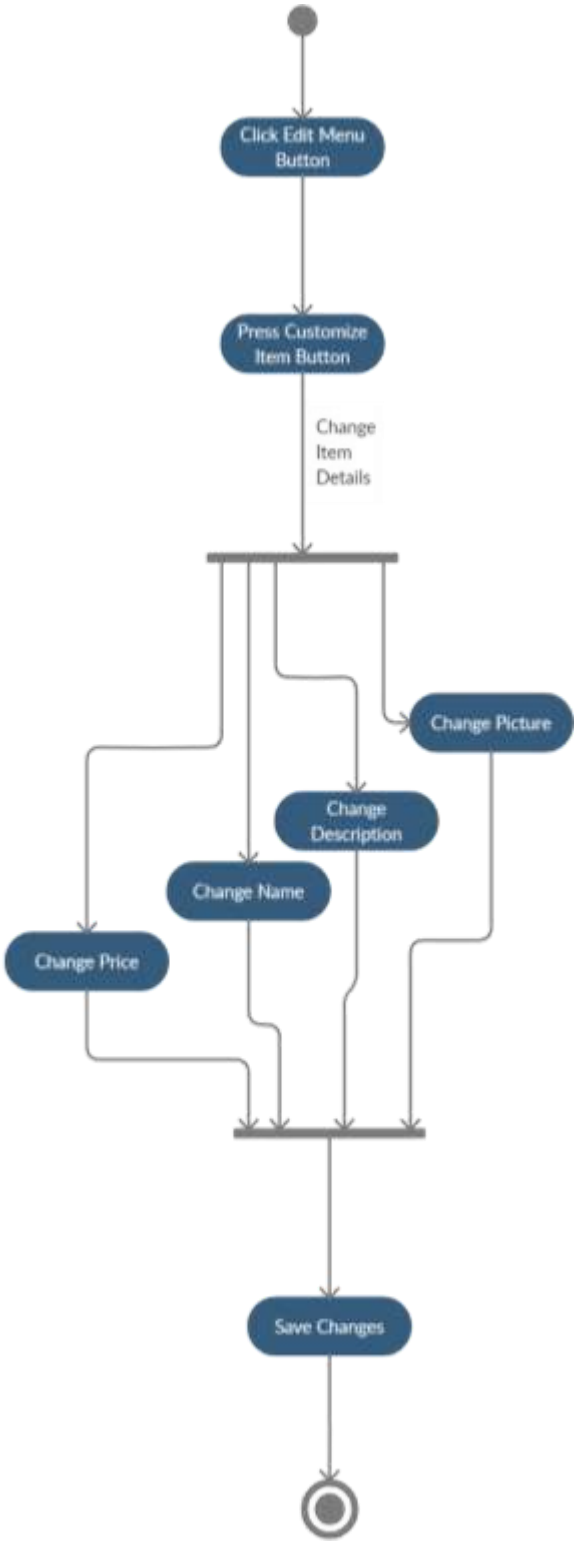

Figure 24- AC 19

3.2.20 View Pending Orders AC20

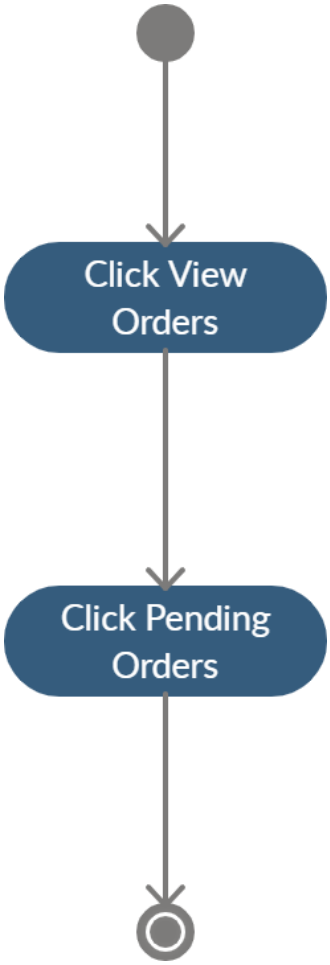

Figure 25- AC 20

3.2.21 Dispatch Orders AC21

Figure 26- AC 21

3.2.22 View Analytics AC22

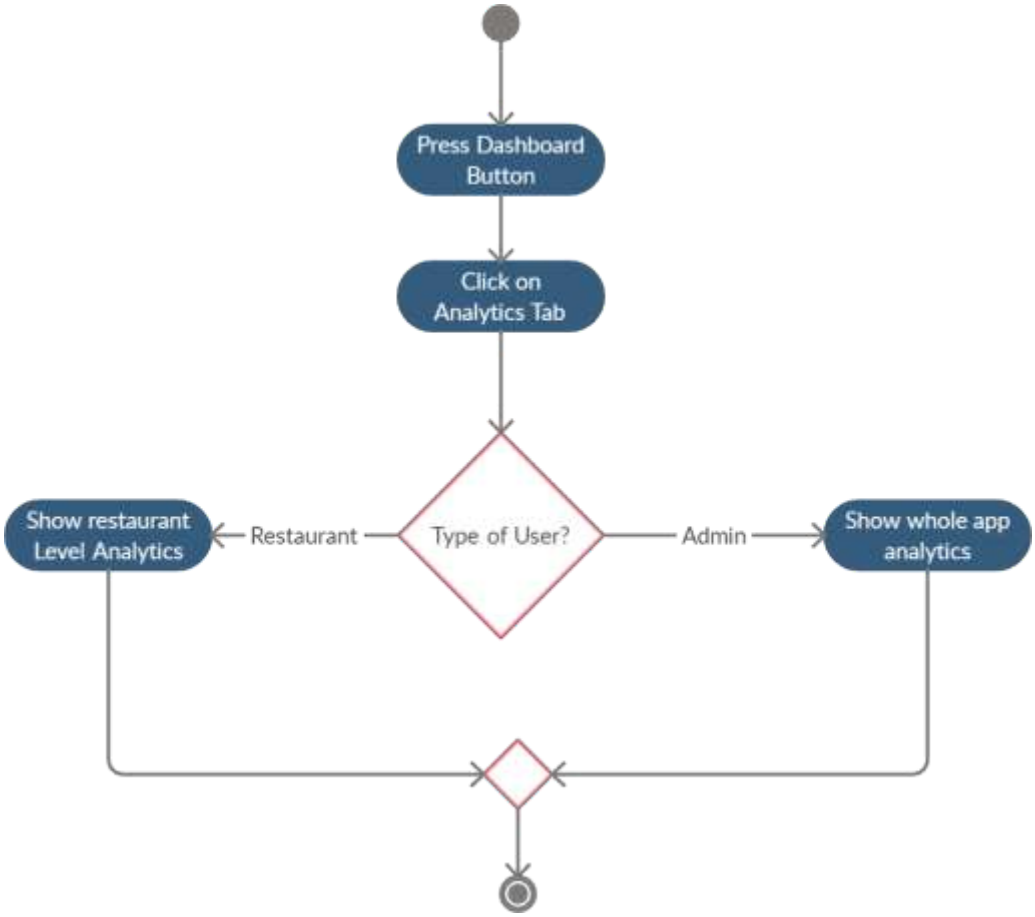

Figure 27- AC 22

3.2.23 Manage Stream AC23

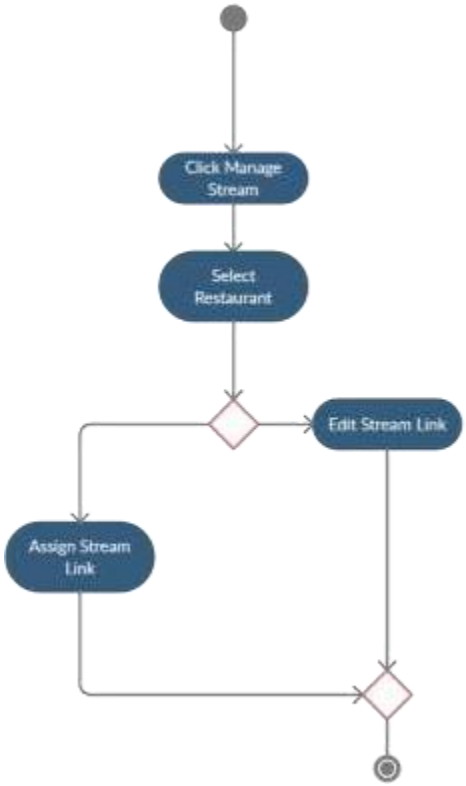

Figure 28- AC 23

3.2.24 View Live Stream

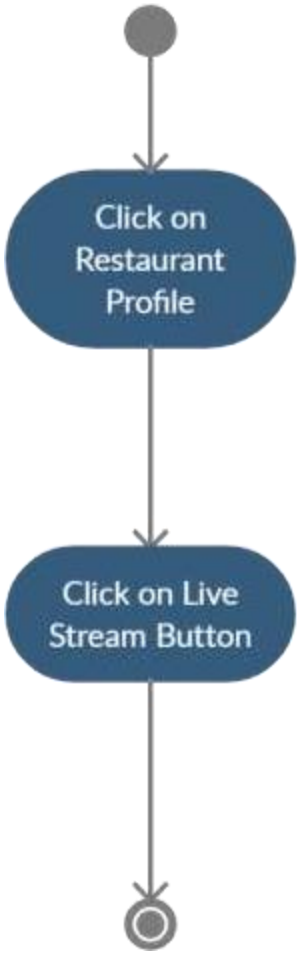

Figure 29- AC 24

3.3 Sequence Diagram

3.3.1 Sign Up SQ01

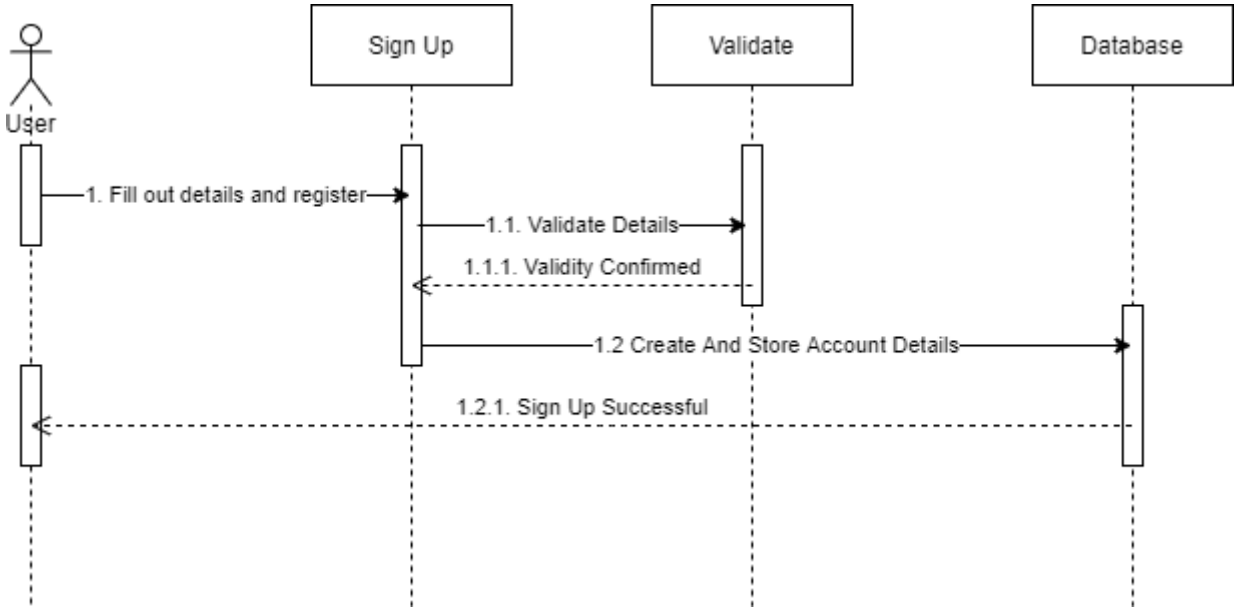

Figure 30- SQ 1

3.3.2 LOGIN SQ02

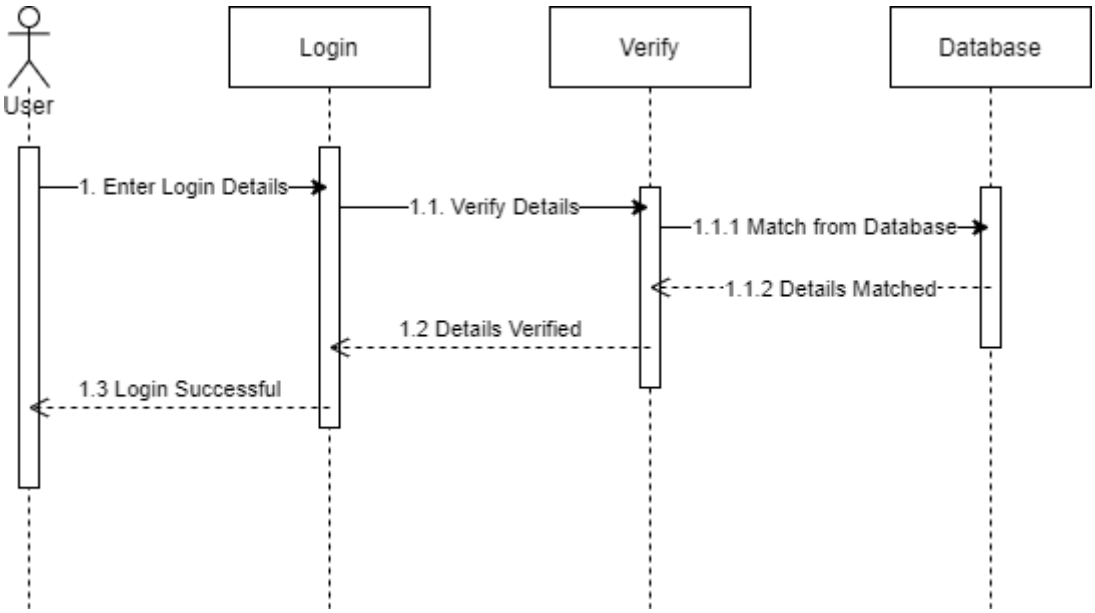

Figure 31- SQ 2

3.3.3 Customize Profile SQ03

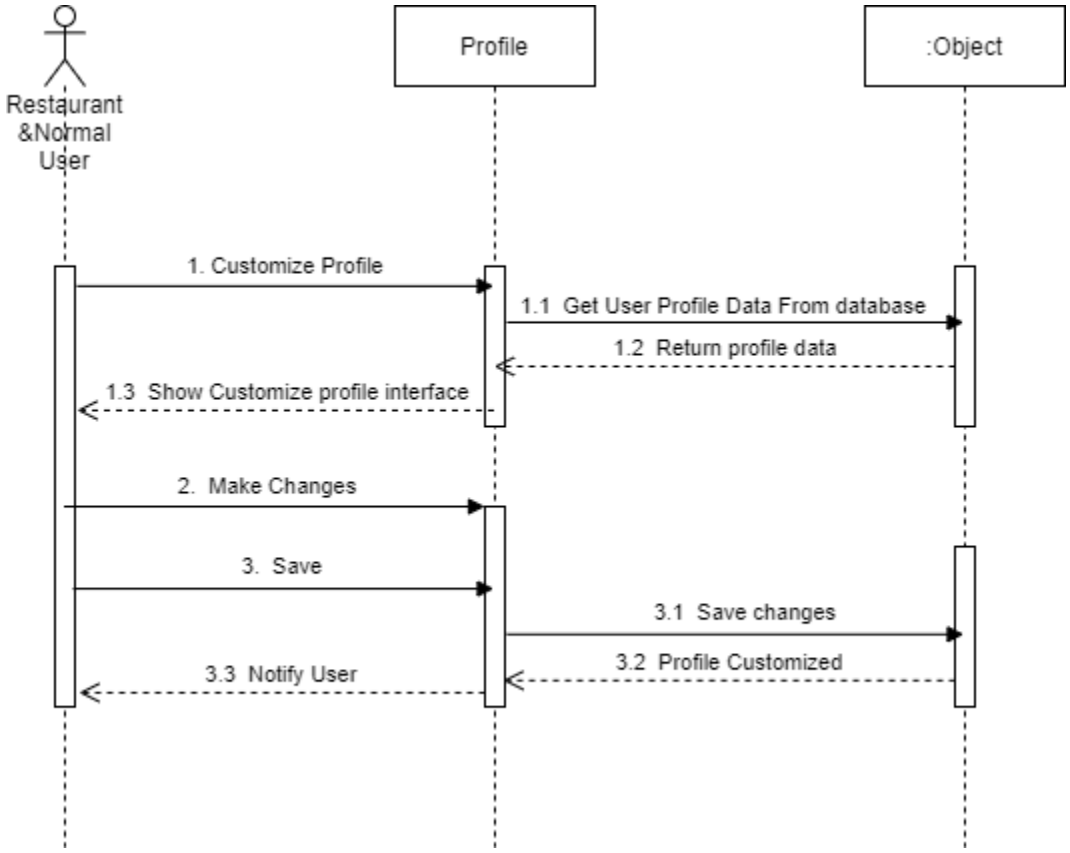

Figure 32- SQ 3

3.3.4 Post Video SQ04

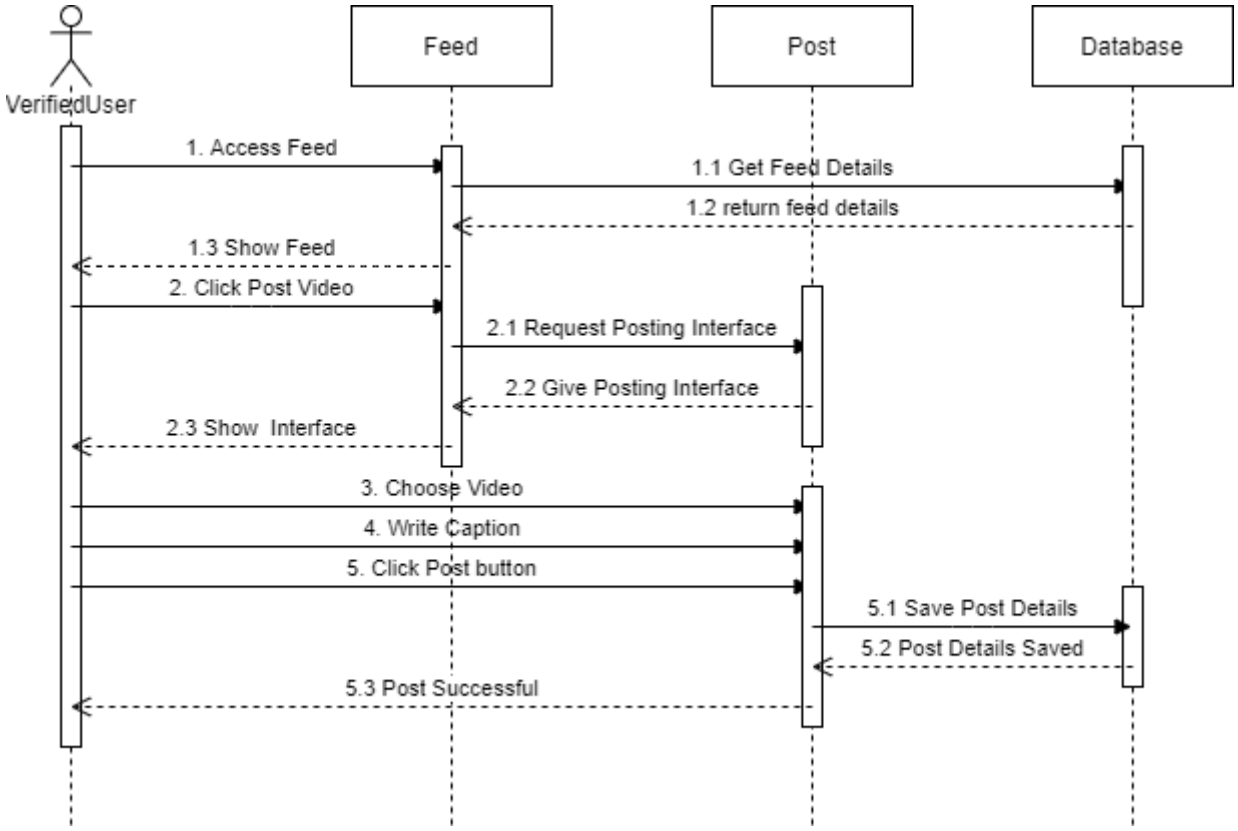

Figure 33- SQ 4

3.3.5 Post Photo SQ05

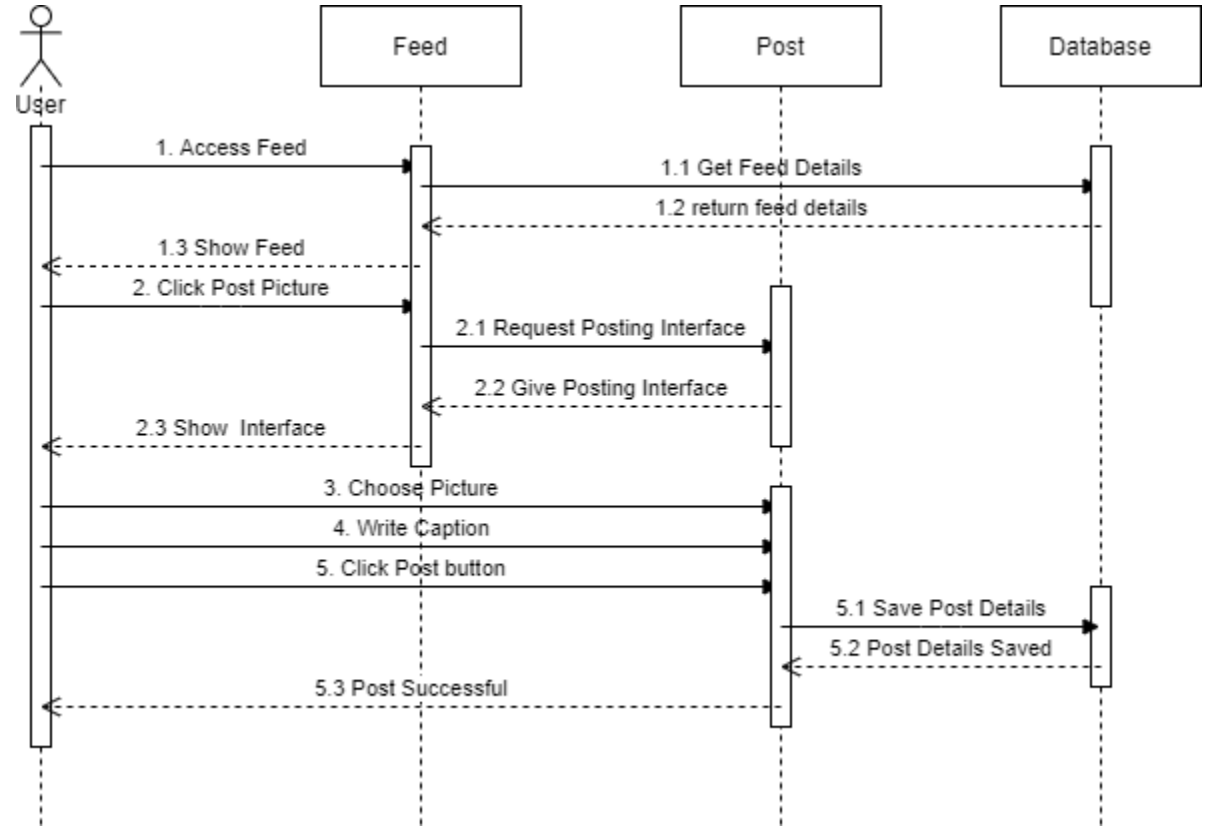

Figure 34- SQ 5

3.3.6 Like Post SQ06

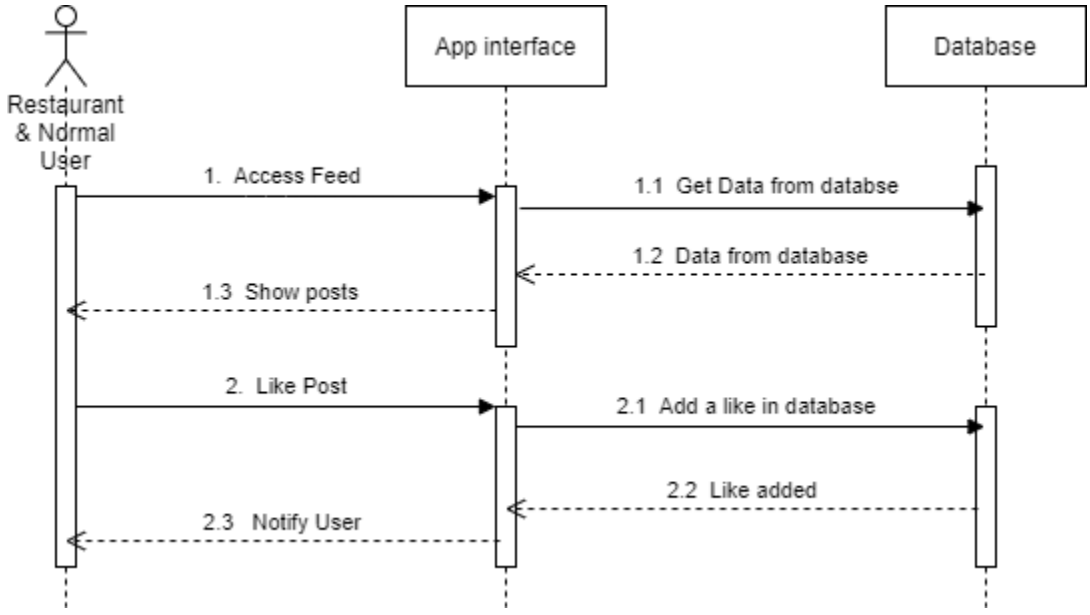

Figure 35- SQ 6

3.3.7 Access Feed SQ07

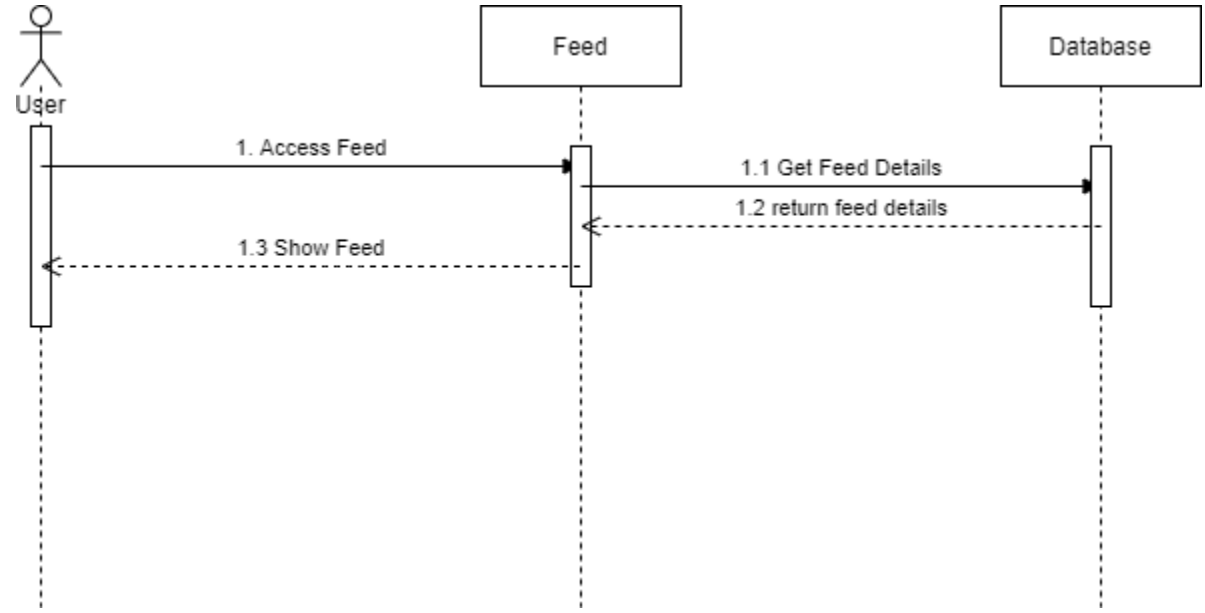

Figure 36- SQ 7

3.3.8 Post Review SQ08

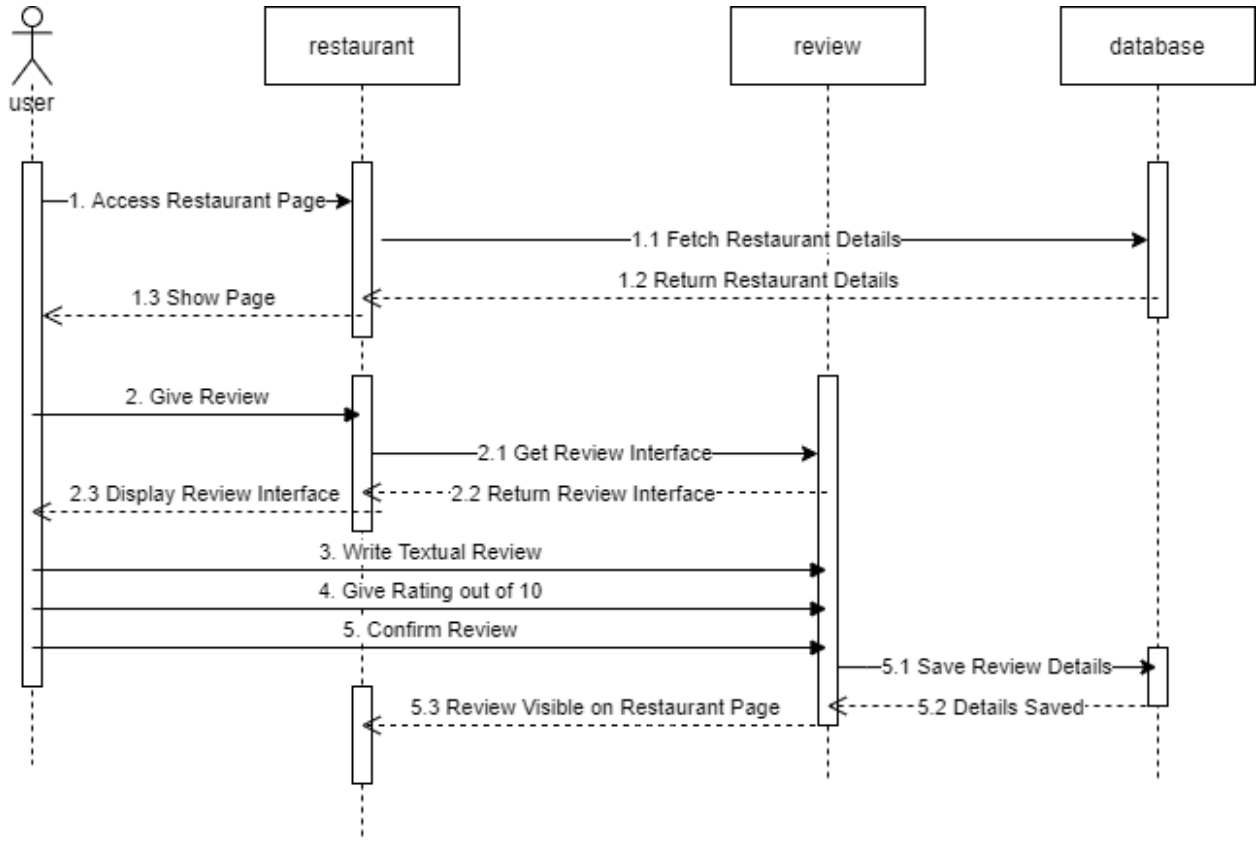

Figure 37- SQ 8

3.3.9 Add FoodBuddy SQ09

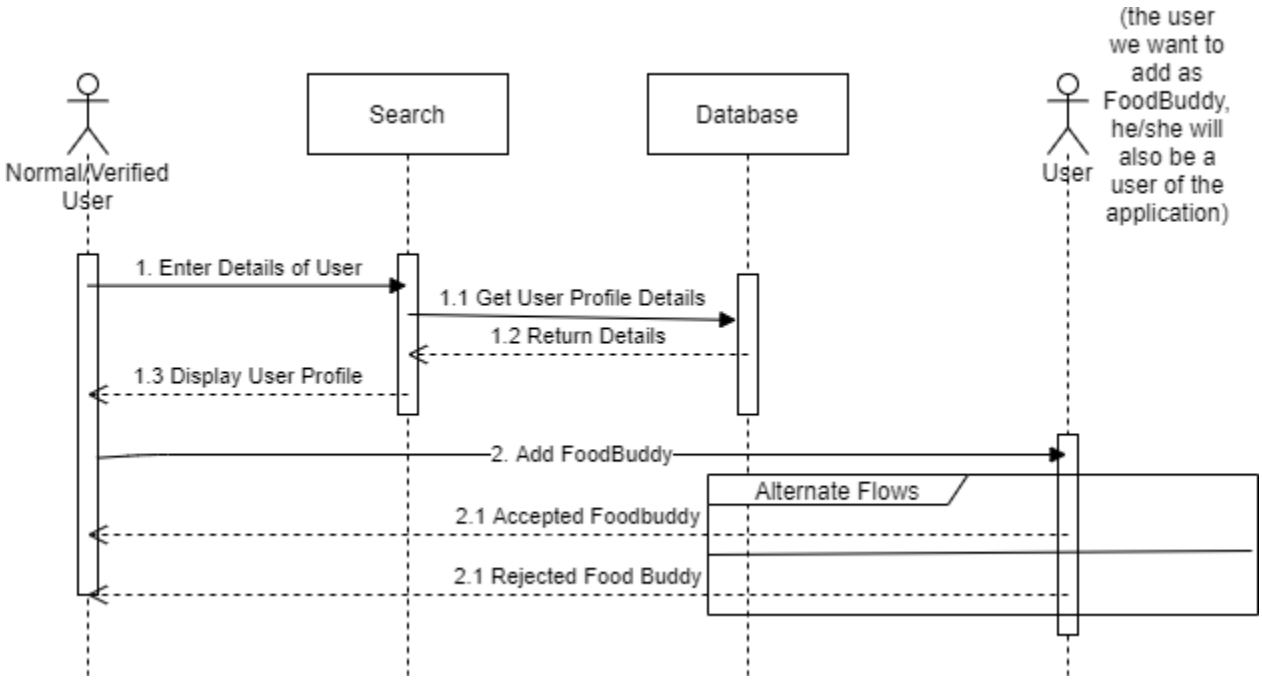

Figure 38- SQ 9

3.3.10 Post Check-in SQ10

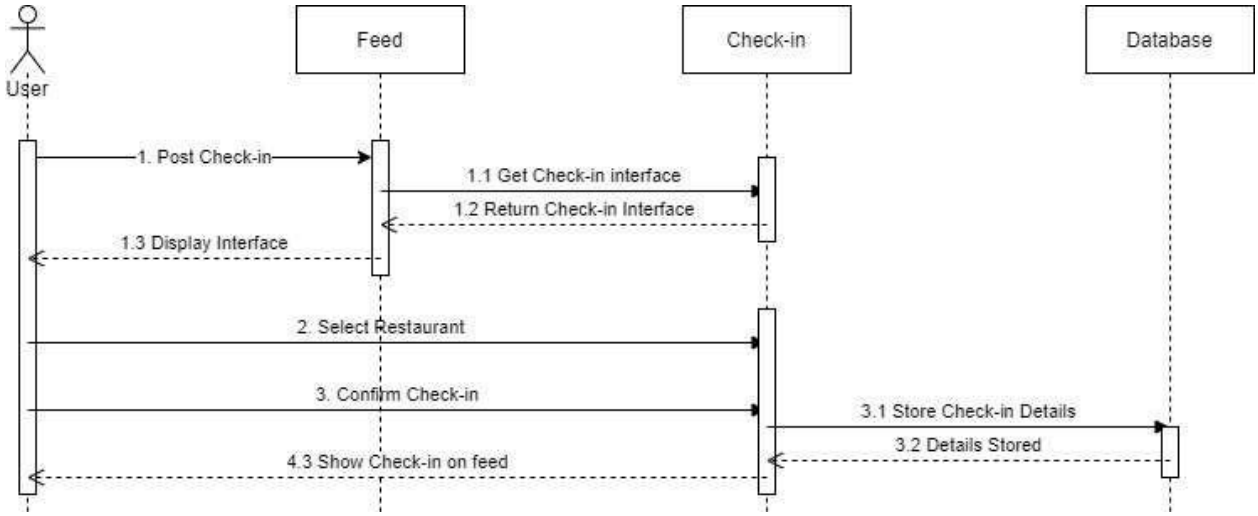

Figure 39- SQ 10

3.3.11 Play Game SQ11

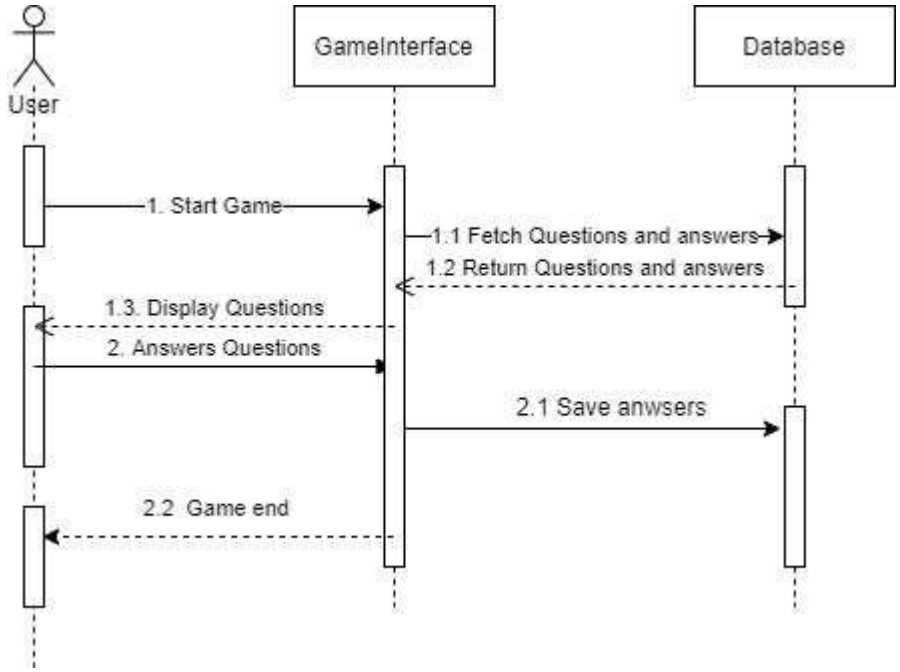

Figure 40- SQ 11

3.3.12 View Menu SQ12

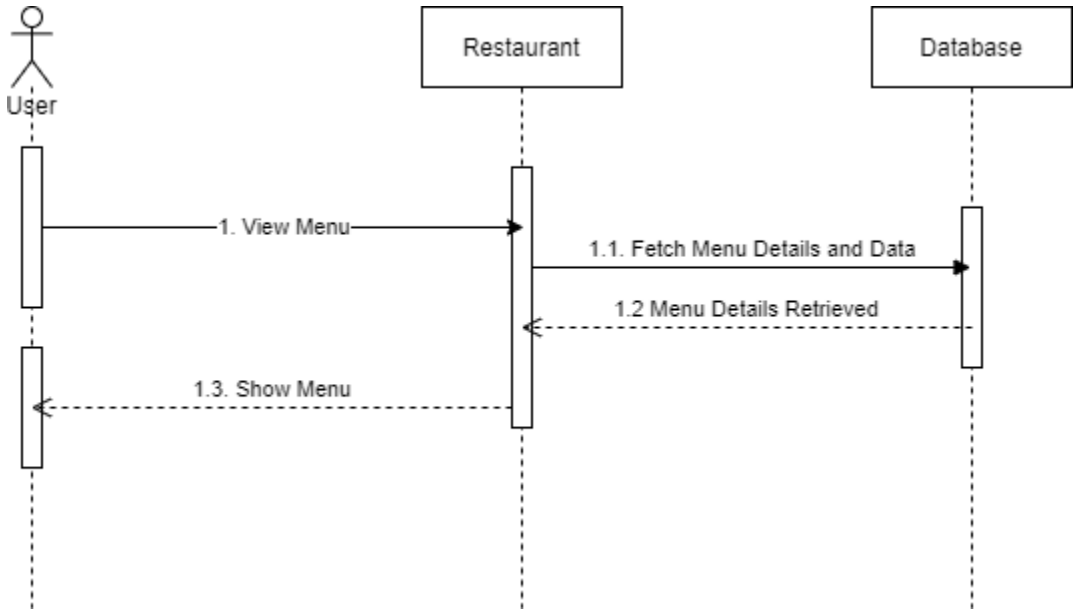

Figure 41- SQ 12

3.3.13 Order SQ13

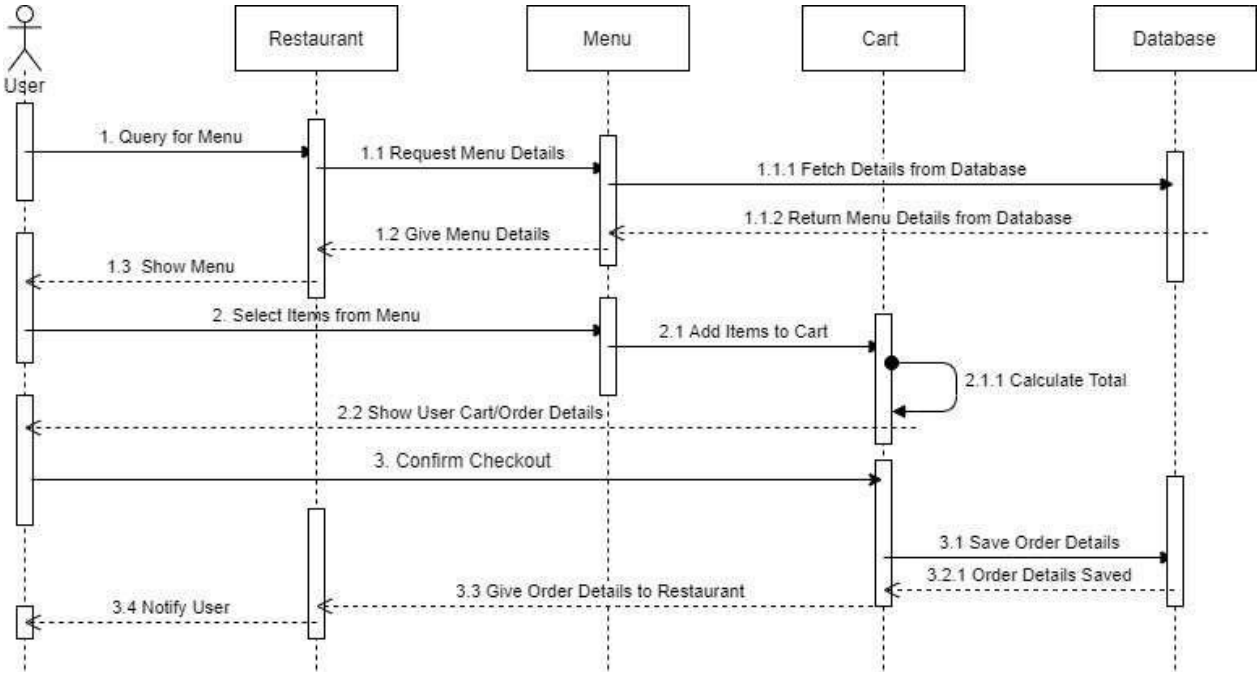

Figure 42- SQ 13

3.3.14 Approve Restaurant SQ14

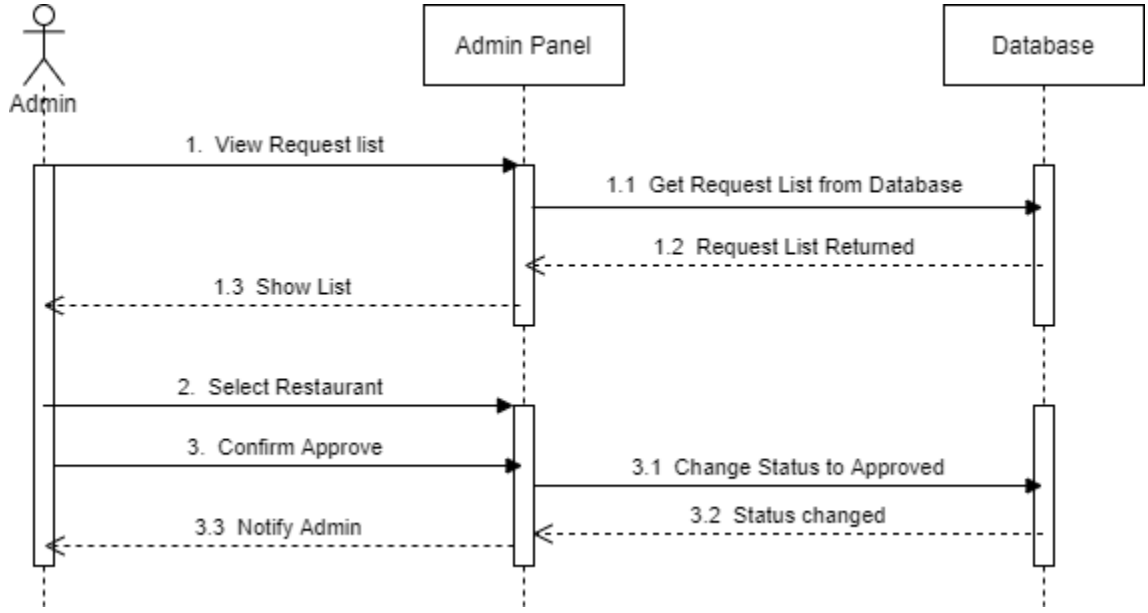

Figure 43- SQ 14

3.3.15 Remove Profile SQ15

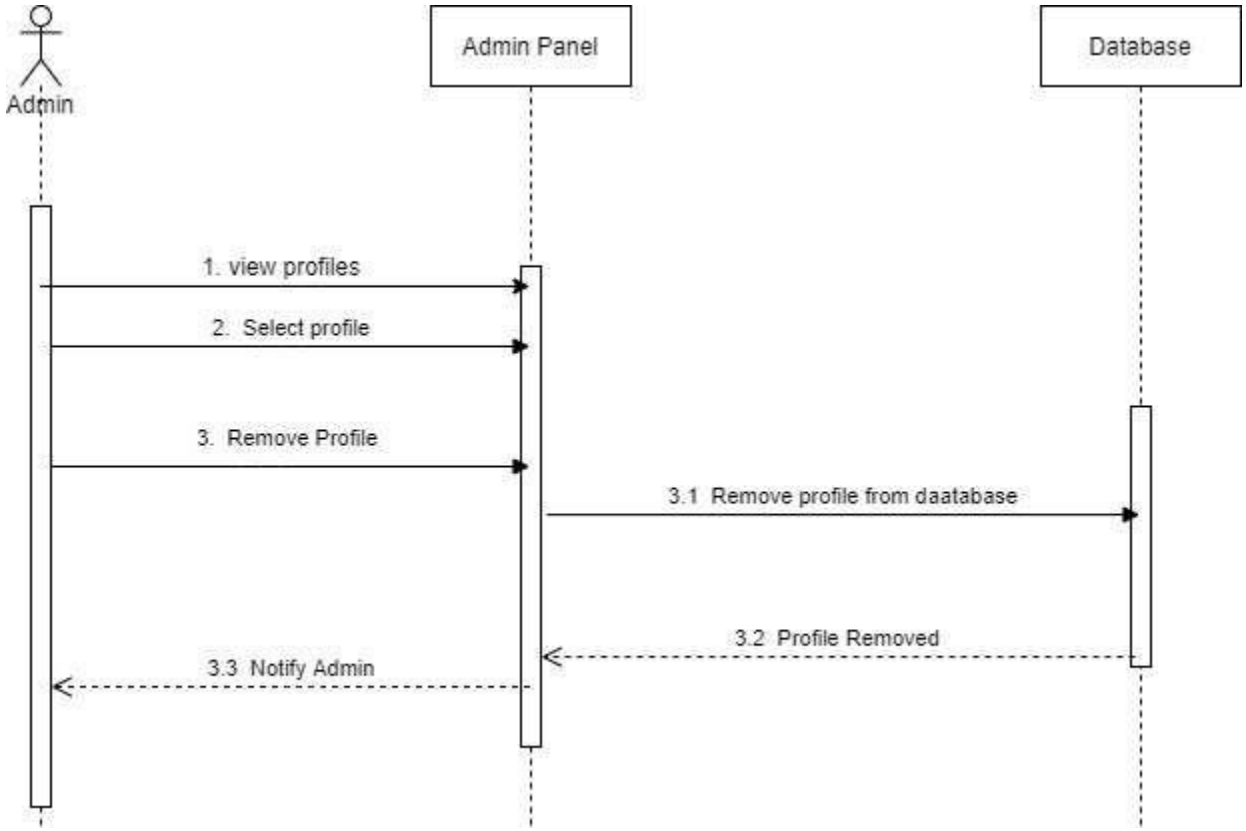

Figure 44- SQ 15

3.3.16 Accept Order SQ16

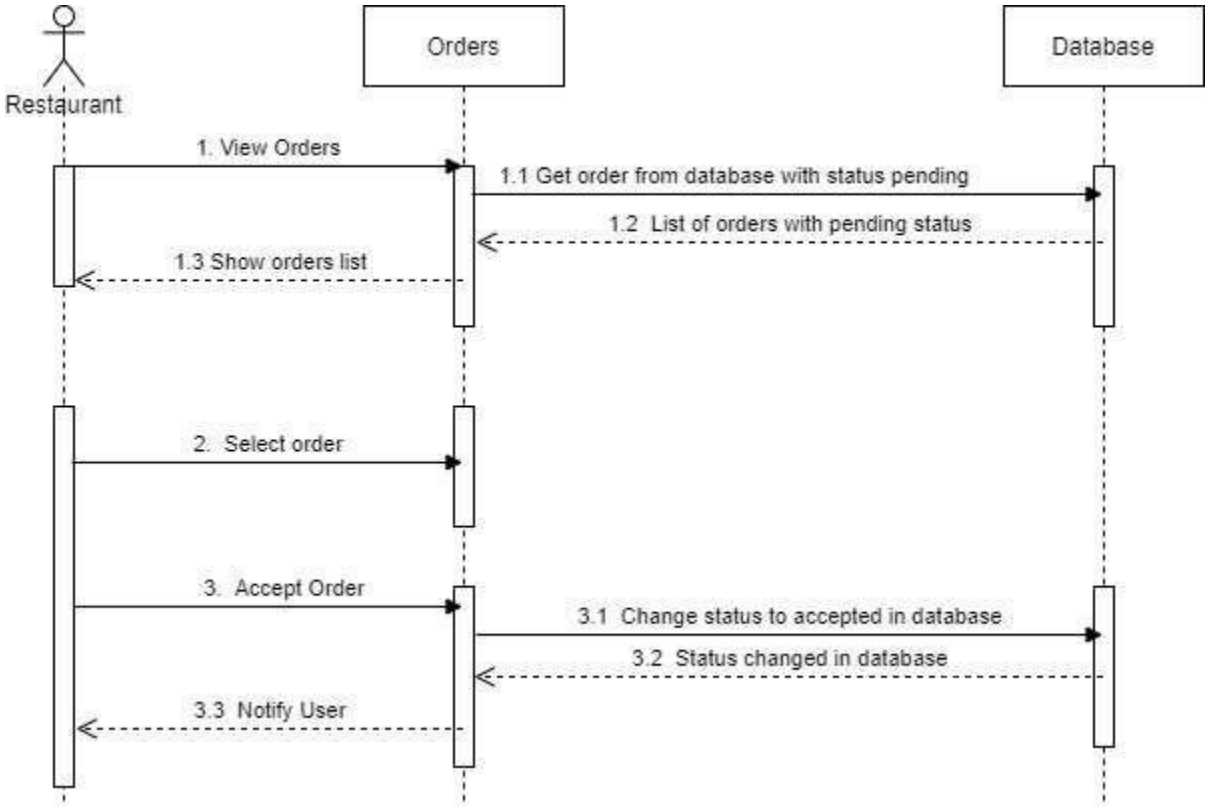

Figure 45- SQ 16

3.3.17 Cancel Order SQ17

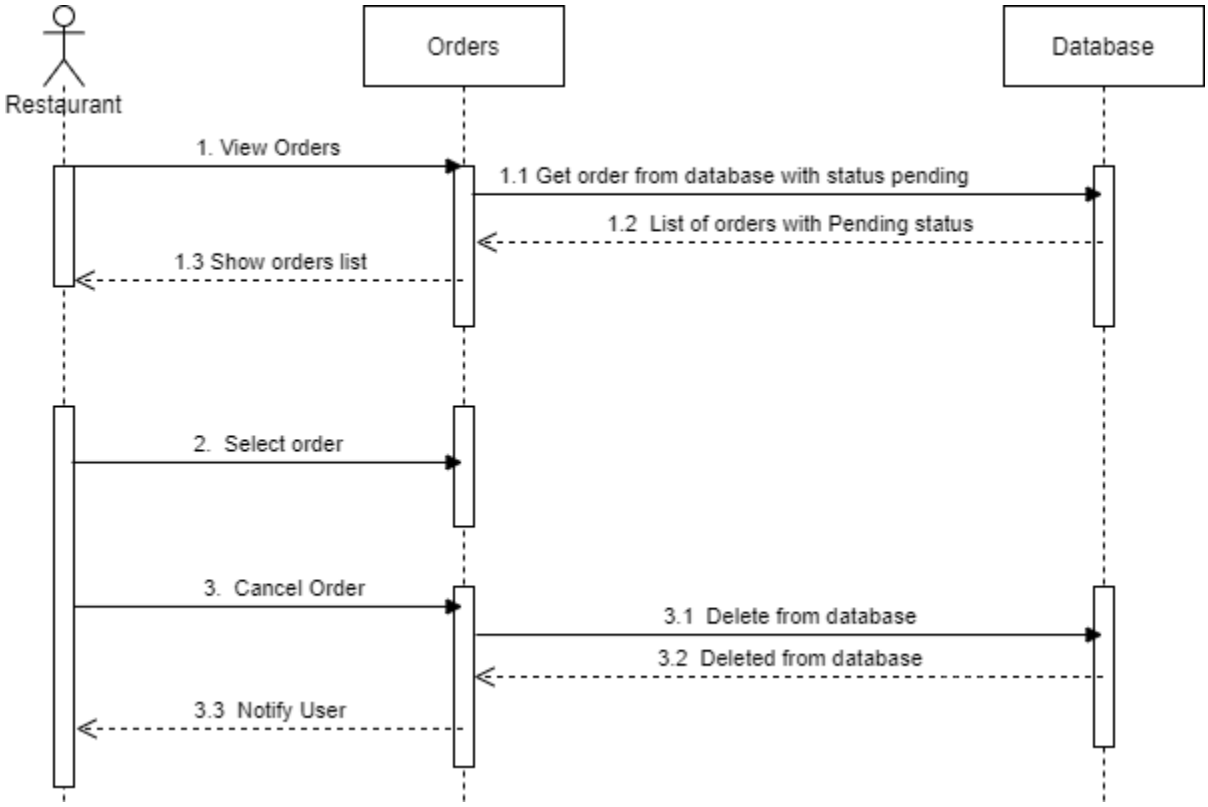

Figure 46- SQ 17

3.3.18 Add Menu ItemSQ18

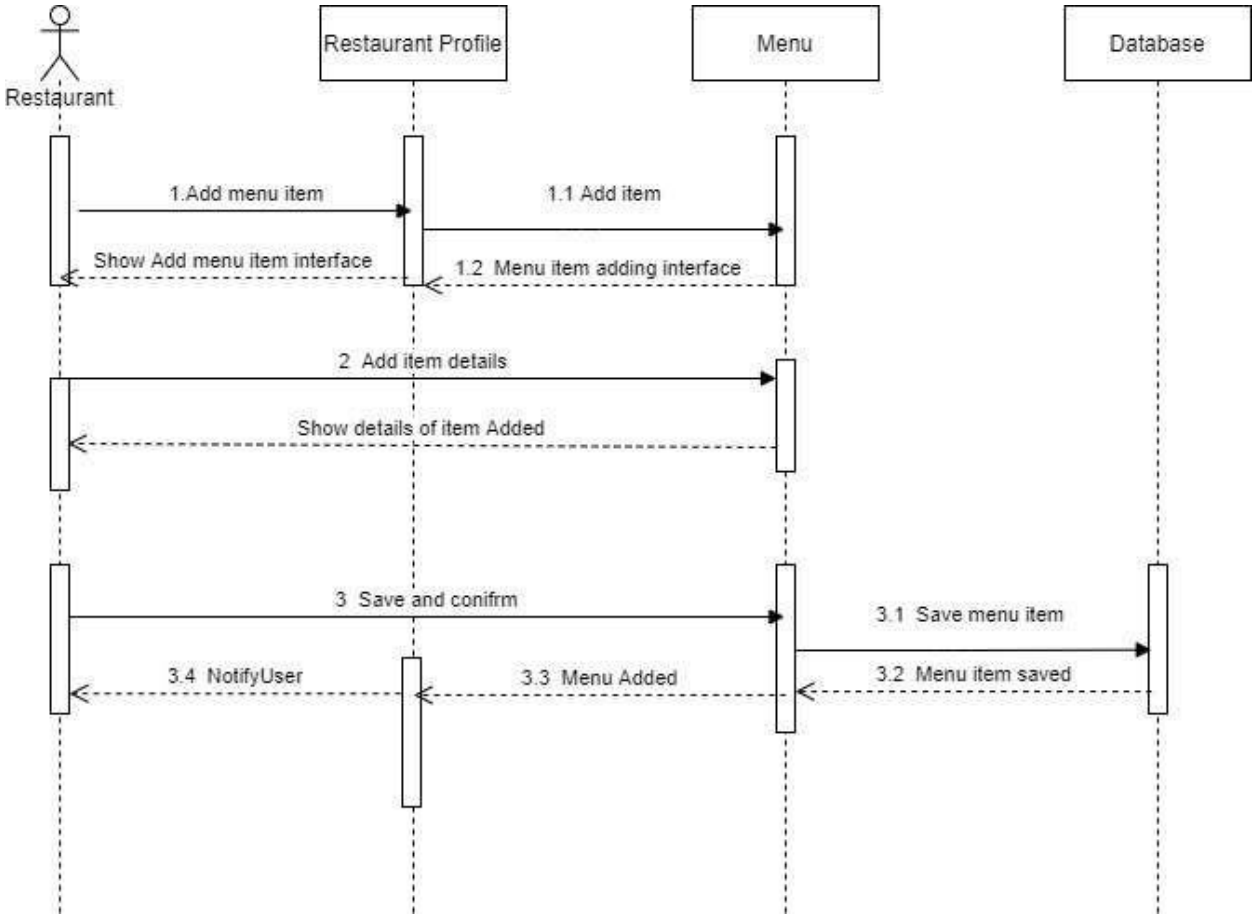

Figure 47- SQ 18

3.3.19 Edit Menu SQ19

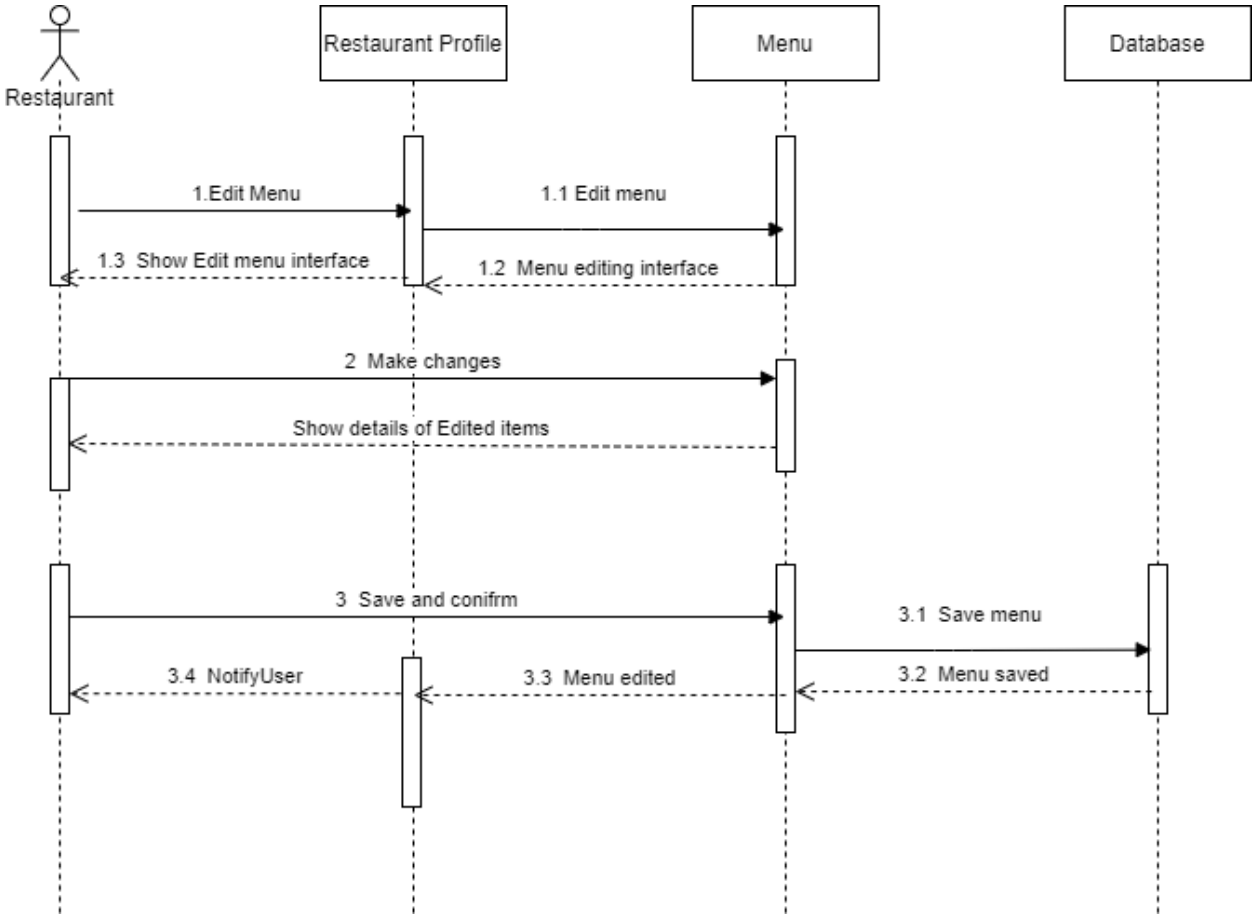

Figure 48- SQ 19

3.3.20 View Pending Order SQ20

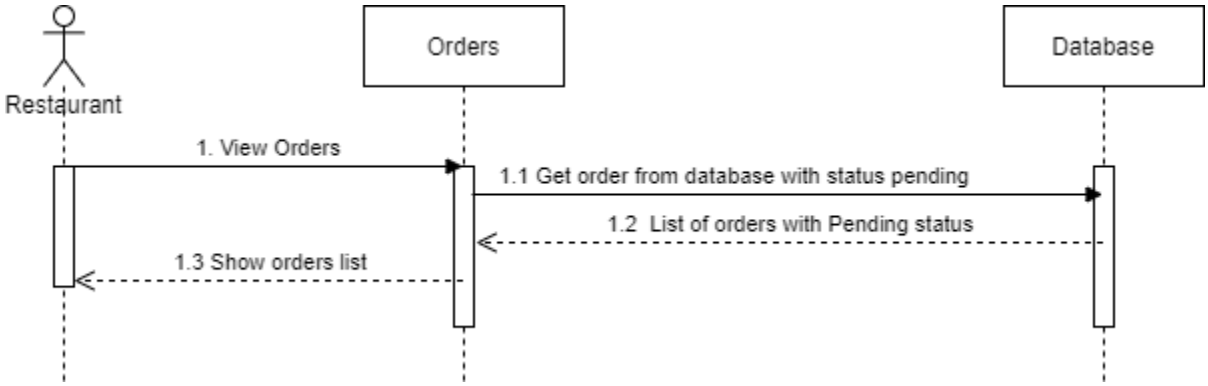

Figure 49- SQ 20

3.3.21 Dispatch Order SQ21

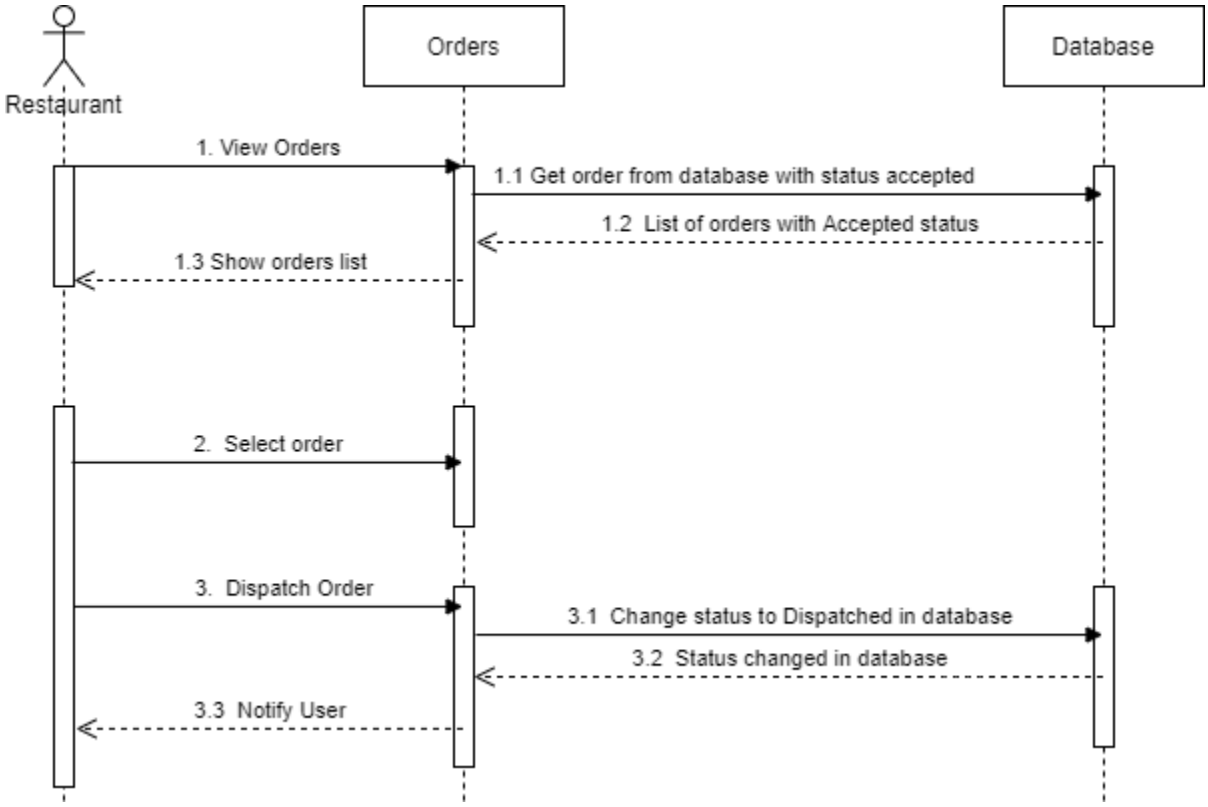

Figure 50- SQ 21

3.3.22 View Analytics SQ22

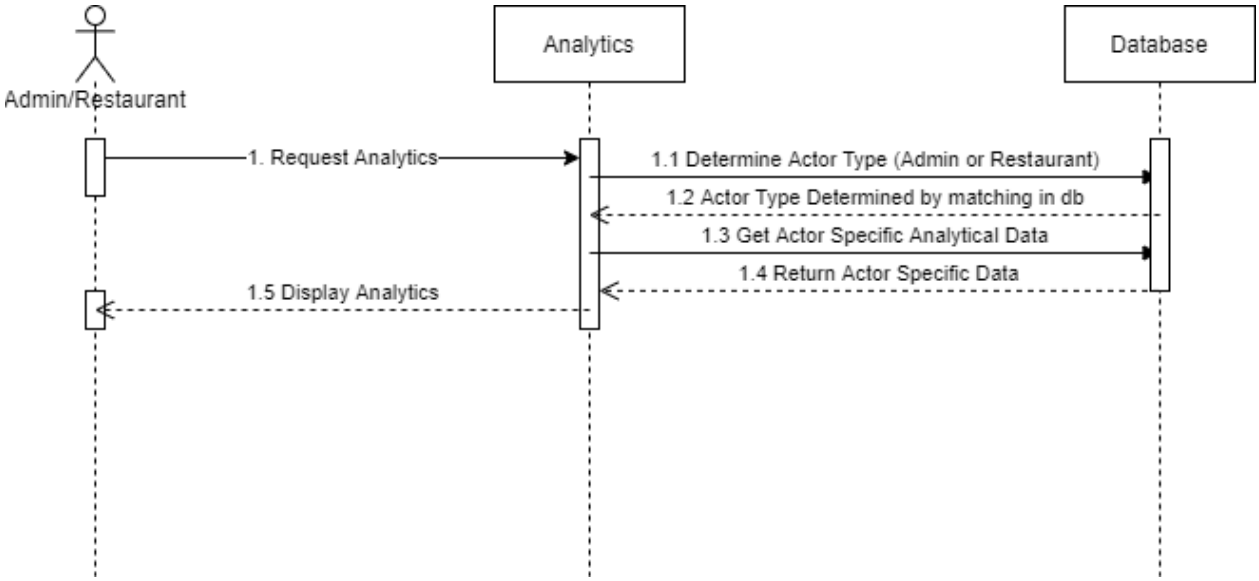

Figure 51- SQ 22

3.3.23 Manage Stream SQ23

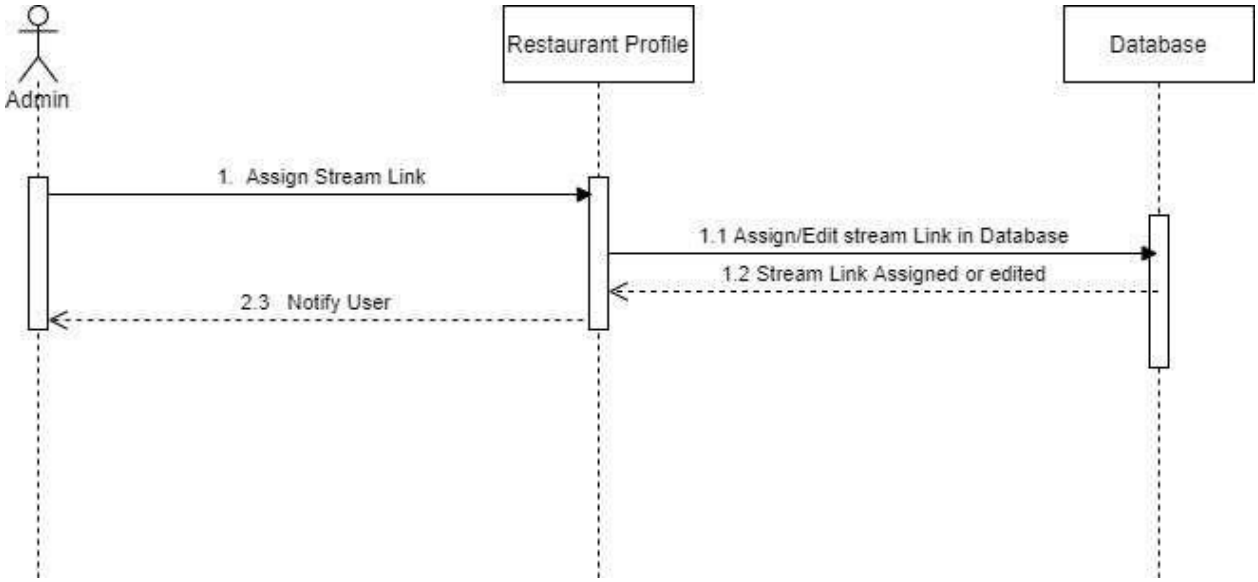

Figure 52- SQ 23

3.3.24 View Live-Stream SQ24

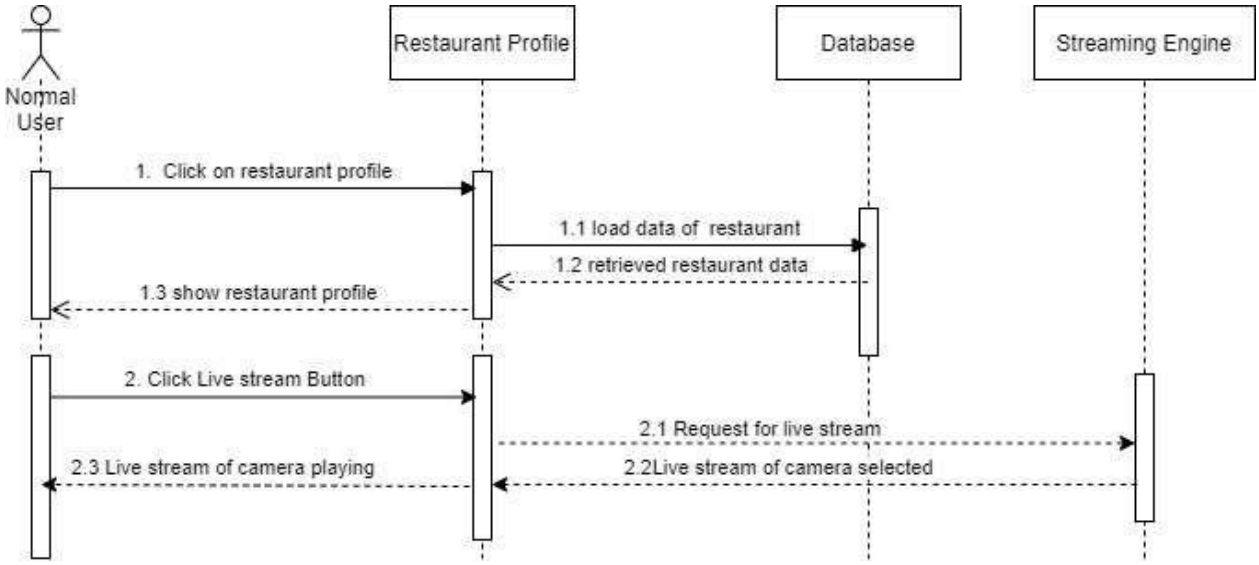

Figure 53-SQ24

3.4 Software Architecture

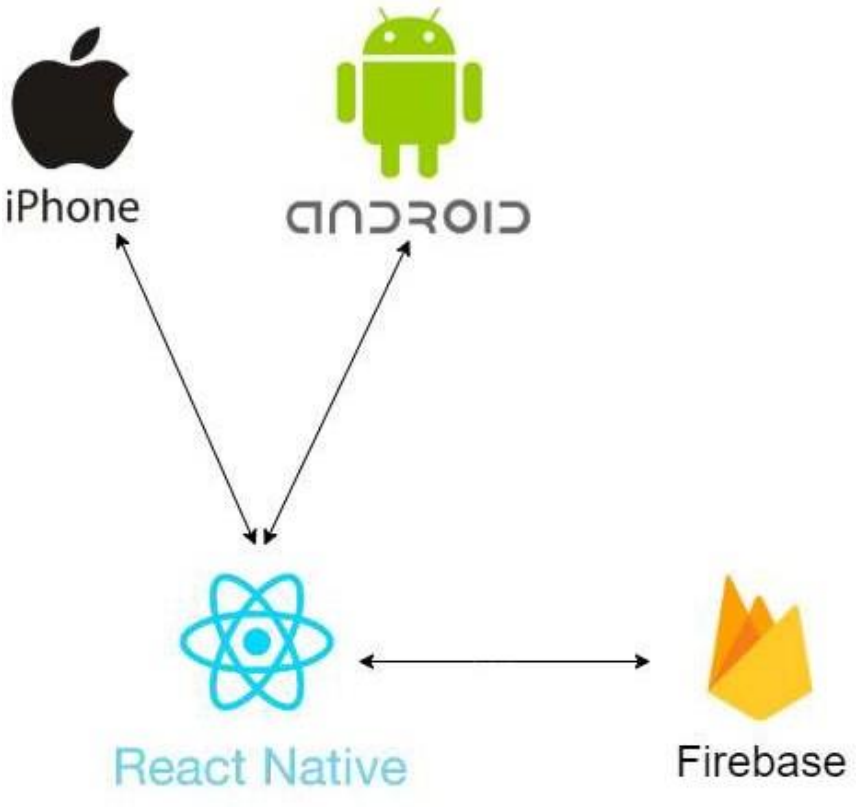

Figure 54- Software Architecture

3.5 Class Diagram

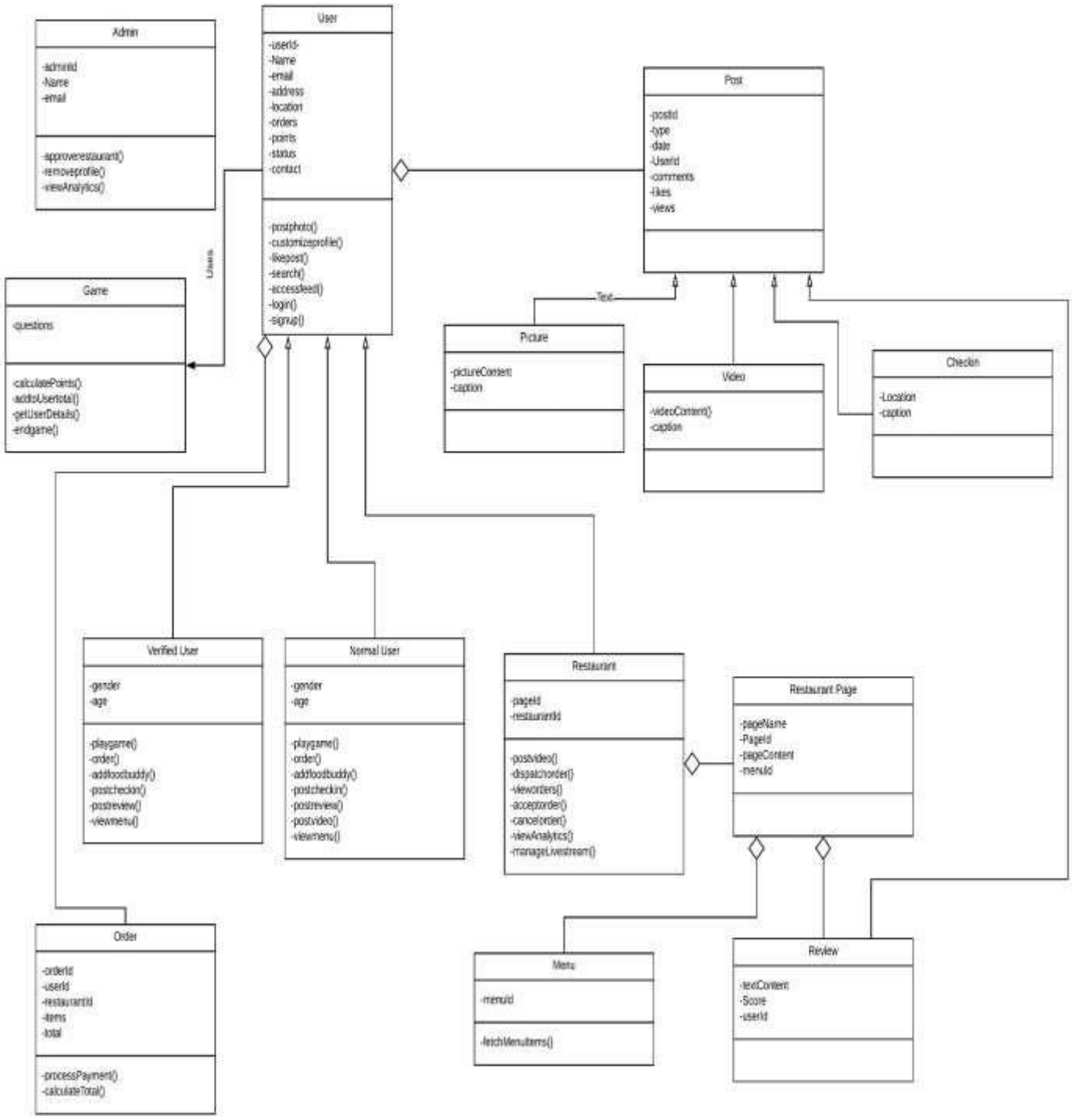

Figure 55- Class Diagram

3.6 Database Diagram (ERD)

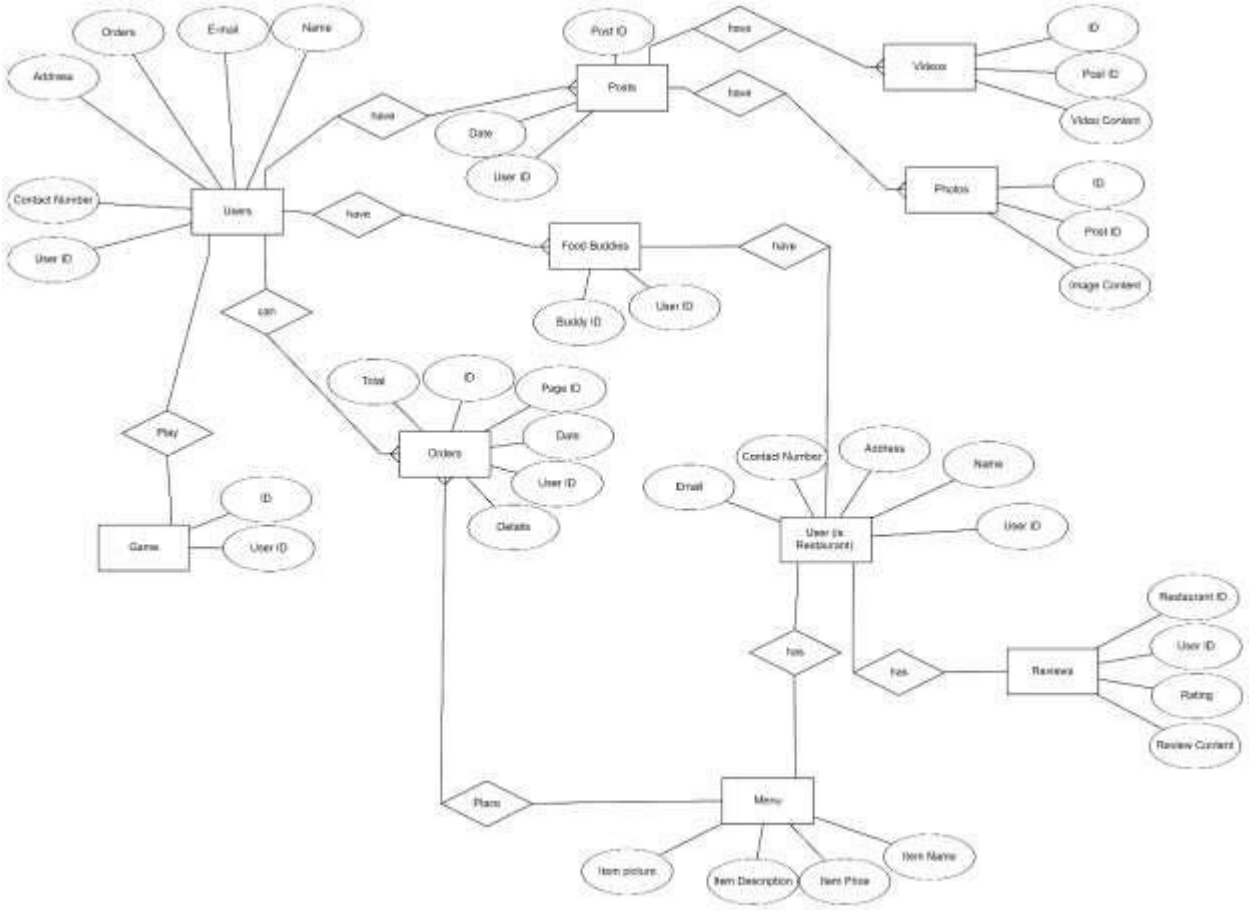

Figure 56- ERD (Database Diagram)

3.7 Network Diagram (Gantt Chart)

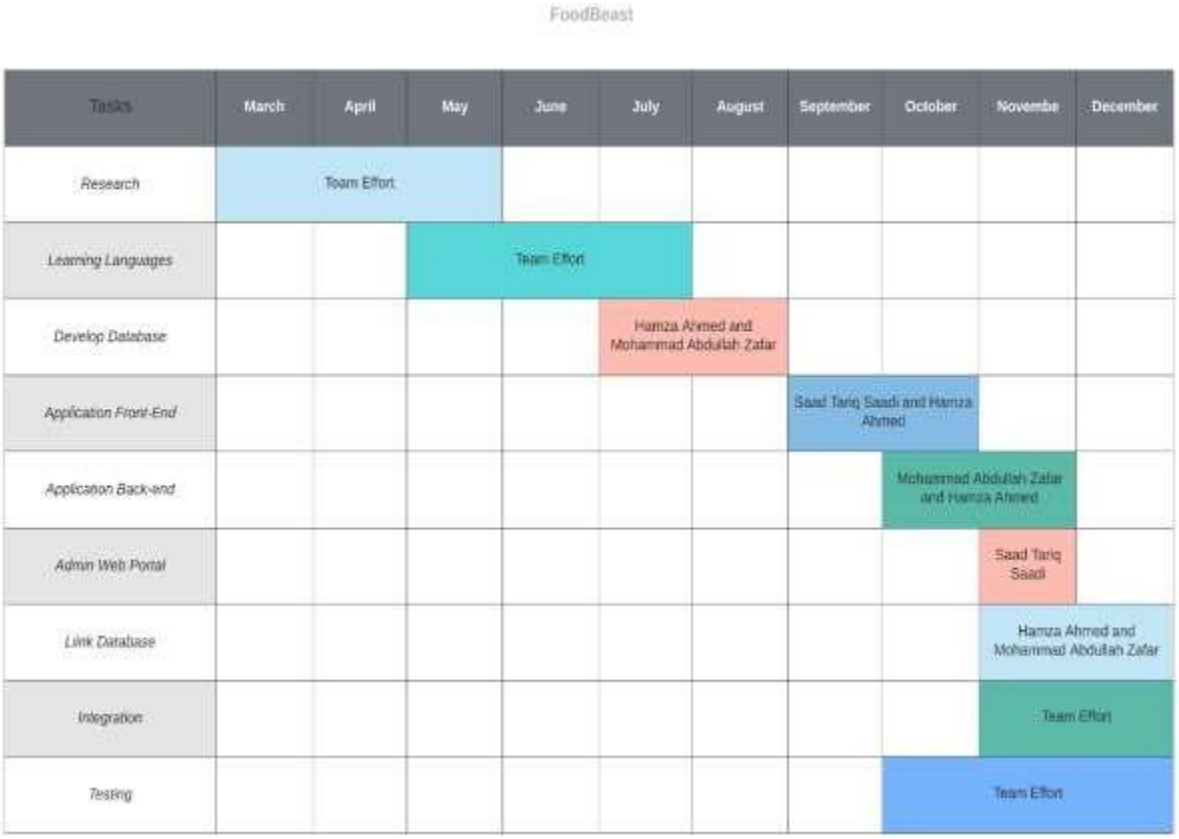

Figure 57- Network Diagram (Gantt Chart)

4 Chapter 4: System Testing

4.1 Test Cases

1. **Test Case 01: Login**

Description: Testing Functionality of Login

Pre – Condition: User is Registered but not Logged In.

Action 1: User enters email and password.

Response: Details are sent to database to match records.

Action 2: User clicks on login

Response: If the details sent to database do not match the records, application prompts user to enter correct details.

Action 3: User Enters correct details.

Response: Application directs user to their Homescreen.

Post Condition: Homescreen is shown to user.

2. **Test Case 02: SignUp**

Description: Testing Functionality of SignUp

Pre – Condition: User is new to the application and want to SignUp.

Action 1: User enters email, password, address, gender, name and contact.

Response: Details are sent to database to match records.

Action 2: User clicks on SignUp

Response: If the details sent to database match the records, application prompts user to enter unique email.

Action 3: User enters a unique email.

Response: Application matches the two passwords required.

If the passwords match user is directed to login.

If the passwords are not matched user is prompted to provide correct passwords.

Post Condition: User needs to verify Email through link sent to the provided email after an alert is shown to user.

3. Test Case 03: Customize Profile

Description: Testing Functionality of Customize Profile

Pre – Condition: User (Normal User/ Restaurant) is logged in the application.

Action 1: User clicks the profile button.

Response: Application opens the user profile.

Action 2: User clicks the Customize profile button.

Response: Application opens the customize profile page.

Action 3: User make changes.

Response: Changes shown on the application user interface.

Action 4: User clicks the Save button.

Response: Application updates the user profile.

Post Condition: User profile has been updated.

4. Test Case 04: Order

Description: Testing Functionality of Ordering food from a restaurant.

Pre – Condition: User is logged in the application and is a normal user.

Action 1: User clicks on restaurant profile.

Response: Application shows the desire page.

Action 2: User clicks on menu button.

Response: Application shows the desire page.

Action 3: User selects items from the menu and add to cart.

Response: Application adds the items in the cart.

Action 4: User clicks on checkout.

Response: Application shows the desire page.

Action 5: User Confirms checkout

Response: Application sends the order request to the restaurant.

Post Condition: Order has been placed and waiting to be accepted.

5. Test Case 05: Accept/Cancel Order

Description: Testing Functionality of Accepting/ Cancelling order.

Pre – Condition: User is logged in the application and is a Restaurant.

Action 1: User clicks on Orders button.

Response: Application shows the desire page.

Action 2: User clicks on Pending orders button.

Response: Application shows the desire page.

Action 3: User selects orders to accept/cancel.

Response: Application shows the selection on user interface.

Action 4: User clicks on accept/cancel order button.

Response: Application changes status of order from pending to accepted and moves the order from pending orders list to orders list

OR

deletes the request from database and removes from the list.

Post Condition: Order has been accepted and waiting to be dispatched OR cancelled and deleted.

6. Test Case 06: Post Photo/Video for Restaurant

Description: Testing Functionality of posting photo/video.

Pre – Condition: Restaurant is logged in the application.

Action 1: User clicks on post button.

Response: Application requests to add photo/video.

Action 2: User clicks on selects items from device.

Response: Application shows attached files on user interface.

Action 3: User clicks on post button.

Response: Application posts photo/video on the user profile and updates the database.

Post Condition: Photo/video has been posted successfully.

7. Test Case 07: Post Photo/Video/Review/Check-In for Normal User

Description: Testing Functionality of posting Photo/Video/Review/Check-in.

Pre – Condition: Normal User is logged in the application.

Action 1: User clicks on post button.

Response: Application requests to add photo/video/review/check-in.

Action 2: User clicks on selects items from device or type review or select location.

Response: Application shows attached files/typed review/location on user interface.

Action 3: User clicks on post button.

Response: Application posts photo/video/review/check-in on the user profile and updates the database.

Post Condition: Photo/video/review/check-in has been posted successfully.

8. Test Case 08: Dispatch Order

Description: Testing Functionality of Dispatching order.

Pre – Condition: User is logged in the application and is a Restaurant.

Action 1: User clicks on Orders button.

Response: Application shows the desire page.

Action 2: User selects orders to dispatch.

Response: Application shows the selection on user interface.

Action 3: User clicks on dispatch order button.

Response: Application changes status of order from accepted to dispatched and moves the order from accepted orders list to completed orders list.

Post Condition: Order has been dispatched successfully.

9. Test Case 09: Manage Stream

Description: Testing Functionality of Manage Stream

Pre – Condition: Admin is Logged In.

Action 1: Admin clicks on Manage stream button.

Response: Application shows the desired page.

Action 2: Admin selects restaurant to assign/edit stream link.

Response: restaurant details are shown

Action 3: Admin clicks on Assign/edit button

Response: Application make the desired changes.

Action 4: User clicks on save button.

Response: Application saves the changes.

Post Condition: Stream has been managed successfully.

10. Test Case 10: Edit Menu

Description: Testing Functionality of editing menu.

Pre – Condition: User (Restaurant) is Registered and Logged In.

Action 1: User clicks on profile button.

Response: Application shows the desired page.

Action 2: User clicks on Edit Menu Item button.

Response: Application directs user to edit item page.

Action 3: User make changes.

Response: Application shows changes on user interface.

Action 4: User clicks on save button.

Response: Application saves the changes.

Post Condition: Menu Item has been updated successfully.

4.2 Unit Testing

The basis of unit testing is to perform the test from the help of developers for each & every components or modules. The main advantage

of it is to make sure minimum cost is spent on each bug fixes by identifying it in the early phase.

The modules to be tested:

• Modules of app

• Modules of website

• Login

• Ordering

• Posting

4.3 Integration Testing

The main aspect of integration test is to perform the test of the integration between components.

The next phase after unit testing is the integration testing. The entire functionality of the application will be tested through integration test.

The modules to be tested during this phase:

• Confirmation or Denial of Order from restaurant by the user.

• Changes in database if users make any changes.

4.4 Acceptance Testing

The aim is to provide remarkable features on one go from the team. The application will be given to different audience sets after the testing.

Audience feedback is very important for perfect user experience and for the deployment of

application. The application will be launched into the market after positive review from the audience sets by performing acceptance test.

5 Conclusion

5.1 Problems Faced and Lessons Learned

The first problem we faced was with the selection of the idea, we had selected an idea which we had to scrap because we submitted the proposal without proper research because we liked the idea soo much but we soon found out that our previous idea would not be feasible. So we ended up changing the proposal to what we are working on now. The lesson learned from this was to fully realize and thoroughly research an idea before you commit to it.

The Problems we faced at the very start were actually selecting the language that we wanted to use to build our project and choose the correct framework. There were too many options and that confused us at the start but we knew that we wanted to work with an emerging and powerful language and framework, since we had plans to make the application cross platform, the best choices of frameworks for us were either React Native or Flutter. Flutter used dart language and was developed by Google while React Native was developed by Facebook and used the JavaScript Language, after much deliberation, we finally decided to use React Native with firebase because React Native is a more mature framework and has a huge community.

The scope of this application is enormous so selecting features to limit the app so that it can be completed within the time frame was hard for us, everyone wants their app to be feature rich so the features we have left out will be implemented sometime in the future. The lesson learned here was that some sacrifices are necessary to make the app function.

During the documentation process, in particular, making the diagrams was quite difficult. While there were a lot of online and offline tools, making a choice to choose one that would best fit our needs proved very confusing and the ones that we really liked ended up being paid. The solution to this was messy as we created multiple trail accounts and used free websites to create diagrams which was difficult to organize at the start. While we did end up making all the diagrams by using LucidChart, Creately and Draw.io, it would’ve been much easier and faster to manage and develop if we had developed all the diagrams at one place. The lesson learned here is that it would be better to pay a small amount of fee to get license for a diagramming software.

The first problem we faced was with our choice to work with firebase. Firebase is a No-Relation, document-based cloud database. This means that what we had learned all the way up till now in the university and database course in specific was not applicable to it since there was no primary key and foreign key

relationships as is the case with relational databases. So to structure our data in a way where it is the most efficient use of the available space and it provides all the necessary attributes for different collections to easily access (collections are like tables in firebase) was a challenge and we learned quite a few tricks and overcame it.

One of the biggest challenges we faced was in our implementation of livestreaming. We had worked on our application for a while now and were quite ahead in the development process. We used the Expo SDK for react native with managed workflow because it was the most recommended by experts for beginners. When it came time to start implementing one of our main features which was livestream, we were horrified to learn that Expo isn't compatible with any RTSP or RTMP libraries. These are the libraries which are used for livestreaming and livestreaming would not be possible without them. We were stuck here for quite a while before we learned a new trick. That was the use of WebView which can render HTML content in our app. Through that and twitch and OBS, we overcame this hurdle and were successful in implementing the livestream. We learned a lot from this experience like how we should think of the long term implications of our choices in any given project.

5.2 Project Summary

The world has become very digital, almost everyone has a smart phone these days. The love of food that Pakistanis have for food is also second to none. We observed that there was no platform available that would bring users with the love of food together on a social media like platform while simultaneously offering the user the option to order food directly from that app. With FoodBeast, we provide this platform to users and restaurants and attempt to create a social eco system that revolves around people with love for food.

Our Application will allow its users to order from restaurants, it will allow restaurants to host menus and manage their orders and provide all of them with a food focused social media platform which will enable users post pictures and reviews about restaurants.

The user only has to install one app that is our app for all things related to food. He/she will be able to become a part of the community and will be able to access restaurants and order all kinds of foods.

5.3 Future Work

There were so much to cover into our application but we had constraints of limited time that's why we have integrated only the core features in the applications. Some works which we will consider in the future are listed down below.

- Website implementation
- Enhance User Interface
- Customized order feature
- Ability to reserve table at restaurants
- AI assisted restaurant suggestion

6 References

[1] “ Yelp.” https://www.yelp.com/. Accessed 29 Jul. 2020.

[2] "OpenTable." https://www.opentable.com/. Accessed 29 Jul. 2020.

[3] "United States - Zomato." https://www.zomato.com/united-states. Accessed 29 Jul. 2020.

[4] "Foodpanda." https://www.foodpanda.sg/. Accessed 29 Jul. 2020.

7 Turnitin Report

12/28/2020 Turnitin

Turnitin Originality Report

Processed on: 28-Dec-2020 18:58 PKT
ID: 1481676460
Word Count: 9189
Submitted: 1

FoodBeast By Hamza Ahmed

Similarity Index	Similarity by Source
13%	Internet Sources: 3% Publications: 0% Student Papers: 11%

11% match (student papers from 24-May-2019)
Submitted to Higher Education Commission Pakistan on 2019-05-24

3% match ()
http://hdl.handle.net/10016/21385

Abstract Welcome to FoodBeast! A mobile application that gives you a live feed of the kitchen where the food is being prepared, a social media platform for foodies where they can post reviews, place orders by choosing from a vast range of their favourite cuisines. FoodBeast enables their users to add their food buddies to their profile and post whatever food related thoughts or media they have on their feeds. Are you really crazy about food? Then write reviews and post food related media. It allows foodbuddies to chat via built in chat. 1. Chapter 1: Introduction 1.1 Introduction Normal users can sign up on the application. They can see all the posts on the feed related to food. He can add food buddies and can check-in to the different restaurants. The user will get the feature of posting videos on feed which can result in influencing the users. The users can place an order through our application. Users can rate the Restaurants from which they have ordered food. Users will be able to view livestream of restaurants. Restaurant will have the same login page as customers but they can't log in until approved by the admin, Restaurants can do all the things that a Normal User can do with the exception of giving reviews to other restaurants and is not able to check-in. The added feature that restaurants has is of hosting their menu on the app. Restaurants can modify their menu as they see fit. Admin will log in through the admin web portal interface, the admin will have the power to add or remove any Restaurant or Normal User. Admins will be able to add other admins. Admin can delete posts made by any user or Restaurant. 1.2 Objectives ? Convenience o One of the main goals of our application is to provide the user with convenience, this is done by providing a multitude of cuisines on one platform from which the user can choose any and place an order through our Application. It also aims to provide convenience in the way that the user will be able to track his/her order through use of maps. ? Security o Our application will be completely secure, this will be done by following various security protocols and measures. We aim to provide our users with peace of mind that their data will be secure. ? Performance o We will ensure the smooth functioning of our application on a multitude of hardware and hardware combinations. Our app will aim to consume as little hardware resources as possible. ? Unique Experience o One of our main goals is to transform the ordinary online food ordering experience into a much more

https://www.turnitin.com/newreport_printview.asp?eqmi1&ebi=1&esm=3&oid=1481676460&sid=0&n=0&m=2&svr=48&r=32.43924717990209... 1/14