Socio-Advisor: Recommender for Social Places

CSP 586 – Software Modelling and Development with UML

Team #10

Singh, Baltej (A20444488)

bsingh18@hawk.iit.edu

Chintakula, Vivek Sumanth (A20457282)

vchintakula@hawk.iit.edu

TABLE OF CONTENTS

1.PROJECT OVERVIEW

The project aims at providing Recommendations for Social Places to a user. The users can search for various food cuisines or directly by the name of the restaurant along with the location where he/she wants to go out.

Once the user searches for the required food/place, he/she gets a list of various places to select. In Addition, the application allows the user to get the information such as the Address, Contact Number, and Ratings for the selected place. The application is designed to display only the places which have a rating that is greater than or equal to 3.5. The application also enables the user to check for the nutritional facts like Calories, proteins per serving weight in grams of the food item selected by the user.

If a user liked a restaurant or a social place, he/she can provide their reviews and rate that social place. The application also allows the logged-in user to view all the places for which he/she has provided reviews or ratings. The application considers all the reviews and ratings provided by a specific user and provides the recommendations on the basis of the user's previous history and the reviews provided.

The recommendations provided for a specific user are generated by creating user and rating interactions.

2.REQUIREMENTS & FEATURES

The System should allow the user to search for food or a place at a specified location and filtering the search criteria of food and place by the user, the system should display all the places/restaurants catering. The system allows the user to provide reviews and ratings for a place. The system provides recommendations to the user based on the previous ratings of the user.

1. Search for different foods or places at a user-specified location.
2. View more information about a selected place viz. Address, Contact Number, and Ratings.
3. Users can provide reviews and ratings for a place of choice.
4. To check for the Nutrients Information for a specifies/searched food item.
5. View the places and their corresponding reviews and rating provided by a user in the past.
6. Provides Recommendations based on history viz. Reviews and Ratings provided for a place.

3.USE CASE DIAGRAM
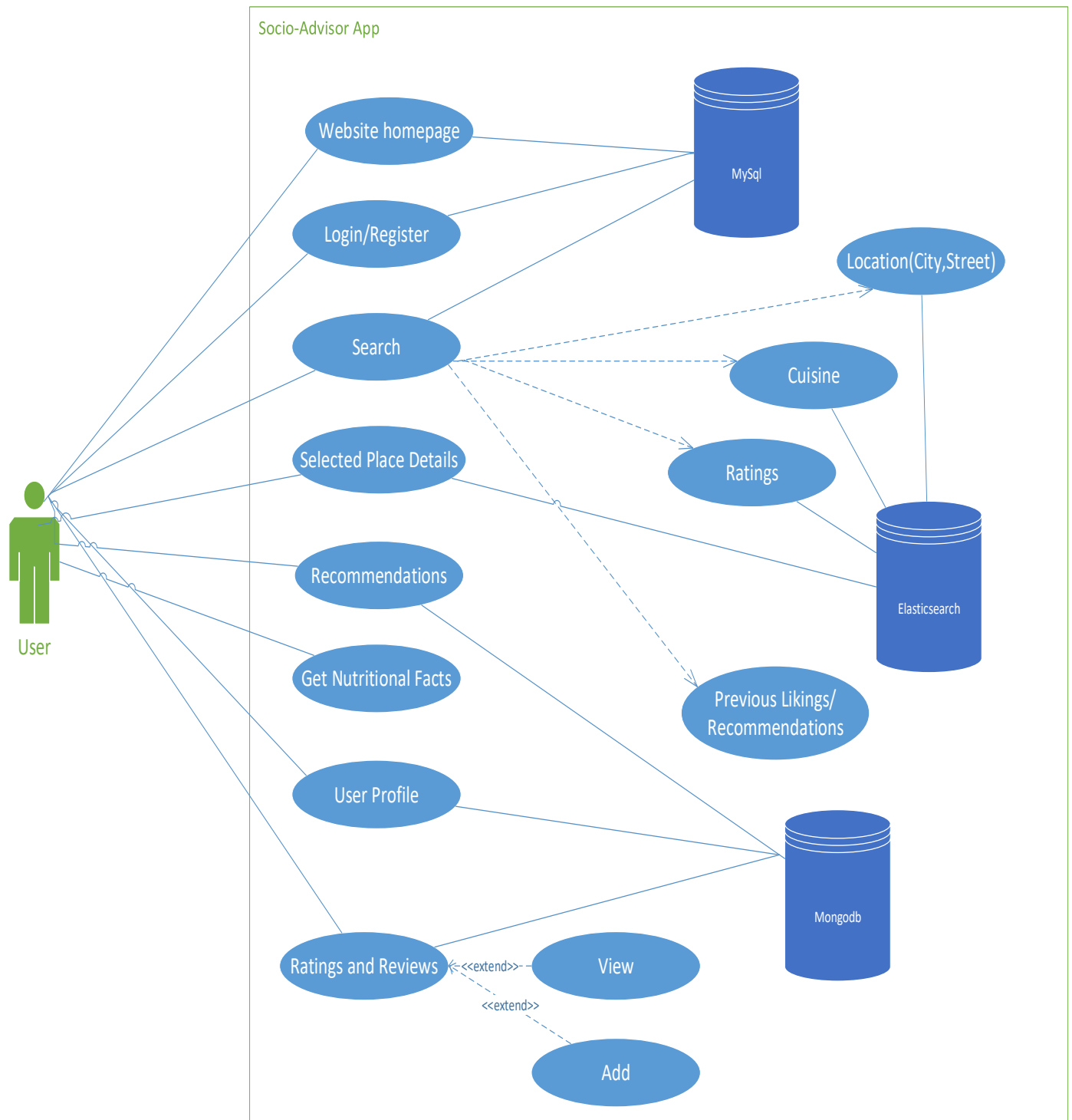
The use cases are defined as tasks executed by the actors to satisfy the requirements supported by the application. They can be defined as a list of actions of event steps typically defining the interactions between a user and a system to achieve a goal.

Use case diagrams are usually referred to as behaviour diagrams, used to describe a set of actions – use cases that some system of systems can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable results to the actors or other stakeholders of the system.

The complete list of use cases is listed below:

1. **User can register and login to the website.**
2. **User can search for a specific food or cuisine at any location.**
3. **User can check the detail of the selected place/restaurant.**
4. **User can check the nutrients for the selected food.**
5. **User can provide reviews and ratings for liked places.**
6. **User can view all the previously provided reviews and ratings.**
7. **User gets Recommendations based on the previous rating for the places.**

4. FULLY DRESSED FORMAT

Use Case Name:	**Register as a new user**	
Triggering Event:	User enters the mandatory fields viz. First name, Email, Password, Confirm Password and hits the register button	
Brief Description:	The user tries to register as a new user to use the application by providing all the mandatory details.	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must enter all the mandatory fields to register	
Post conditions:	New User get successfully registered and is redirected to the login page	
Flow of events:	Actor	System
	• The user enters all the mandatory information to get himself registered as a new user.	• System validates the input information from the UI. • Upon successful validation, the new user gets registered and a corresponding entry is created in the application database
Exception Conditions	User tries to register by not completing all the mandatory fields. Provided passwords do not match. User already exists.	

Use Case Name:	**Login as an Existing User**	
Triggering Event:	User enters the Username (email) and Password and hits the login button	
Brief Description:	The user tries to log in to the application by providing the credentials.	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must enter correct username and password	
Post conditions:	User Logs in successfully and is redirected to the homepage of the application	
Flow of events:	Actor	System
	• The user enters his/her credentials to log in to the application.	• System validates the input credentials from the UI. • System authenticates the user credentials from the database. • Upon successful authentication, the user gets redirected to the application homepage.
Exception Conditions	User tries to login by providing incorrect credentials.	

Use Case Name:	**Search for Places**	
Triggering Event:	User enters the name of food/ cuisine and the location	
Brief Description:	The user searches for a food/cuisine at a specific location and gets a list of places Catering to the same, at that location	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must enter the food item name/cuisine name User must enter the location name where he/she wants to search for the places	
Post conditions:	Search results visible in a tabular content format	
Flow of events:	Actor	System
	• The user enters the search parameters (food & location) and hit search button.	• Searches for the provided Input from the Elasticsearch Database • Shows the search results in tabular format on the UI.
Exception Conditions	User performs a search with one of the Input parameters left empty.	

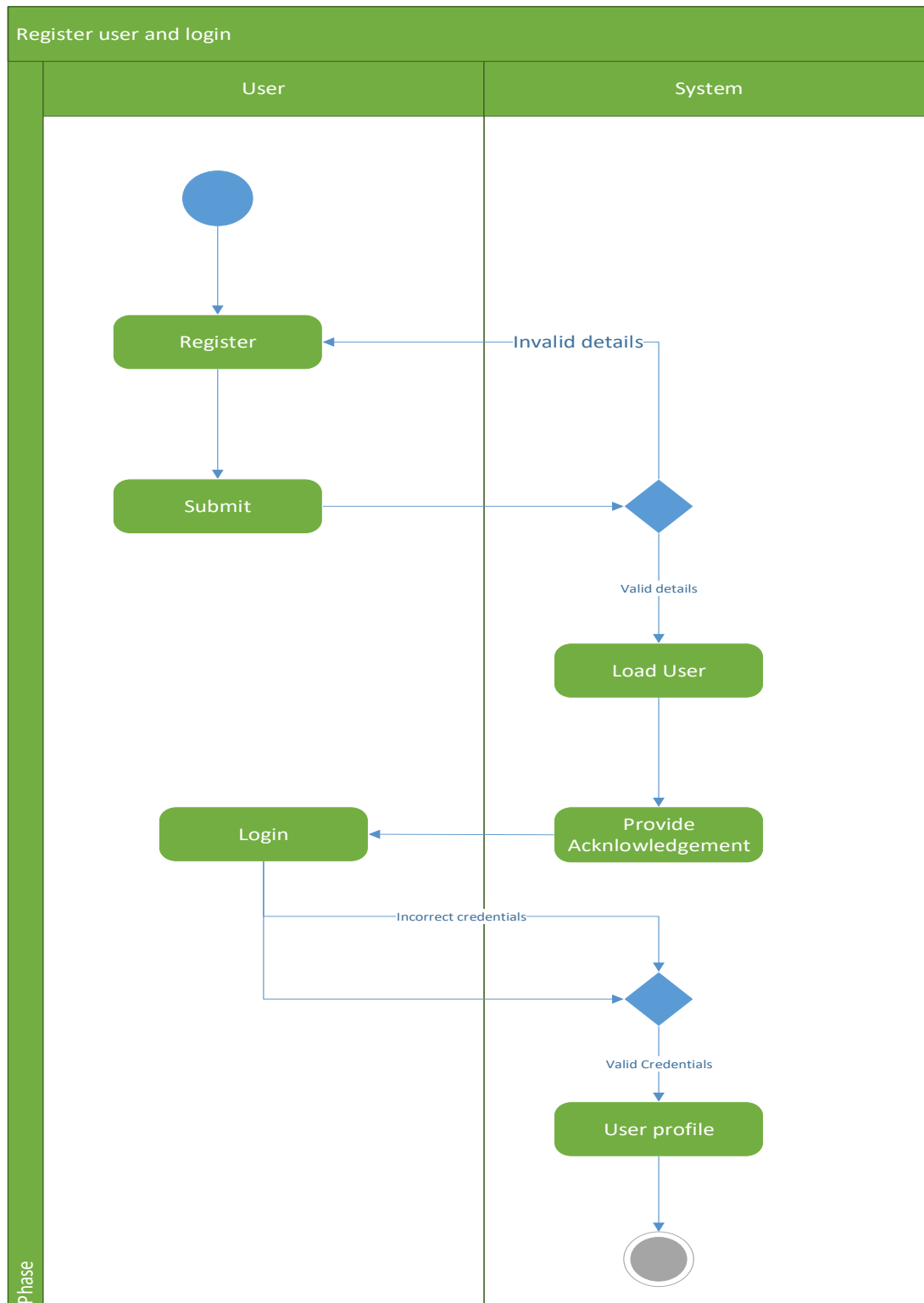| Use Case Name:	**Details of the Selected Place**	
Triggering Event:	User presses the 'More Info' button corresponding to the place of choice	
Brief Description:	The user wants to get more information on a place from the list of places	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must enter the food item name/cuisine name User must enter the location name where he/she wants to search for the places	
Post conditions:	Details of the selected place visible in a tabular content format	
Flow of events:	Actor	System
	• The user presses the 'More Info' button in front of the place of choice.	• Searches for the additional information like Address, contact number and rating of the selected place • Shows the details on the UI.
Exception Conditions	None	

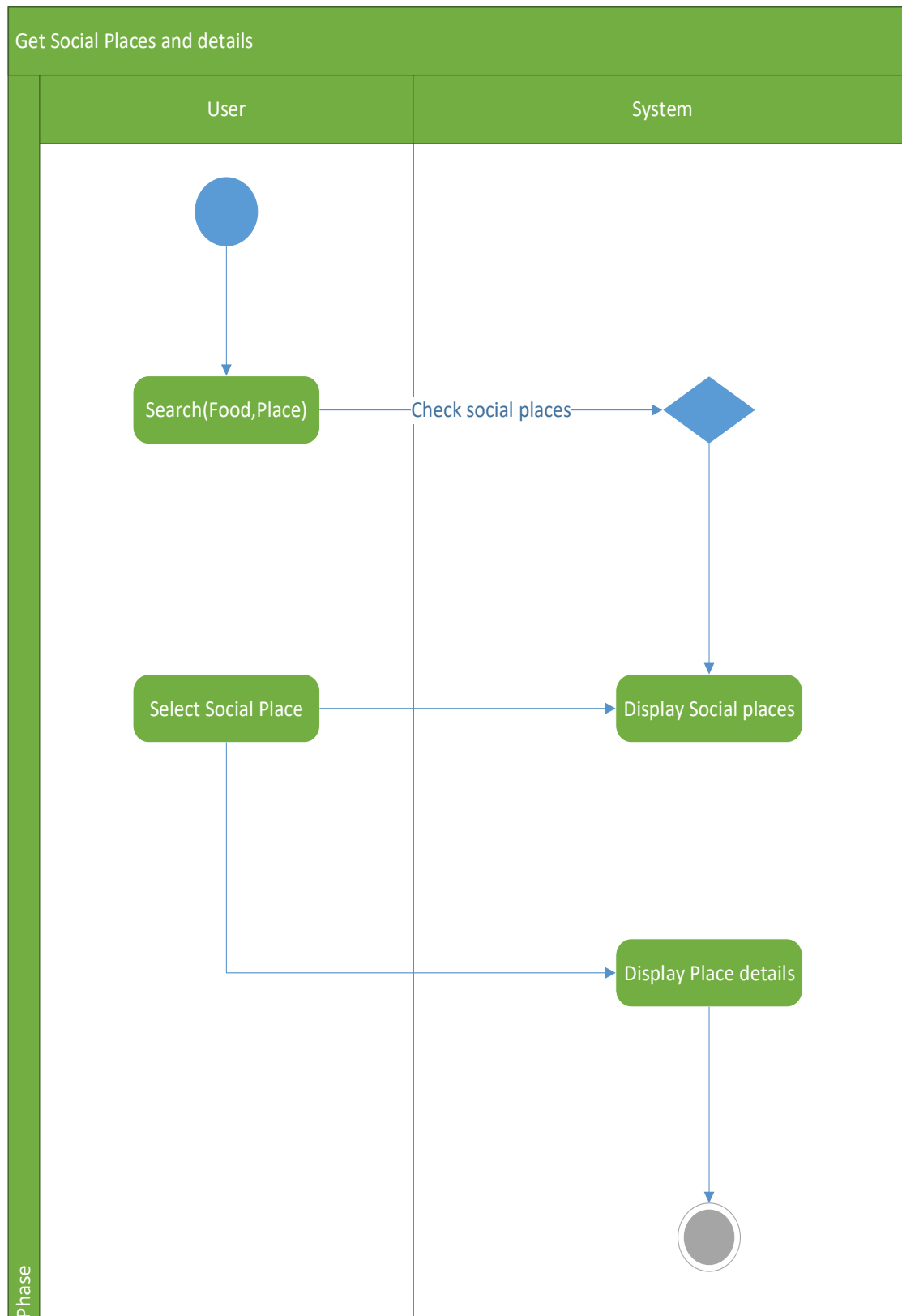| Use Case Name:	**Getting the Nutrition facts**	
Triggering Event:	User presses the 'Get your Food Nutrients' button on top of the place details page	
Brief Description:	The user wants to get Nutrients information of the selected food/cuisine	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must be on the place details place	
Post conditions:	Details of the Nutrients Information of the selected food item	
Flow of events:	Actor	System
	The user presses the 'Get your Food Nutrients' button on top of the place details page.	• Searches for the nutrient's information for the selected food item. • Shows the details on the UI.
Exception Conditions	None	

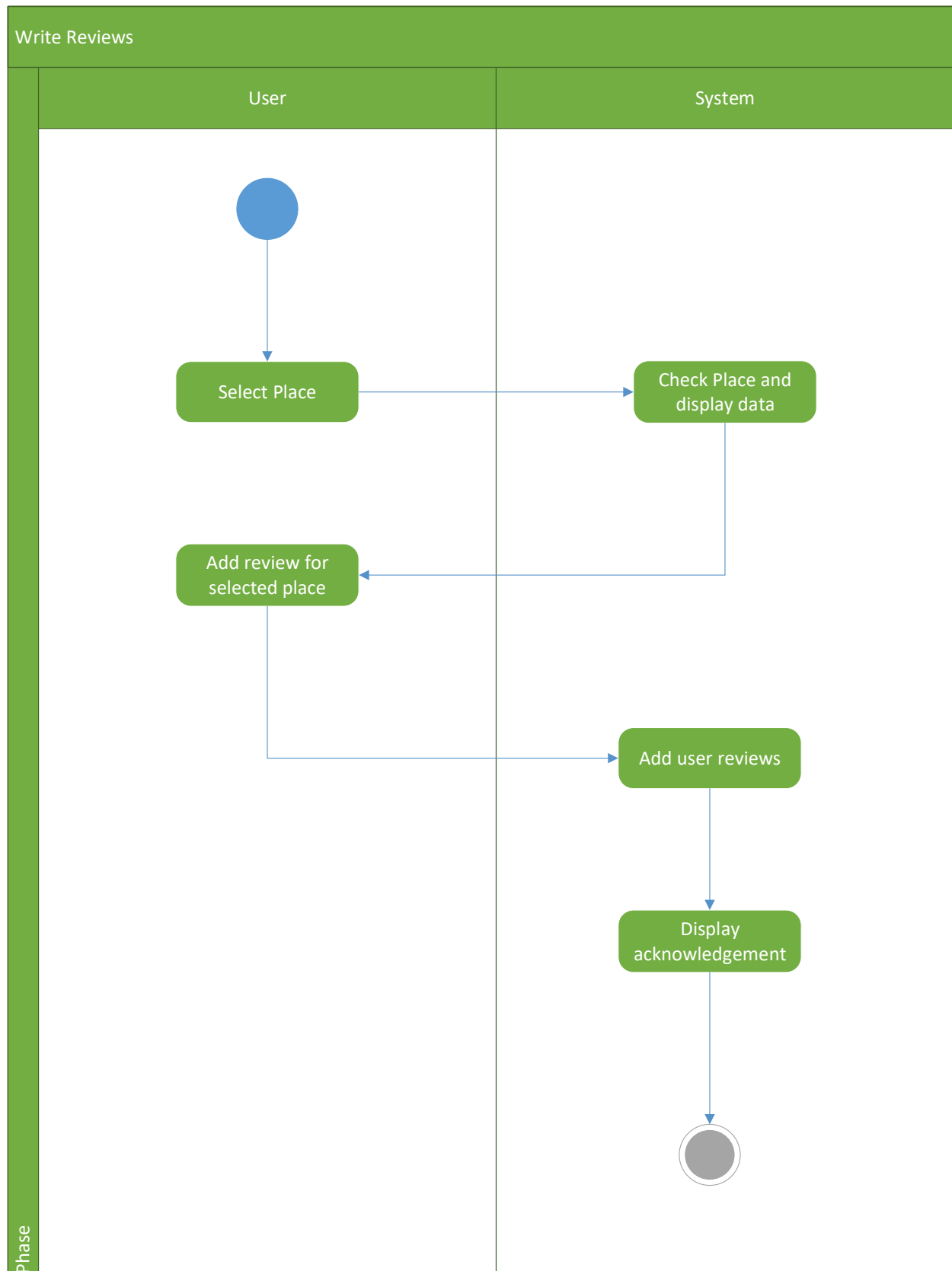| Use Case Name:	**Write Reviews**	
Triggering Event:	User presses the 'Write Review' button on the place details page	
Brief Description:	The user wants to write a review for the selected place	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must be on the place details page for the selected place	
Post conditions:	User is redirected to the write reviews page and gets Input boxes for adding reviews and rating for the selected place	
Flow of events:	Actor	System
	The user presses the 'Write Review' button on the place details page of the selected place.	• Retrieves the Input information added by the user for the reviews and rating for the selected place. • Shows the alert saying 'Review submitted successfully' when submit review button is pressed.
Exception Conditions	None	

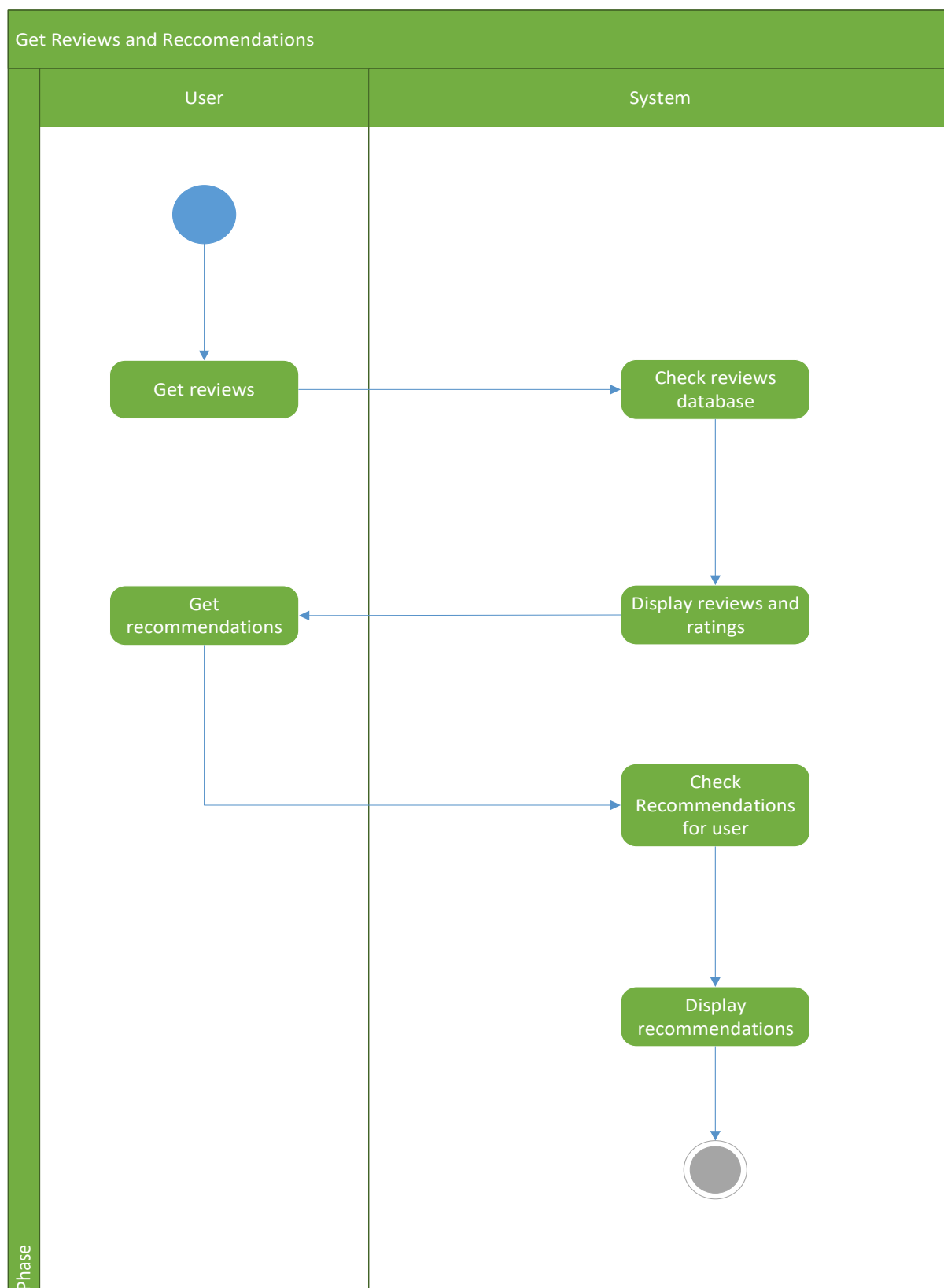| Use Case Name:	**View Reviews**	
Triggering Event:	User presses the 'Reviews' button on the profile dropdown on the top navigation-bar on the user home page	
Brief Description:	The user wants to view all the reviews provided by him/her	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must be logged in to the application to view the reviews	
Post conditions:	User is redirected to the get user reviews page and is able to view all the reviews provided by them in the past	
Flow of events:	Actor	System
	The user presses the 'Reviews' button on the profile dropdown on the top navigation-bar on the user home page.	• Checks from the application database if the current user (logged in user) has provided in any reviews in the past or not. • Displays the list of all the reviews provided by the current user on the UI.
Exception Conditions	If the user has not provided any reviews yet.	

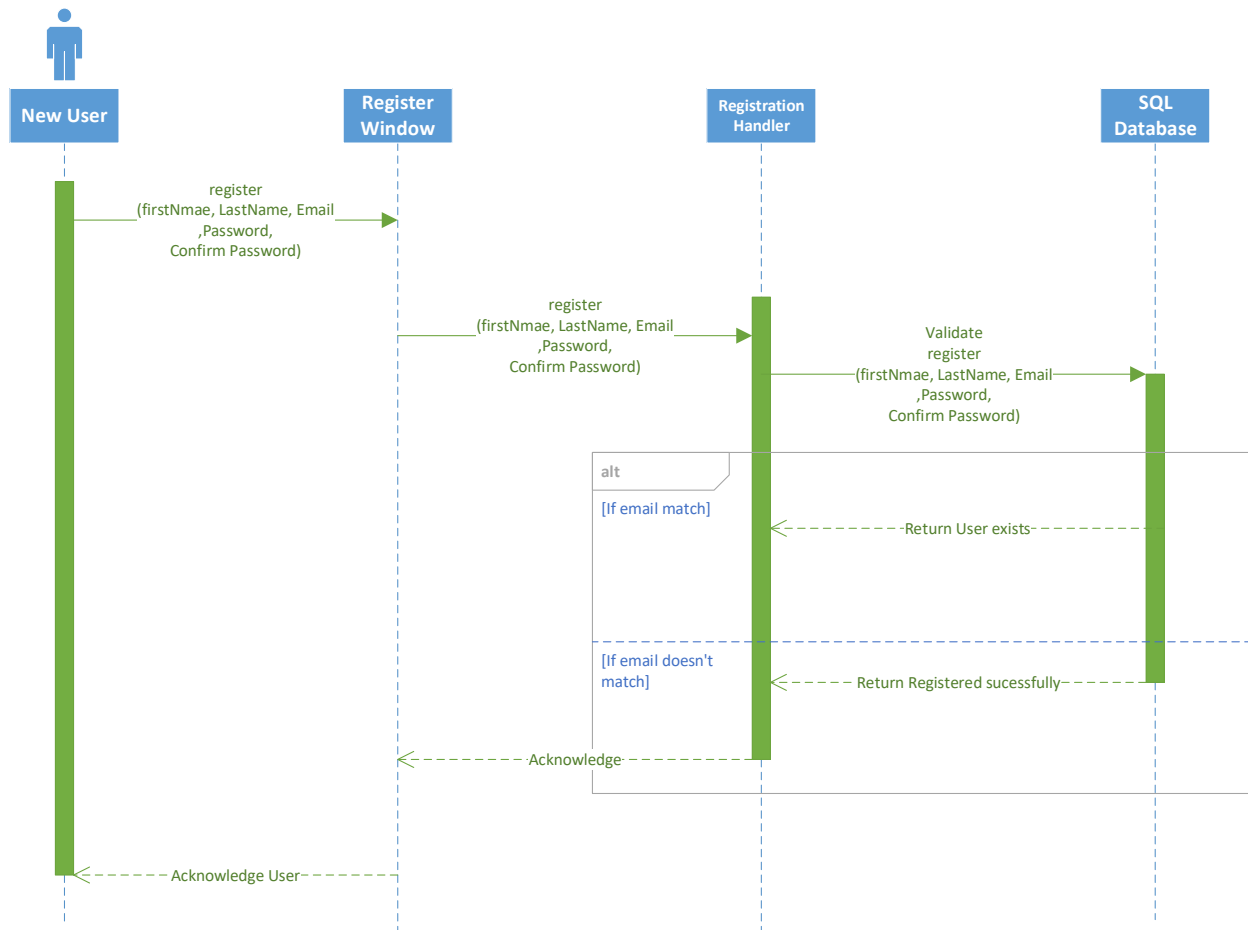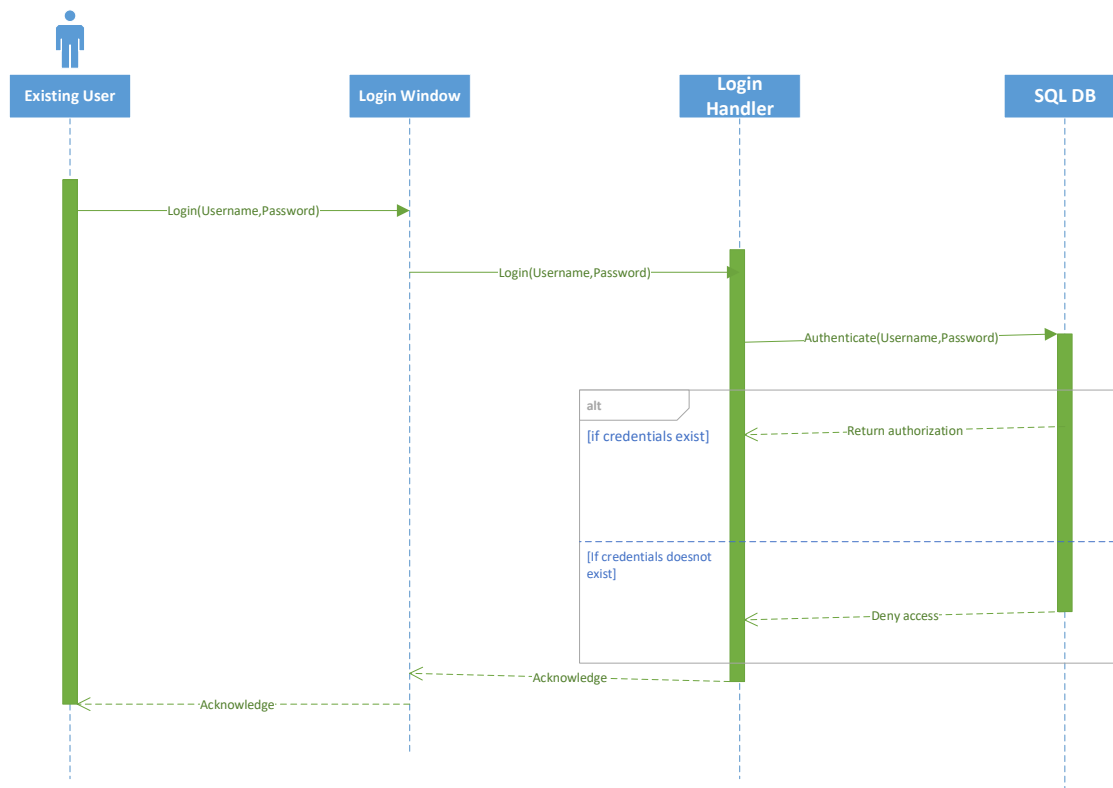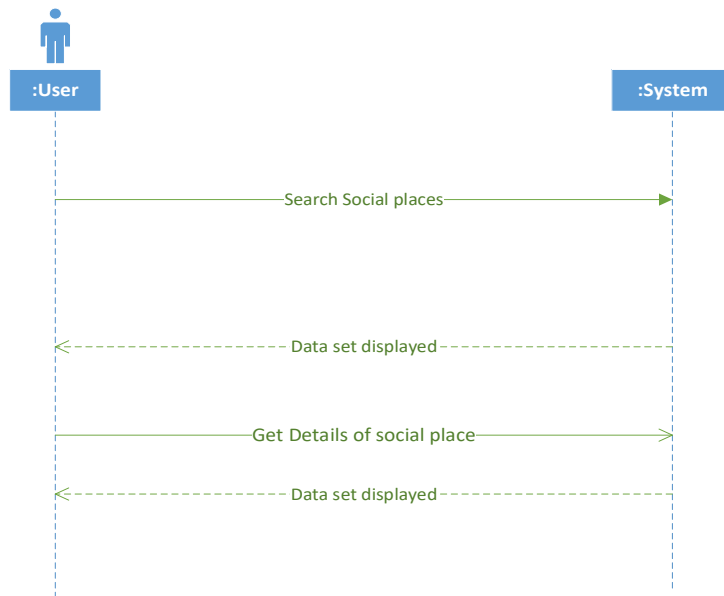| Use Case Name:	**View Recommendations**	
Triggering Event:	User presses the 'Recommendations' button on the profile dropdown on the top navigation-bar on the user home page	
Brief Description:	The user wants to view all the Recommendations for him/her	
Primary Actors:	Users	
Stakeholders:	User	
Preconditions:	User must be logged in to the application to view the Recommendations User must have provided at least 5-8 reviews to get recommendations	
Post conditions:	User is redirected to the Recommendations page and can view all the Recommendations provided by the application	
Flow of events:	Actor	System
	The user presses the 'Recommendations' button on the profile dropdown on the top navigation-bar on the user home page.	• Checks from the application database if the current user (logged in user) has any recommendations or not. • Displays all the Recommendations for the current user on the UI.
Exception Conditions	If the user has not provided enough reviews in the past.	

5.ACTIVITY DIAGRAMS

Register user and login

User	System
Register	Invalid details
Submit	Valid details
	Load User
Login	Provide Acknlowledgement
	Incorrect credentials
	Valid Credentials
	User profile

Phase

Get Social Places and details

User	System
Search(Food,Place)	Check social places
Select Social Place	Display Social places
	Display Place details

Write Reviews

User	System
Select Place	Check Place and display data
Add review for selected place	Add user reviews
	Display acknowledgement

Phase

15

Get Reviews and Reccomendations

User	System

Get reviews → Check reviews database

Check reviews database → Display reviews and ratings

Display reviews and ratings → Get recommendations

Get recommendations → Check Recommendations for user

Check Recommendations for user → Display recommendations

6.SEQUENCE DIAGRAMS

New User Registration

Existing User Login

Existing User → **Login Window** → **Login Handler** → **SQL DB**

Login(Username,Password)

Login(Username,Password)

Authenticate(Username,Password)

alt

[if credentials exist]

Return authorization

[If credentials doesnot exist]

Deny access

Acknowledge

Acknowledge

Get Place Details

:User — **:System**

Search Social places

Data set displayed

Get Details of social place

Data set displayed

Get Nutritional Facts
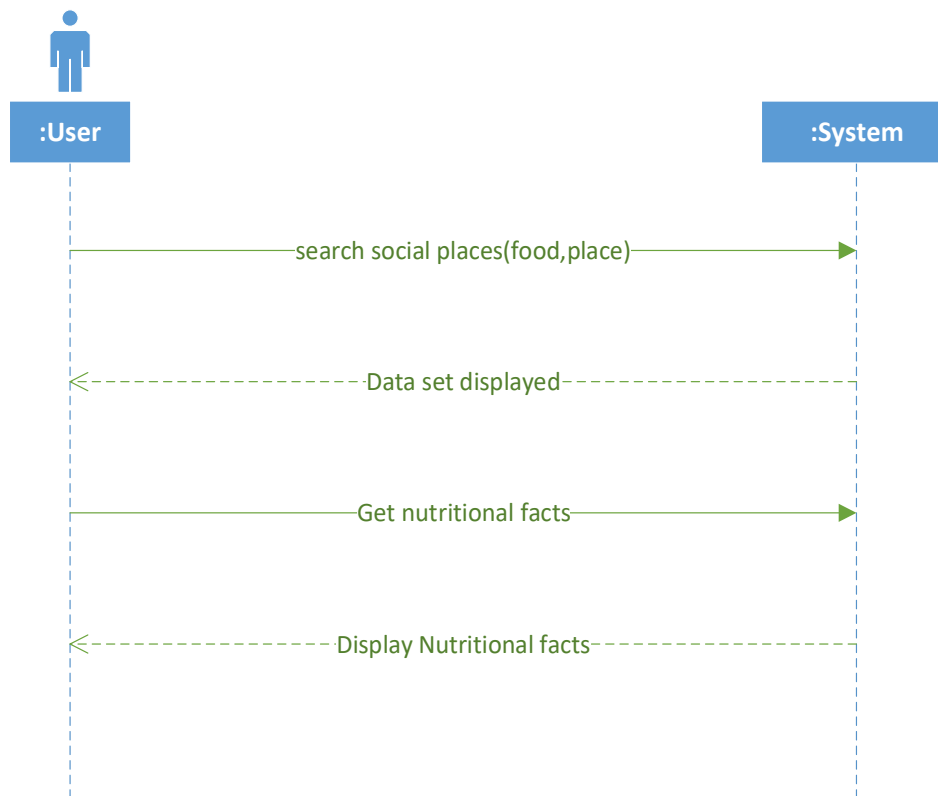
Search Places, Place Details & Add Reviews

Search social places(food,place) →

← Display social places

Get details of a social place →

← Display detail info

Add Review(Review,Rating) →

← Review acknowledgement
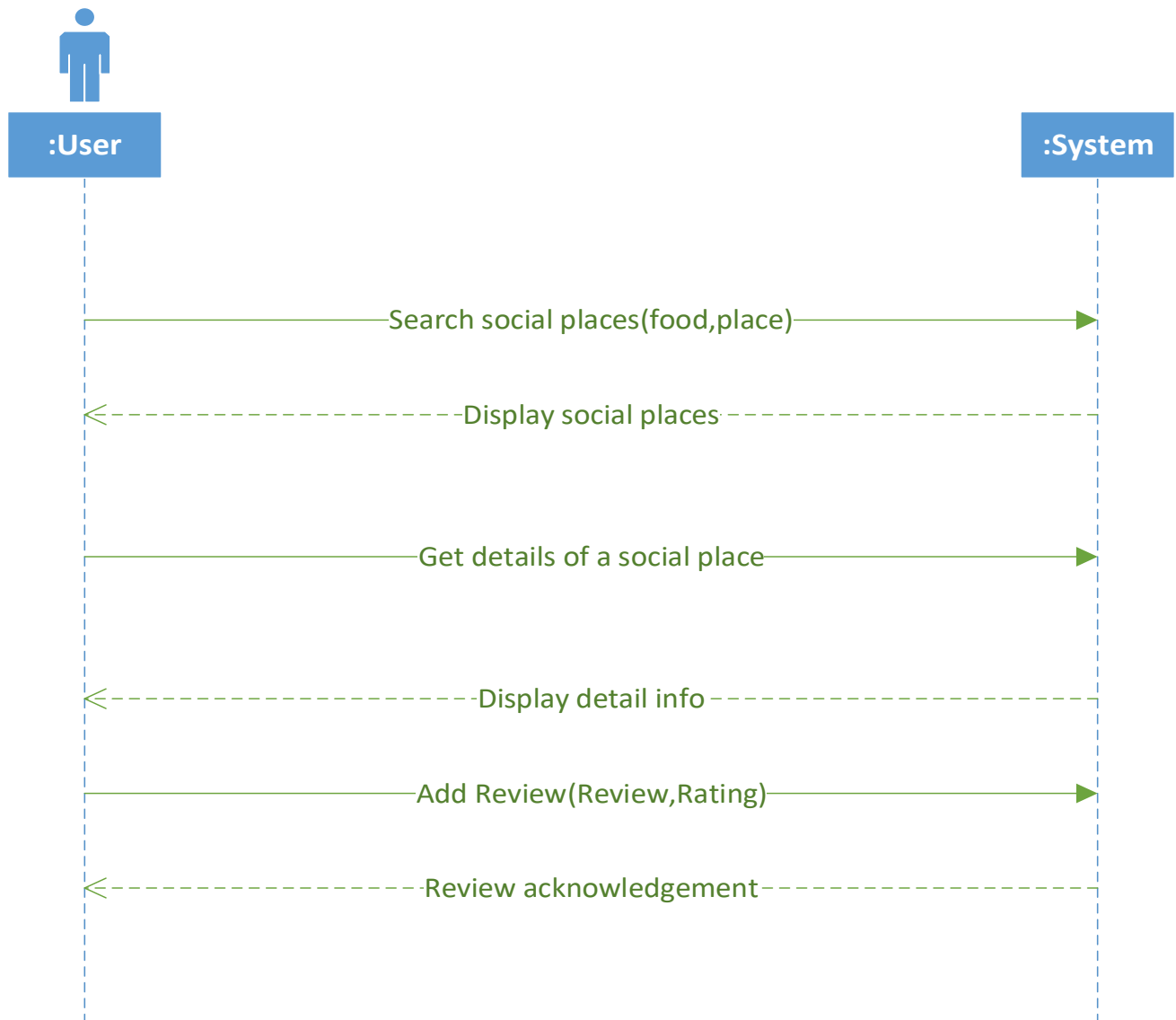
Get Reviews and Get Recommendations
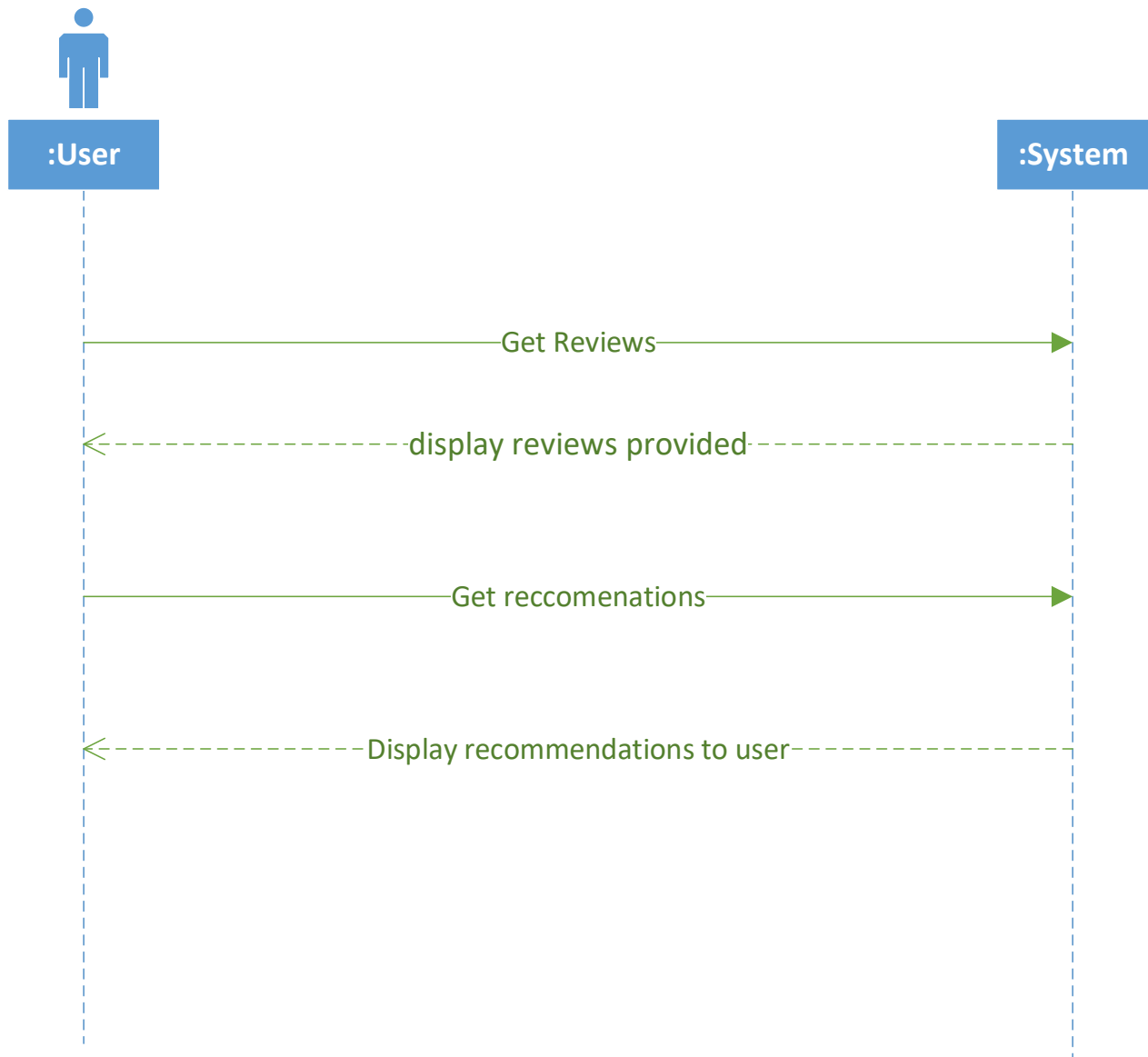
7.ANALYSIS MODEL CLASS DIAGRAM
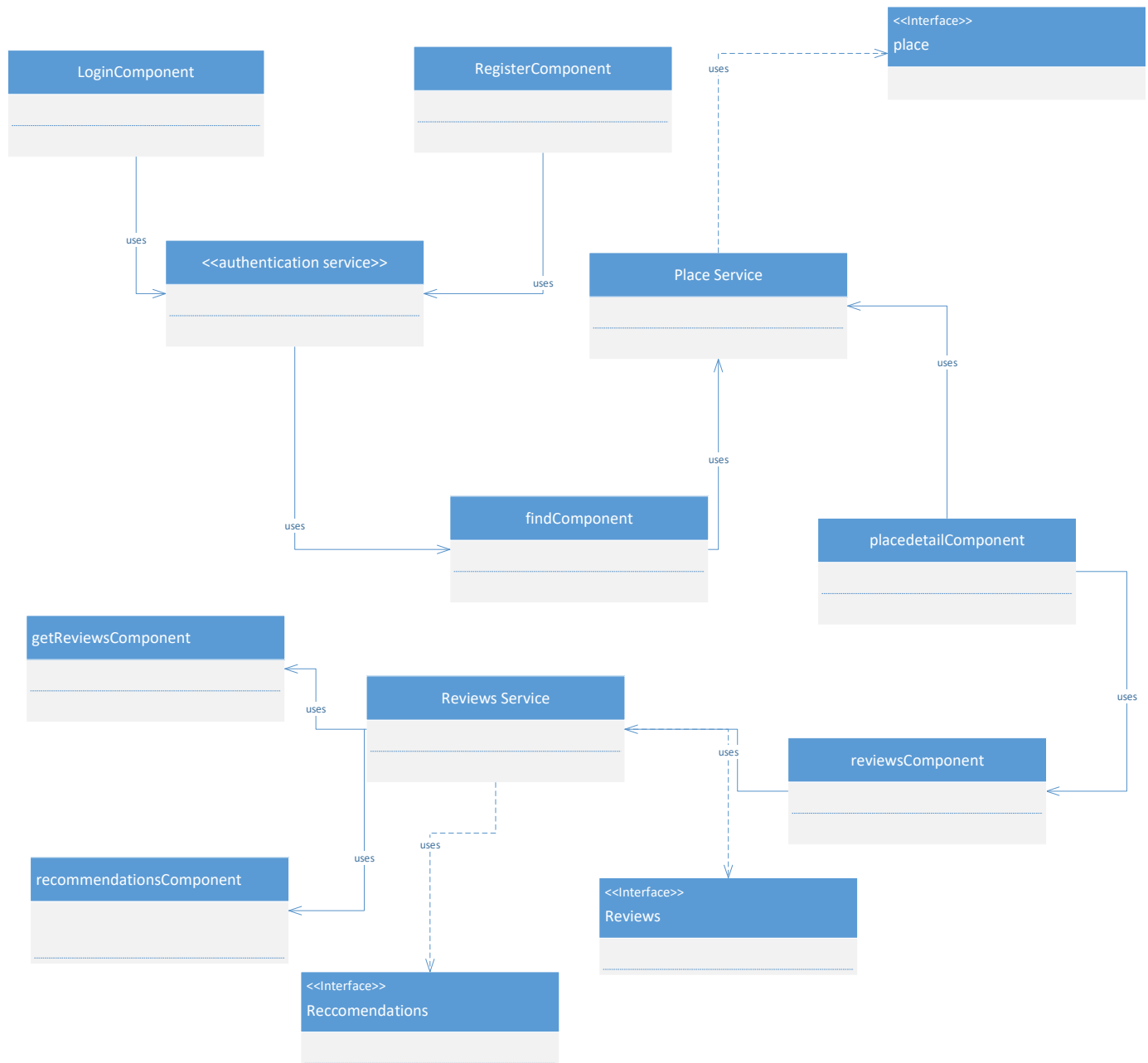
8.DOMAIN MODEL CLASS DIAGRAM

LoginComponent

-userName:String
-userPassword:String

-userLogin()
-userRegister()

RegisterComponent

-userName:String
-userPassword:String
-userConfirmPassword:String
-userFirstName:String
-userLastName:String

-Register()

<<Interface>>
place

- name: String
-display_phone: String
-address1: String
-is_closed: String
-rating: String
-review_count: String
-latitude: Number
-longitude: Number
-id:string

<<authentication service>>

-Validator
-Injectable
-HttpClient
-Router
-Map

-getToken()
+getUserDetails()
+isLoggedin()
+register()
+login()
+profile()
+logout()

Place Service

-injectable
-httpClient
-obseravble

-getPlaces()
-getPlaceselected()
-findPlaces()
-findPlacedetail()

findComponent

-find:String
-where:string

-findPlaces()

placedetailComponent

-placeSelected:String

-getPlacedetail()
-fetchPlacedetails()
-getInformation()

getReviewsComponent

-username:String

getReviews()

Reviews Service

-injectable
-httpHeaders
-httpClient

-getReviews()
-getReccomendations()

reviewsComponent

-reviews:String
-rating:Number

-fetchPlacedetails()
-writeReviews()

recommendationsComponent

-username:String

getReccomendations()

<<Interface>>
Reviews

-reviews: String
-rating: String
- restaurant_name: String

<<Interface>>
Reccomendations

-restaurant_name: String
-user:String

uses

9.DESIGN PATTERNS

Observer Pattern:
- Dropdown Menu for viewing reviews and recommendations contains each values of the respective objects which belongs to the user Logged in.
- This data should be dynamically updated based on the data present in the database for a specific User.
- We use the Reviews Interface of Recommendations Interfaces to receive the and update the menu content.

This works according to the observer pattern (publisher subscriber) where getReviews and getReccomendations would fetch the changes in the data according to the current user logged in to the app.
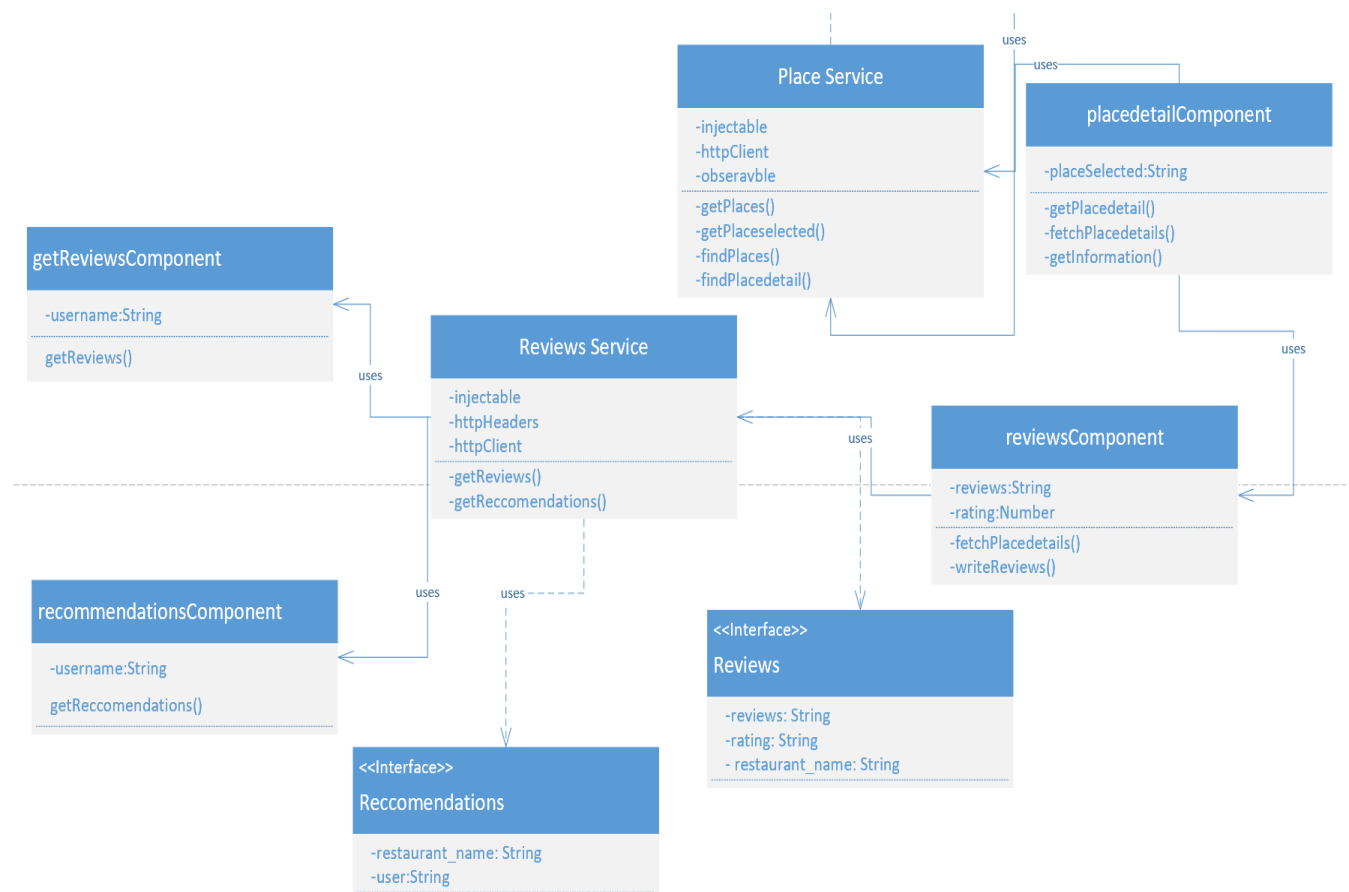
Decorator Pattern:

- Place Details component for viewing the details of the place selected by the user uses the place interface.
- When a place it selected the place object implements the attributes of the place, according to the selected place object and sets the value for Address, Rating etc.
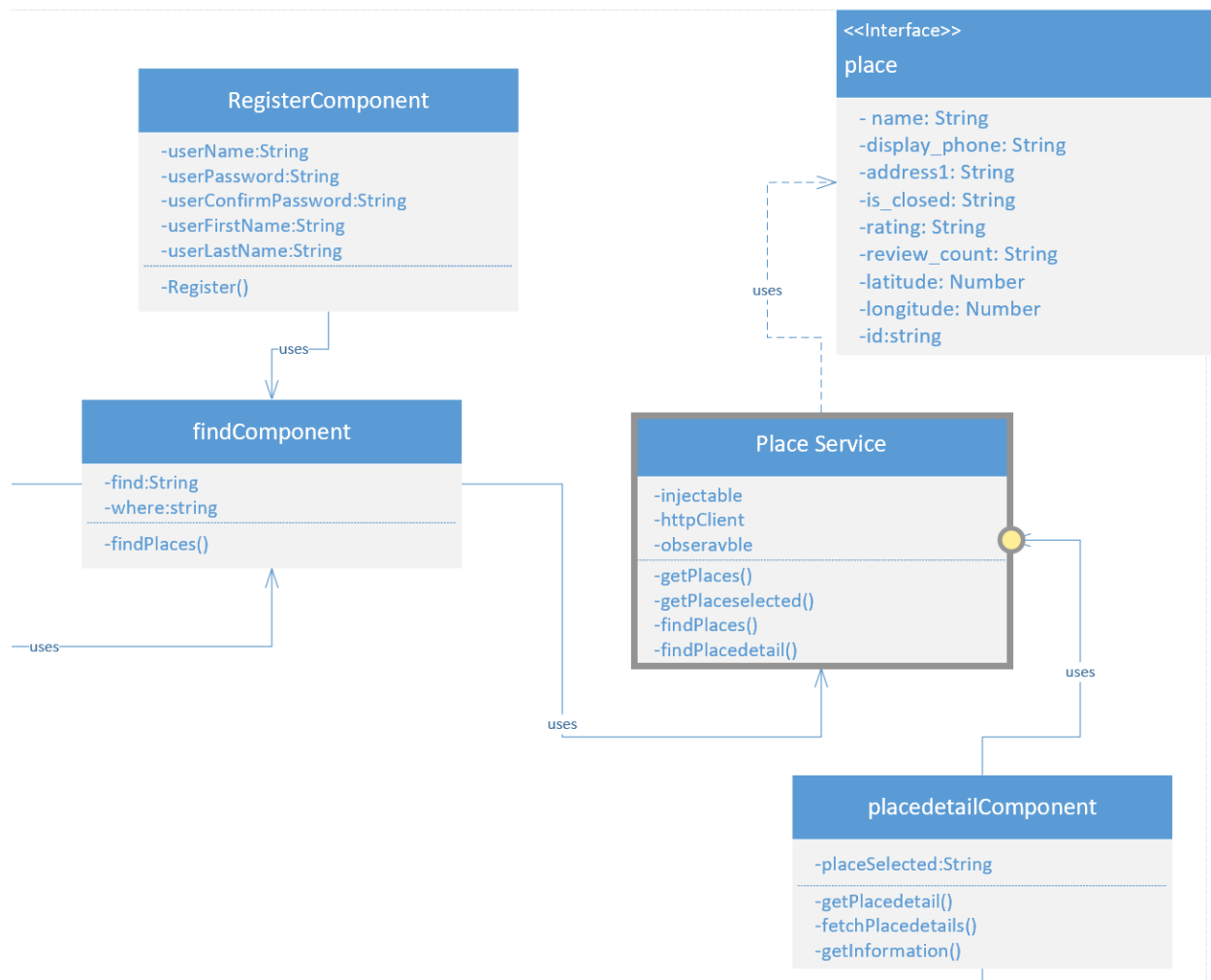
10.APPLICATION SCREENSHOTS

Register Page

Login Page

Application Homepage

Places page, upon searching for a place
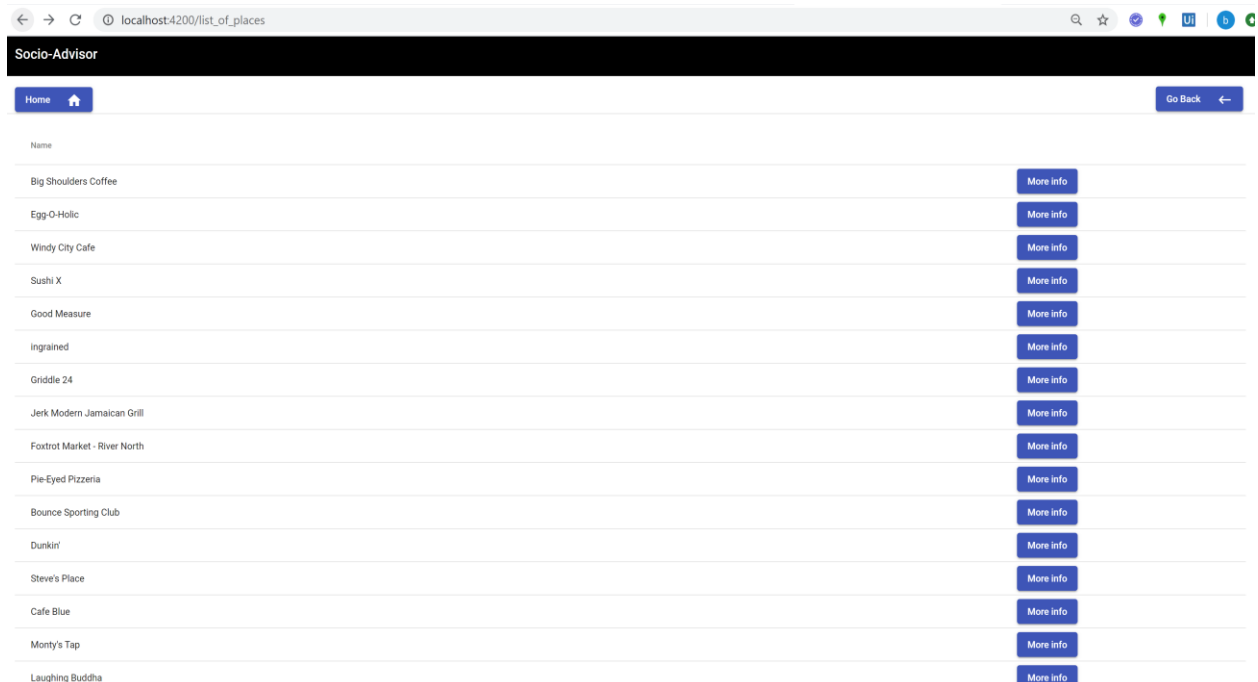
Place Details page

Getting Nutritional Facts

Write Review for a place

Getting User Reviews

Getting User Recommendations