

CSE 343 Software Engineering

Final Report



Team Members:

Hikmet Mete Varol - Project Manager

Bahadır Etka Kılınç - Product Owner

Ömer Kaan Uslu - Developer

Barış Yurdakul - Developer

gitHub : <https://github.com/vegatlas/Vegatlas>

1. User Story and Use Case Scenario

Vegatlas is an restaurant aggregator for vegan and vegetarian community. User can see details about restaurant, reviews about restaurant. User can review and rate the restaurant and add restaurant to favorites. User also is informed about new restaurants on system with notifications.

1.1 User Story

Baily Cargill is a professional football player for Forest Green Rovers football club. Forest Green Rovers is the world's first vegan football club based in England. All club employees and football players have the habit of eating vegan. The club also uses renewable energy sources such as solar energy.

One day Forest Green Rovers football team came to Kadikoy for friendly match with Fenerbahce. Baily's Turkish friend in England, Acun, had told him that the vegan restaurants in Kadikoy were very good. Thinking about this, Baily started to search for a food app. Then he came across the Vegatlas app in the app store. Baily installed the app and signed up for the first time by entering required information such as name, email etc. Then he started browsing restaurants from the map screen. Choosing was difficult for Bailey as there were so many restaurants pinned on the map screen. Then he considered using the filtering feature. He filtered by choosing vegan options and restaurants that had falafel on the menu from the filter menu. So choosing a restaurant became easier for him. When clicking on the restaurants on the map screen, he saw a screen where he could see the details. This screen contained general information about the restaurant and users' comments about the dishes. Then Bailey has added several restaurants to the favorites section using the favorite feature.

Baily invited his friend Luke to dinner and they went to Kadikoy together. Baily looked at the restaurants he added to the favorites section of the Vegatlas app, then chose one and ate dinner with his friend Luke. They ate falafel at the restaurant and they were both very satisfied. Then they added a review to the restaurant detail section in the Vegatlas app of the restaurant where they ate.

They also added rating points to the restaurant. They gave the restaurant 5 rating points because they both liked the food very much. Baily discovered a restaurant based on his eating habits using the Vegatlas app and had a nice dinner with his friend.

1.2 Use Case Scenarios

1- Register/Login into the application

- a) If the user is not registered with the application, he/she registers with his/her name, e-mail etc.
- b) If the user is registered and logged out before, he/she logs into the application with his/her e-mail address and password.

2- Edit account details

- a) User can change e-mail and phone number in Account Screen.
- b) User can change password in Account Screen or forget password button.

3- Get Notification

- a) User can get notification from app and see all notifications in Notification Screen.

4- Browse Map

- a) User can browse map in Map Screen.
- b) User can click restaurant pins on the map and see restaurant details.

5- Search and Filter Restaurant

- a) User can search restaurants that stored in the database
- b) Users can filter restaurants according to their wishes.

6- Restaurant reviews

- a) User can see restaurant reviews and add reviews to any restaurant.
- b) User can give rating point to any restaurant.

7- Favorites feature

- a) User can add any restaurant to favorites section.
- b) User can see favorite restaurants in the Favorites Screen after adding operation.

8- Add/Delete restaurants in system

- a) Administrator can add and delete restaurants in system.
- b) Administrator can see stored restaurants in system.

9- Edit Reviews

- a) Administrator can edit and delete user's restaurant reviews.
- b) Administrator can see any restaurant reviews on administration panel.

10- Ban user

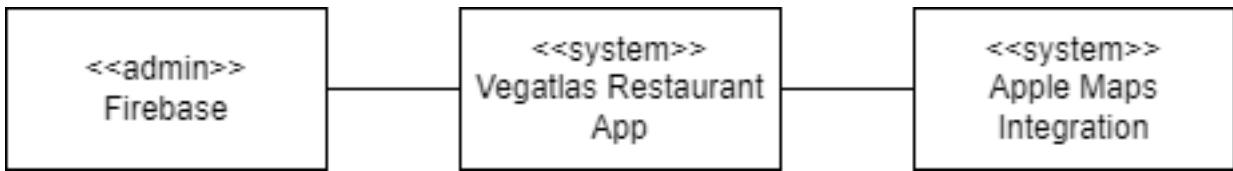
- a) Administrator can ban user permanently or temporarily from system.

2. Project Backlog Documentation

Project Backlog						
ID	As a..	I want to be able to..	So that..	Priority	Sprint	Status
1	User	register and login to system	I can use the benefits of app	Must	2	Done
2	User	change my account details	I can edit details according to my wishes	Should	2	Done
3	User	see my account details	I can see if there is a wrong information	Should	2	Done
4	User	get notification	I can be informed about news	Could	2	Done
5	User	browse map	I can see restaurant at location	Must	3	Done
6	User	see restaurant detail	I can have an opinion about restaurants	Must	3	Done
7	User	add restaurant reviews	I can share my opinion with community	Should	5	Done
8	User	add restaurants to favorites tab	I can easily access my favorite restaurants	Must	4	Done
9	User	send and get message	I can communicate with Vegatlas community	Could	6	Removed
10	User	like and dislike restaurant reviews	I can share my thoughts about reviews	Could	6	Removed
11	Administrator	add and delete restaurant in system	I can update restaurant data	Must	4	Done

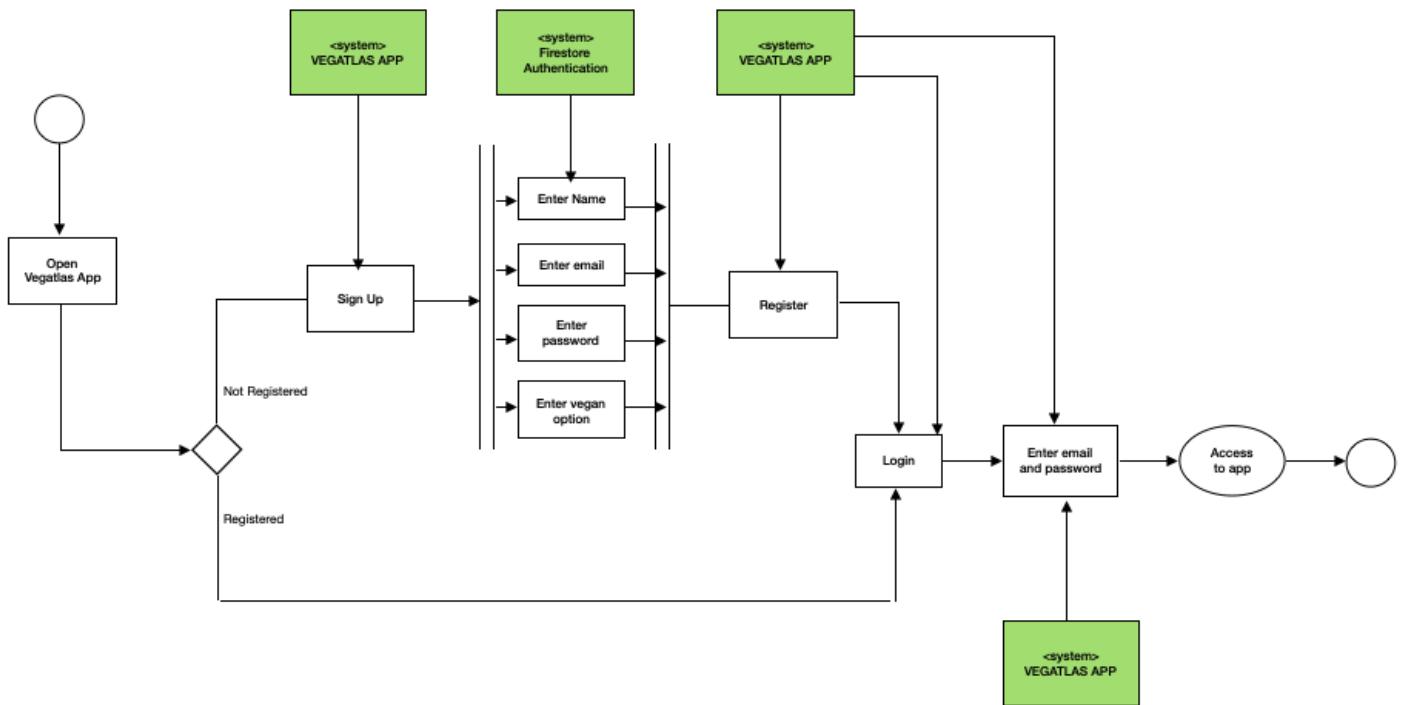
3. UML Diagrams

3.1 Context Model

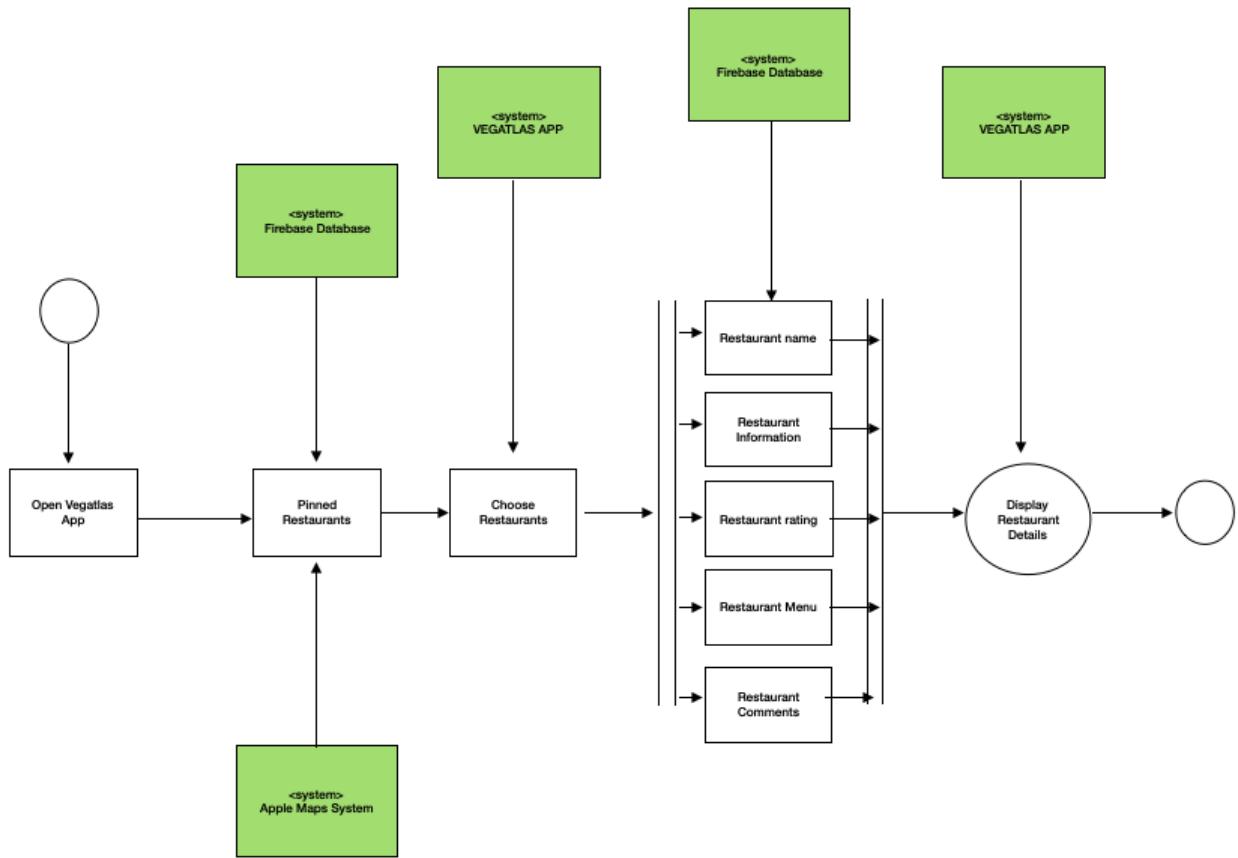


3.2 Process Model

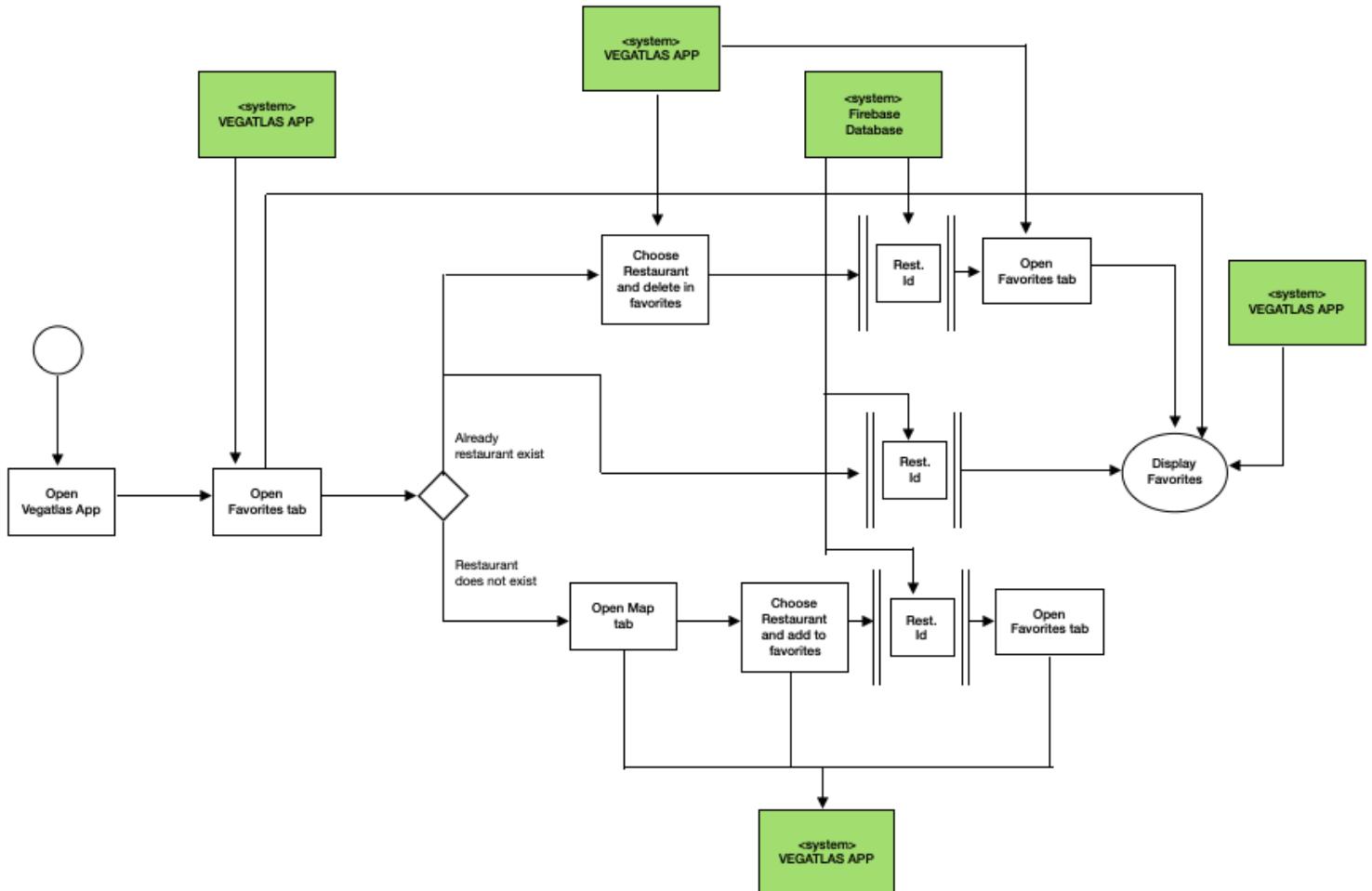
3.2.1 Sign up Process



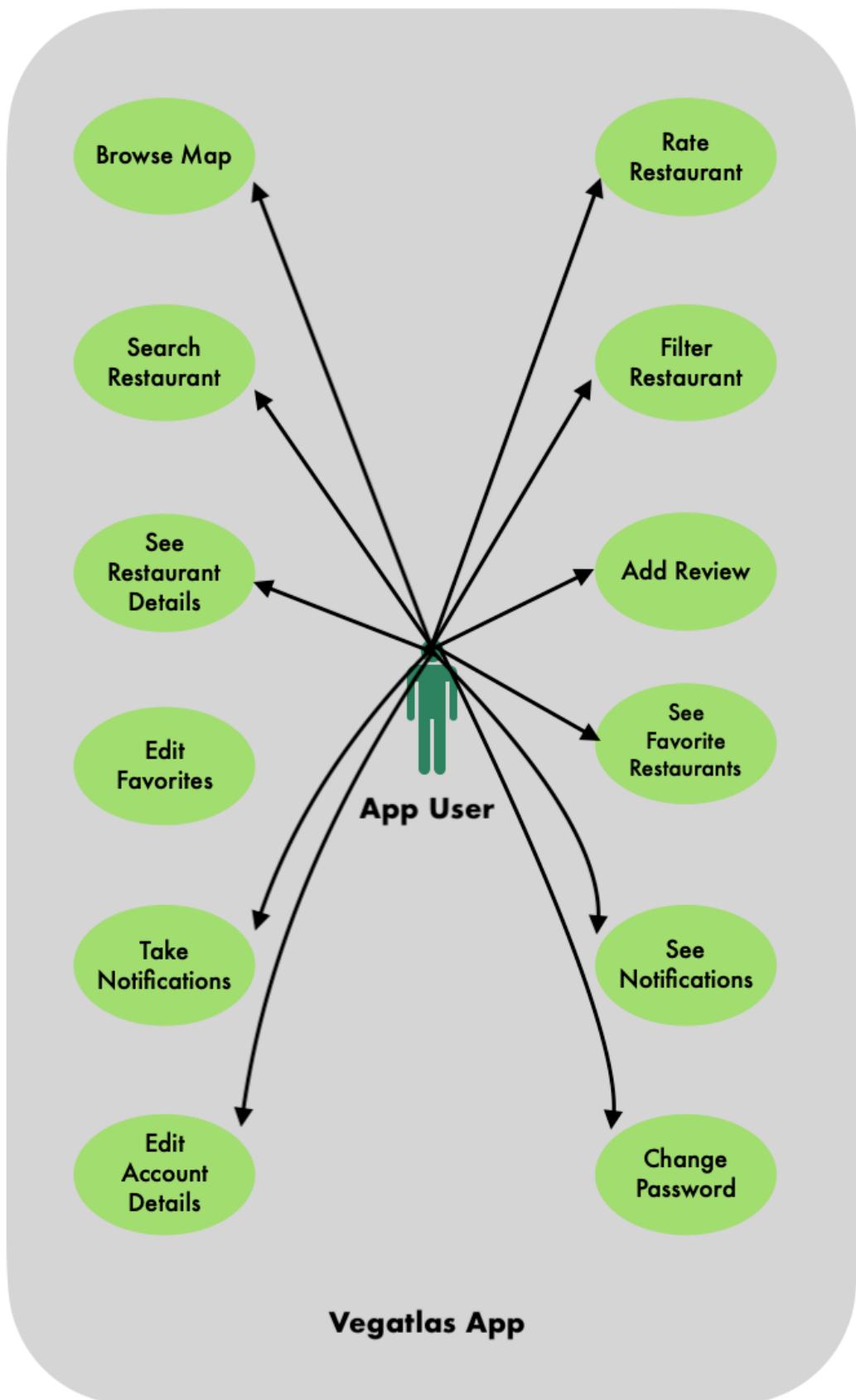
3.2.2 Displaying Restaurant Detail Process

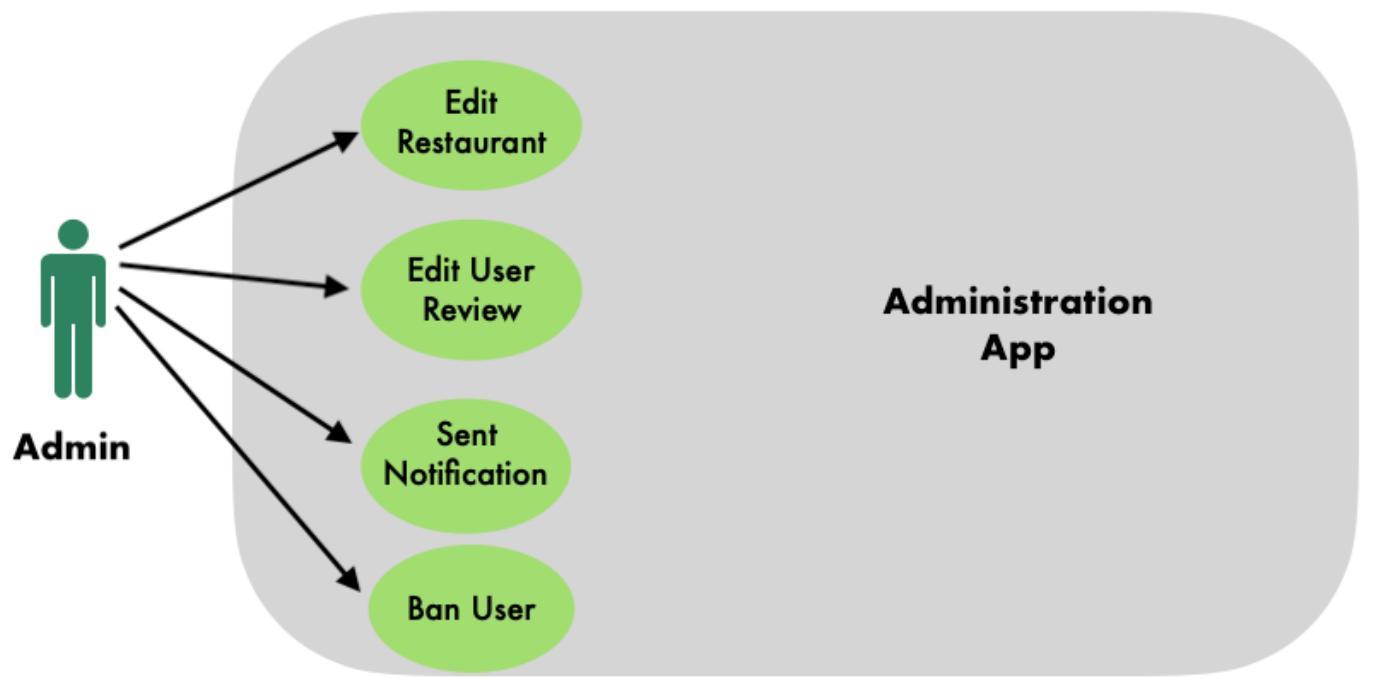


3.2.3 Favorites Screen Process



3.3 Use Case Diagrams & Tables





System	User System
Use case	Filter Restaurant
Actors	User
Data	Restaurant's distance, Restaurant Type, Food Type
Stimulus	There is filter box in main menu. User presses the button and open the filter menu. Selects his choices for search.
Response	Restaurants will be listed according to user's choices. In addition, restaurants will be shown on the map.
Comments	User should filter correctly. If user select the distance 0 for example, there will not be shown any restaurant.

System	User System
Use case	Add review
Actors	User
Data	Comment user wrote
Stimulus	There is comment box in restaurant page. User selects restaurant and in opened page, user can add comment.
Response	Every user's comments will be shown at restaurant page and every user can see that comments.
Comments	If your comment is out of bounds for admin, admin can remove your comment. Every user should enter his comments carefully.

System	User System
Use case	Add Restaurant to Favorites
Actors	User
Data	Restaurant Name, Database
Stimulus	There is your favorites page in main page. User can add favorites by pressing favorite button in every restaurant page.
Response	You can see your favorite restaurants in your favorites page in main page.
Comments	User can remove favorite restaurant whenever he wants.

System	User System
Use case	Change Password
Actors	User
Data	E-mail, password, new password
Stimulus	There is change password button in account details. User can go there and press change password.
Response	<p>App asks from user his e-mail and password. If that data is correct, then user can enter new password.</p> <p>App asks new password twice for authentication.</p> <p>If e-mail or password is not correct, then app asks again.</p>
Comments	User should enter powerful password for safety.

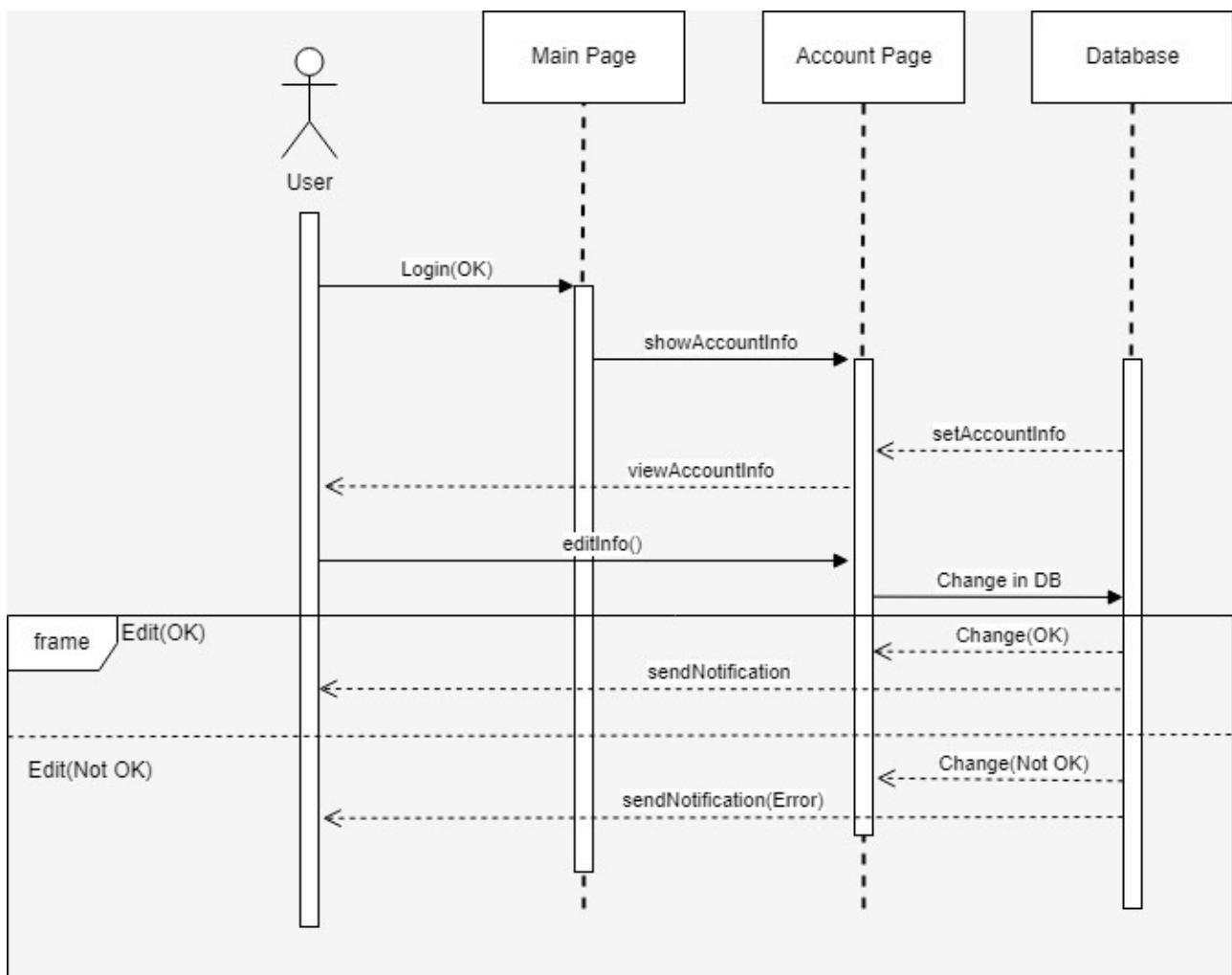
System	Administration System
Use case	Add Restaurant to App
Actors	Admin
Data	Name and location of the Restaurant
Stimulus	Admin wants to add a new restaurant to app. Enters the name of the restaurant to the system and presses add button.
Response	If there is the name of the restaurant info in the system then system gives the error that there is restaurant in here already and asks restaurant name again. If there is no restaurant with that name, new restaurant is added to the system.
Comments	Only admin can add restaurant. Admin must enter restaurant name correctly.

System	Administration System
Use case	Delete Restaurant
Actors	Admin
Data	Name and location of the Restaurant
Stimulus	Admin wants to delete a restaurant from the system. Enters the name of the restaurant to the system and presses delete button.
Response	If there is the name of the restaurant info in the system then deleted from the system along with all information. If there is no restaurant with that name, gives error and asks restaurant name again.
Comments	Only admin can delete restaurant. Admin must enter restaurant name correctly.

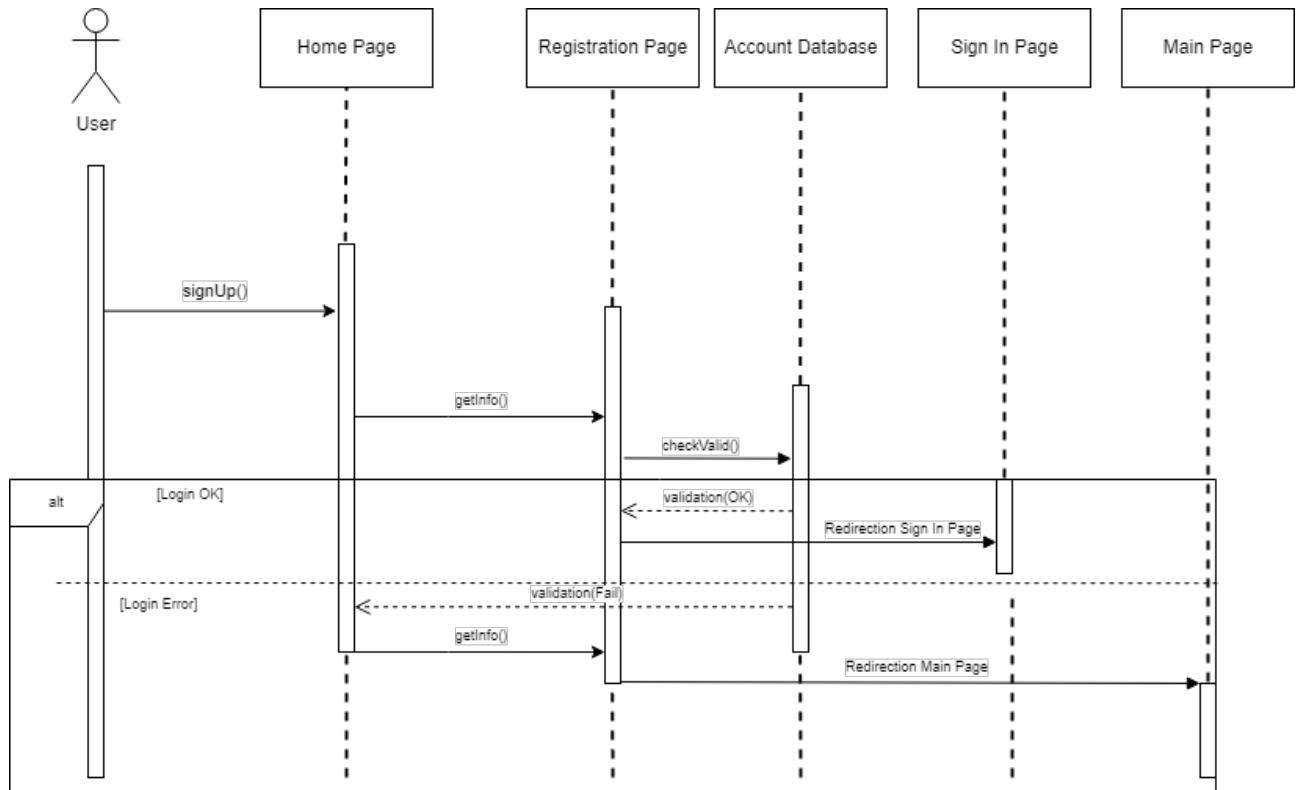
System	Administration System
Use case	Ban User
Actors	Admin
Data	Username of App User
Stimulus	Admin wants to remove app user from the system. Enters username and presses ban button.
Response	If there is username , app user is banned from the system. If there is no app user with that username, any app user cannot register with that username anymore.
Comments	Admin should use it for removing bad users for the app.

3.4 Sequence Diagrams

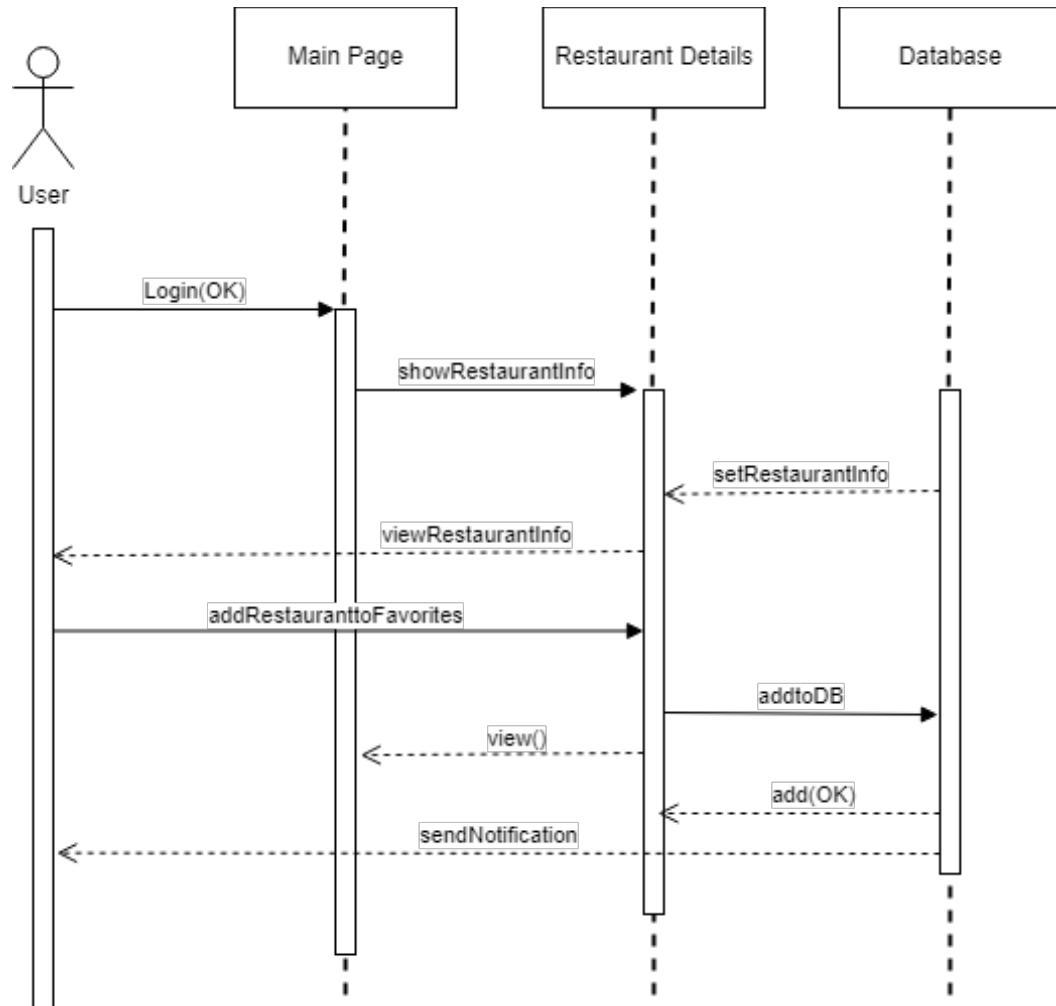
3.4.1 Account Detail Screen Sequence



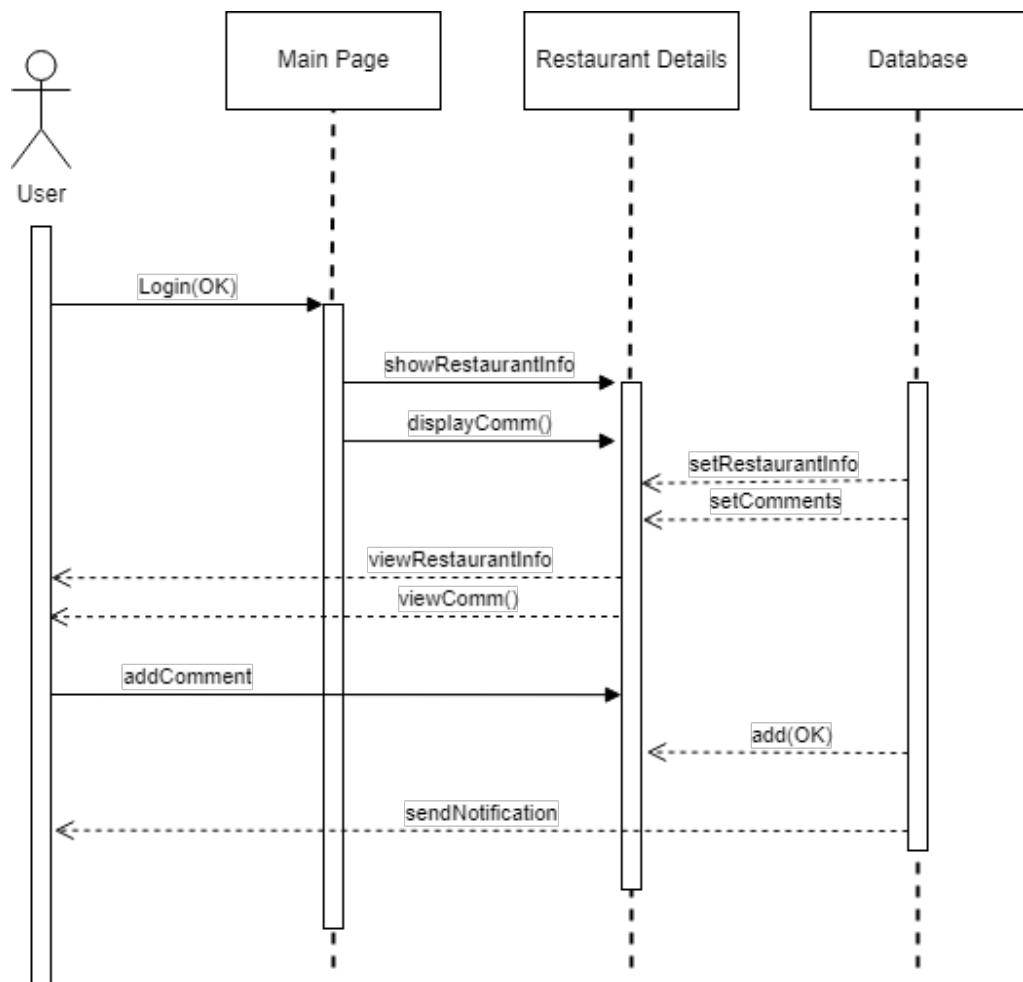
3.4.2 Sign Up Sequence



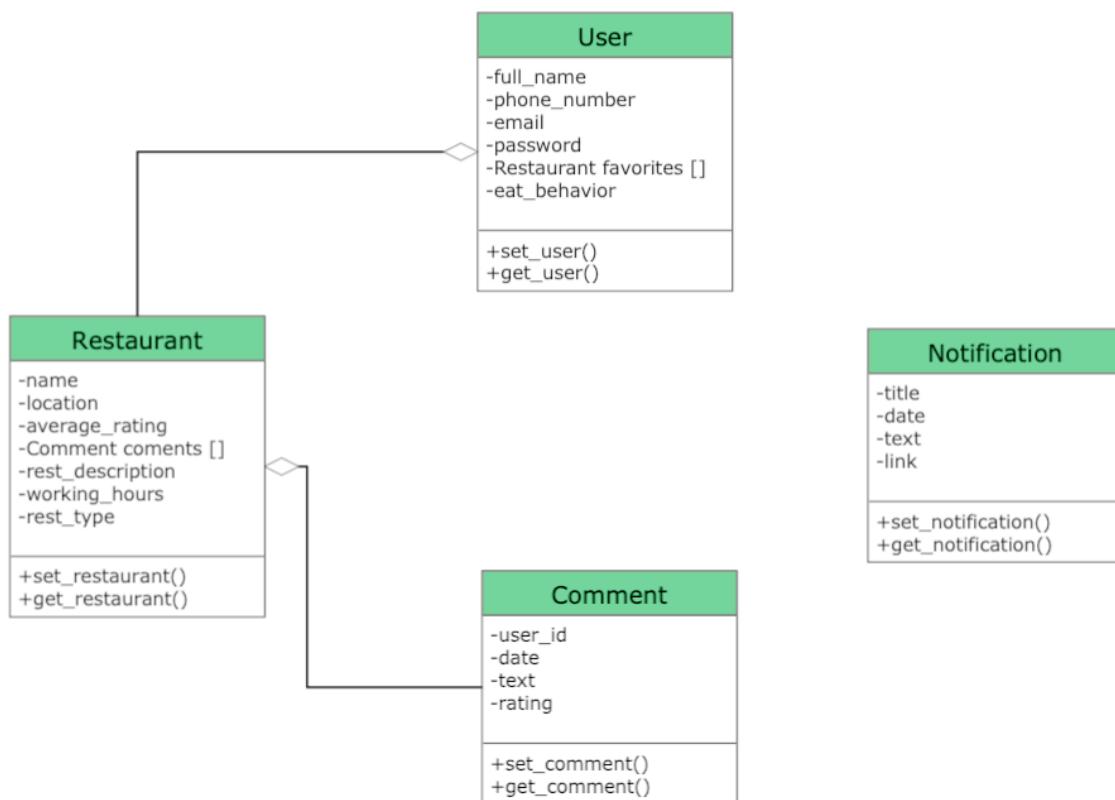
3.4.3 Add Favorite Sequence



3.4.4 Add Review Sequence

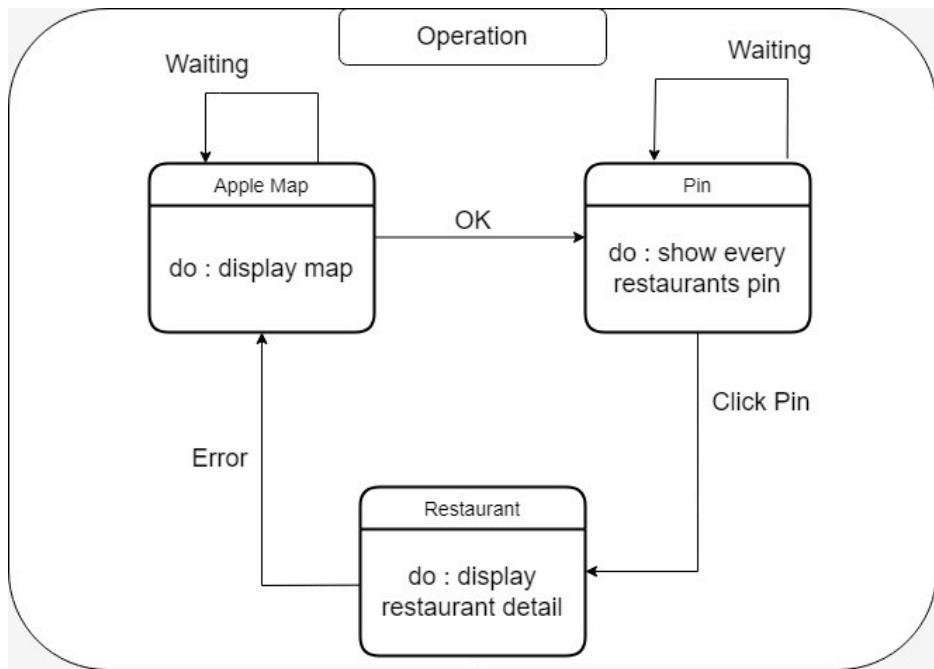


3.5 Class Diagram

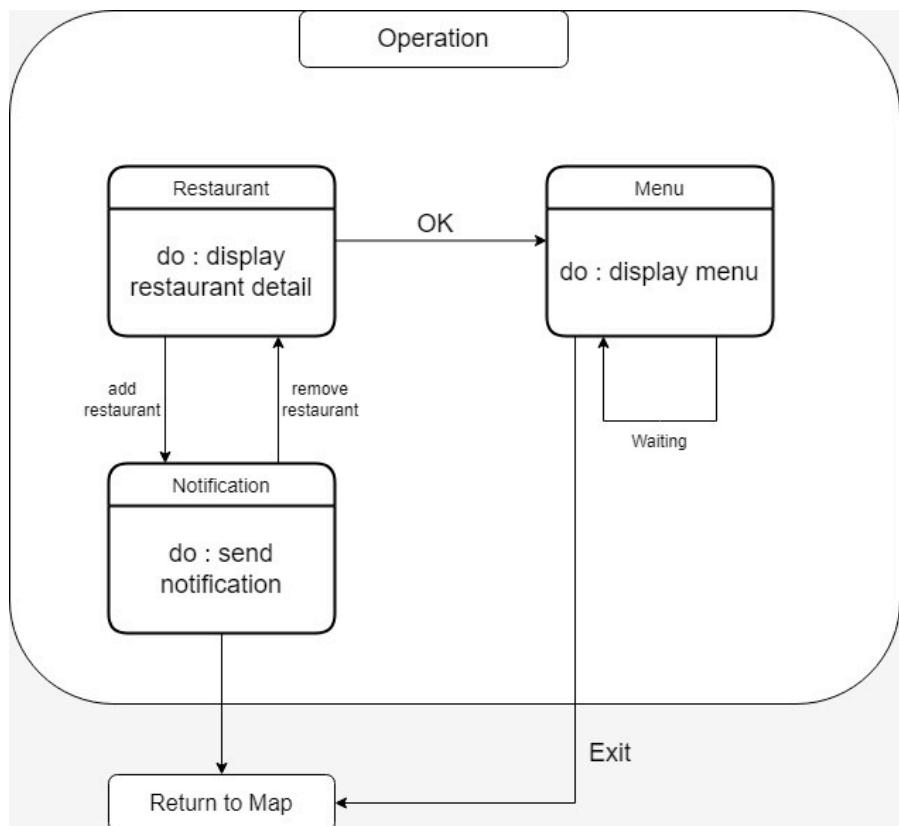


3.6 State Diagrams

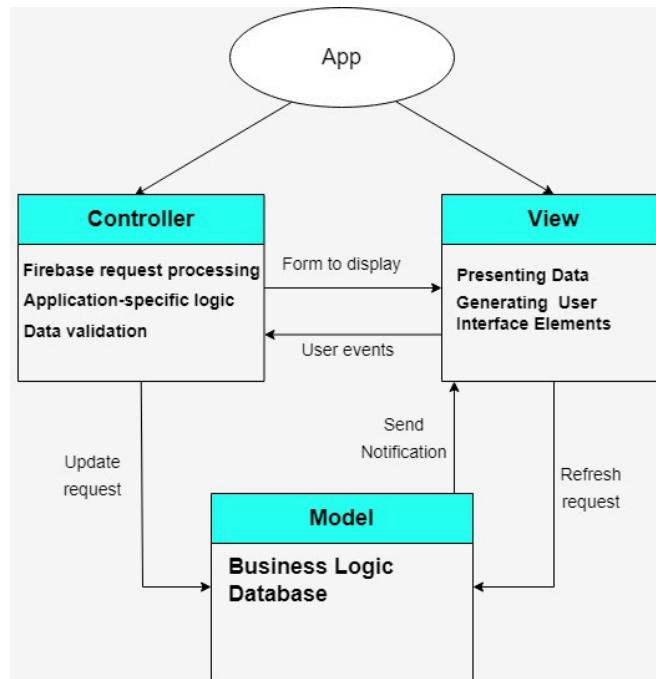
3.6.1 Map Subsystem



3.6.2 Menu Subsystem



3.7 Simple System Architecture



Model View Controller (MVC) Pattern

Model-View-Controller (MVC) is an architectural pattern that separates an application into three main components: the model, the view, and the controller.

- The model is responsible for managing the data of the application. It stores and retrieves data from a database, and provides an interface for the controller to access this data.
- The view is responsible for presenting the data to the user. It defines how the data should be displayed, and generates the necessary user interface elements for the user to interact with the application.
- The controller is the intermediary between the model and the view. It receives input from the user, communicates this input to the model, and updates the view to reflect any changes in the model.

We choose MVC pattern in our project because MVC pattern are that it allows for a separation of concerns, and it makes it easier to test and maintain the application.

Advantages of MVC Pattern

There are several advantages to using the MVC pattern for a Vegatlas mobile app:

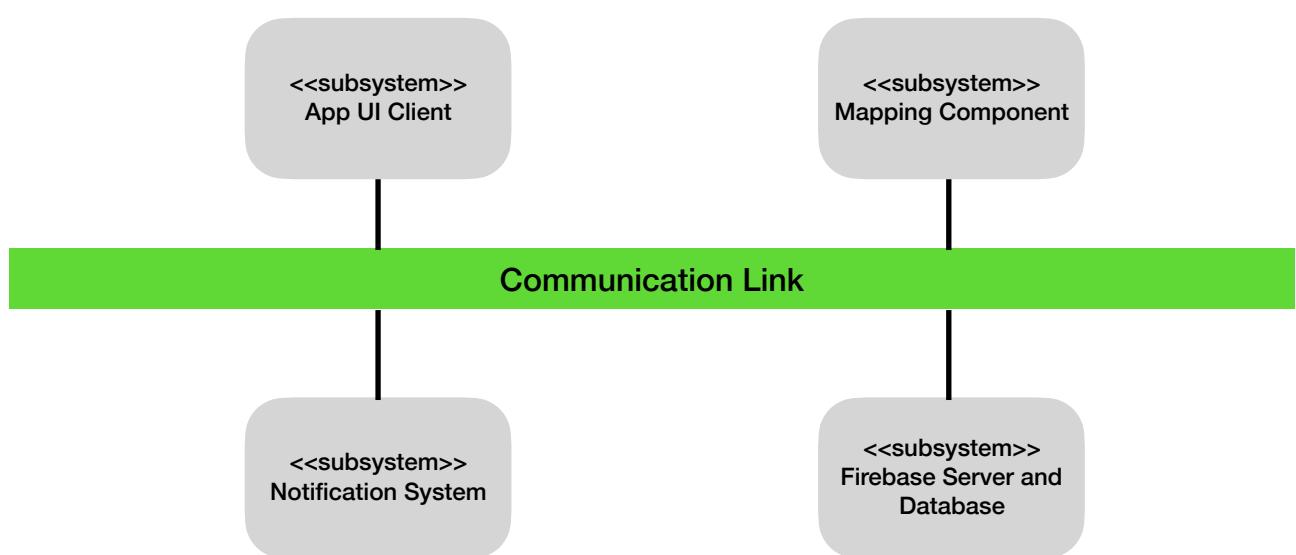
Separation of concerns: MVC allows us to separate the different aspects of our app into distinct components. This means that the model, view, and controller can be developed and tested independently, which makes it easier to maintain and modify the app.

Reusable code: The MVC pattern encourages the use of reusable code, which can save time and effort when developing our app.

Improved testability: MVC makes it easier to test individual components of our app, which can help us catch and fix bugs more quickly.

Better performance: By separating the different aspects of our app into distinct components, we can optimize the performance of each component and improve the overall performance of our app.

Enhanced scalability: MVC makes it easier to scale our app as it grows and evolves, since we can add new features or modify existing ones without affecting the rest of the app.



App UI client: This is the interface that users interact with. The App UI client displays the list of restaurants and allows users to search for specific restaurants or filter the list by various criteria.

Firebase server: This is the server-side component of the app. It is responsible for handling requests from the front-end client, querying the database for restaurant information, and returning the results to the client.

Firebase Database: This is where all of the restaurant data is stored. The back-end server queries the database to retrieve the requested information and returns it to the client.

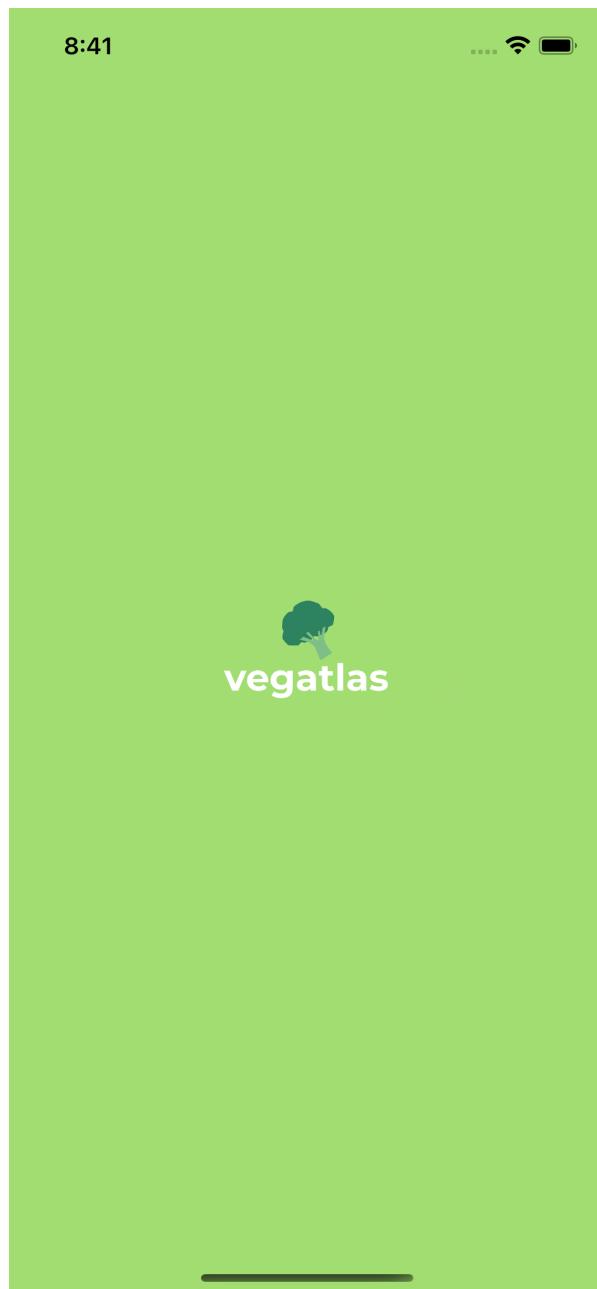
Mapping component: This component is responsible for displaying the location of the restaurants on a map and allowing users to search for restaurants within a specific radius.

Notification system: This component sends notifications to users when new restaurants are added to the app or when their favorite restaurants have special deals or events.

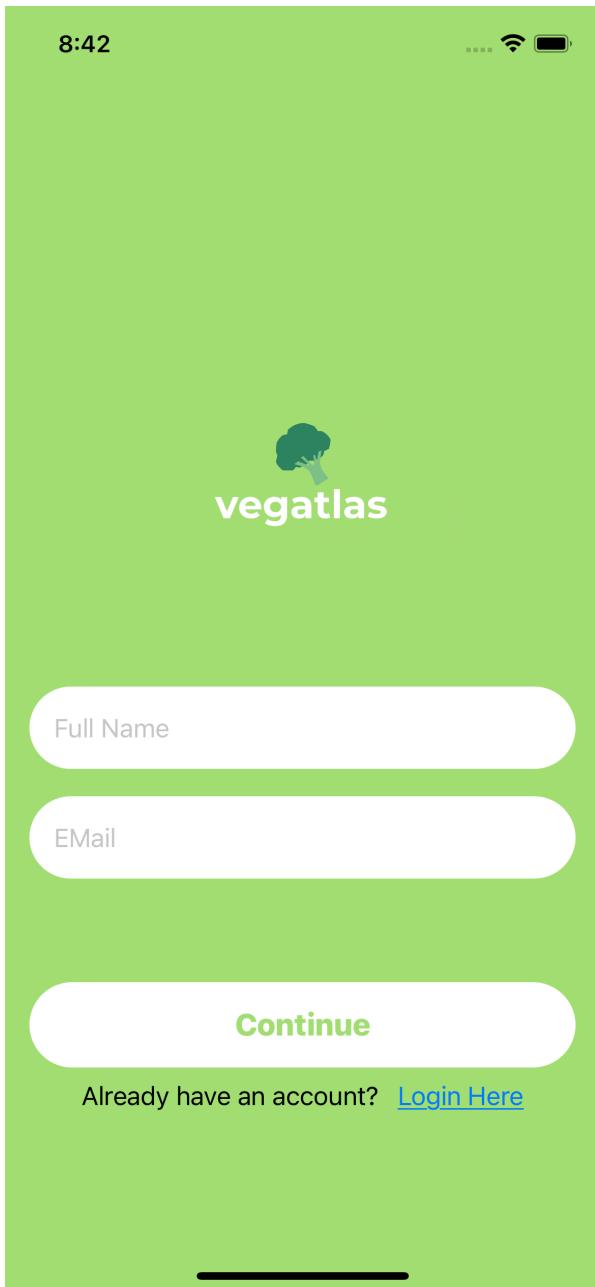
4. Working Graphical Interface with backend services

4.1 Working Graphical Interface

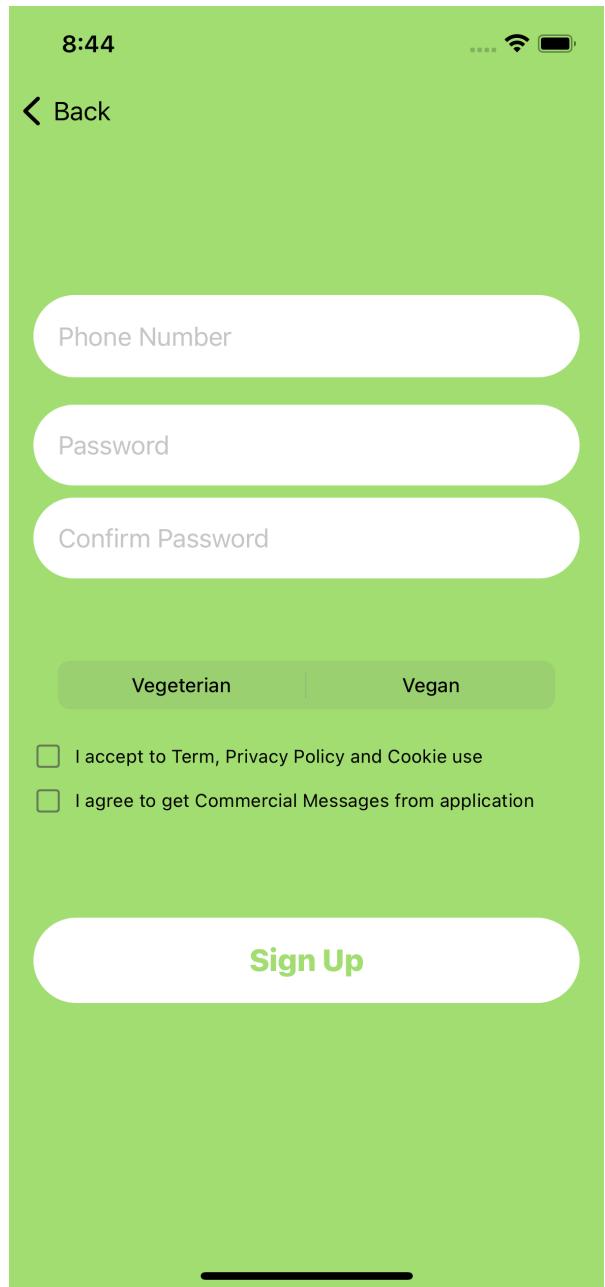
Launch Screen



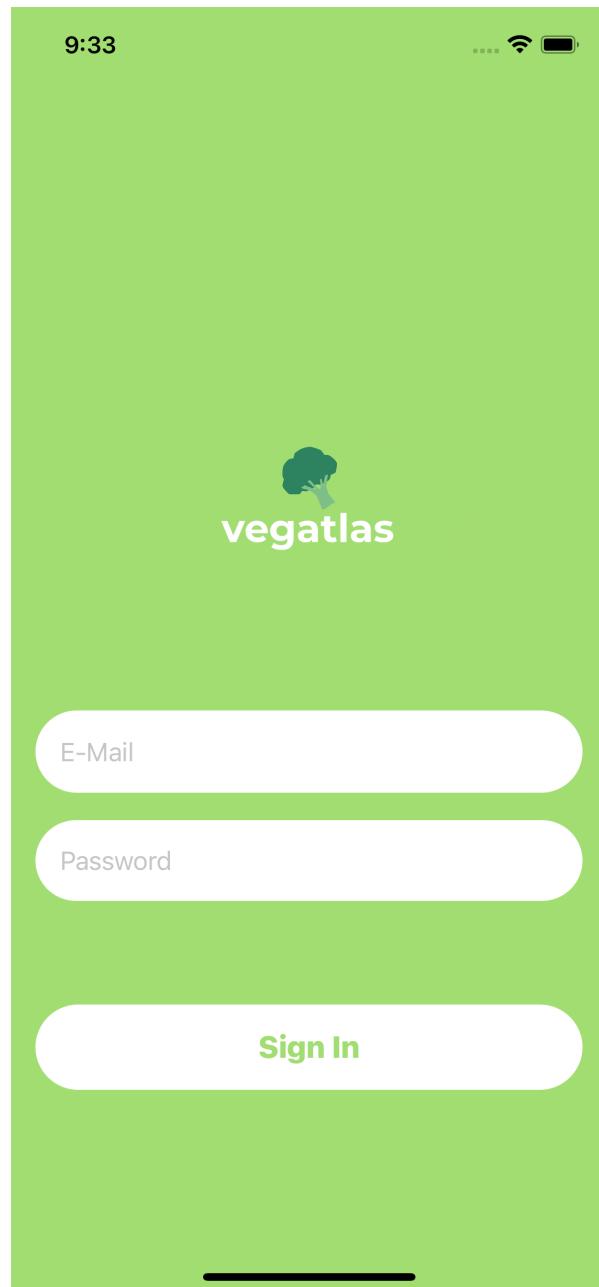
Register Screen 1



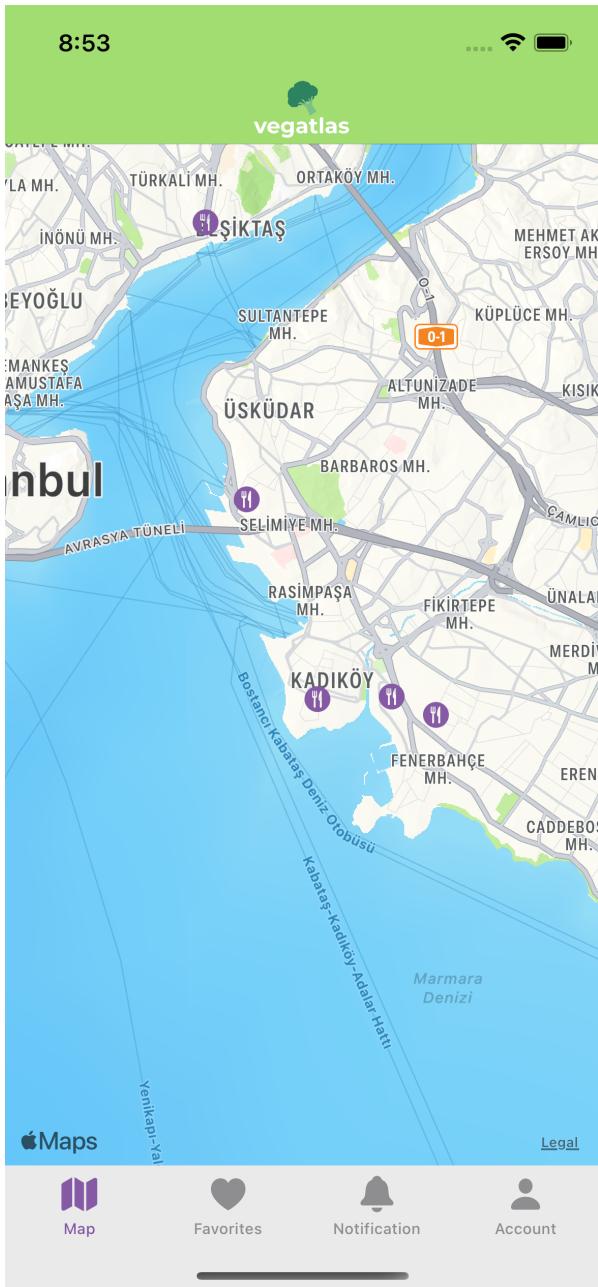
Register Screen 2



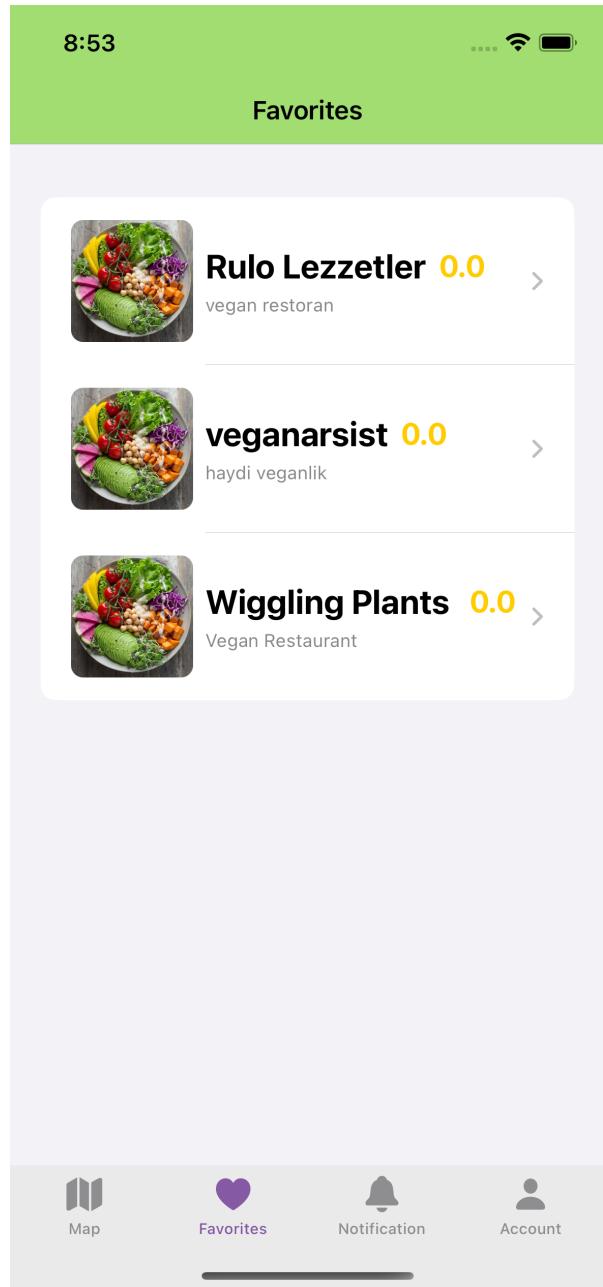
Login Screen



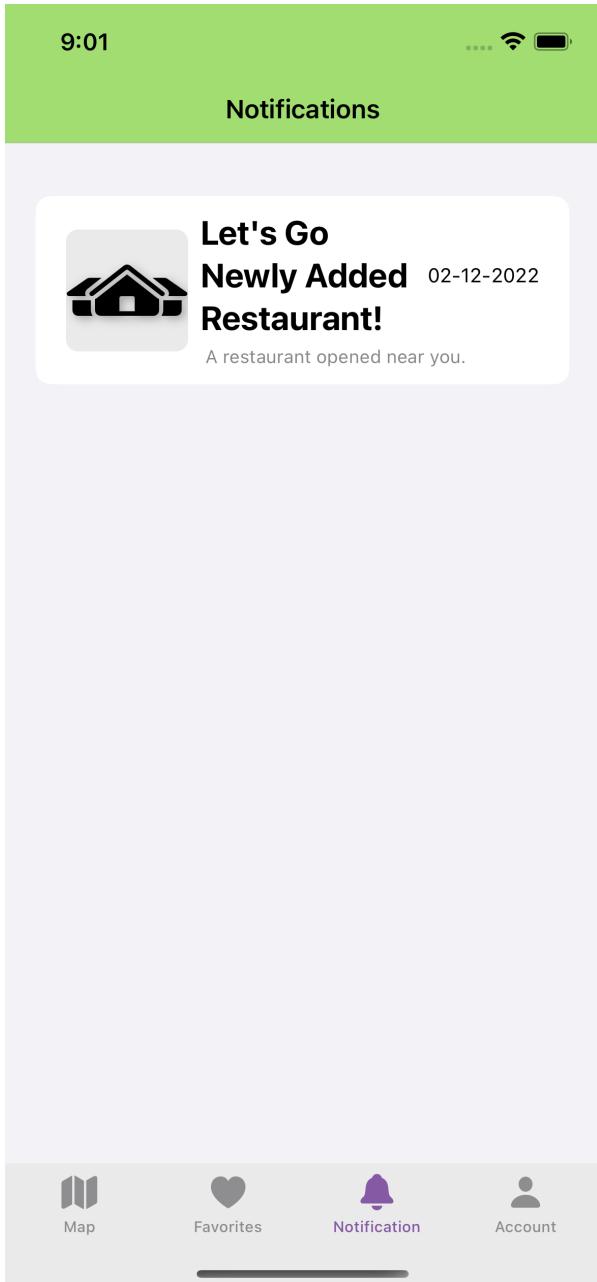
Map Screen



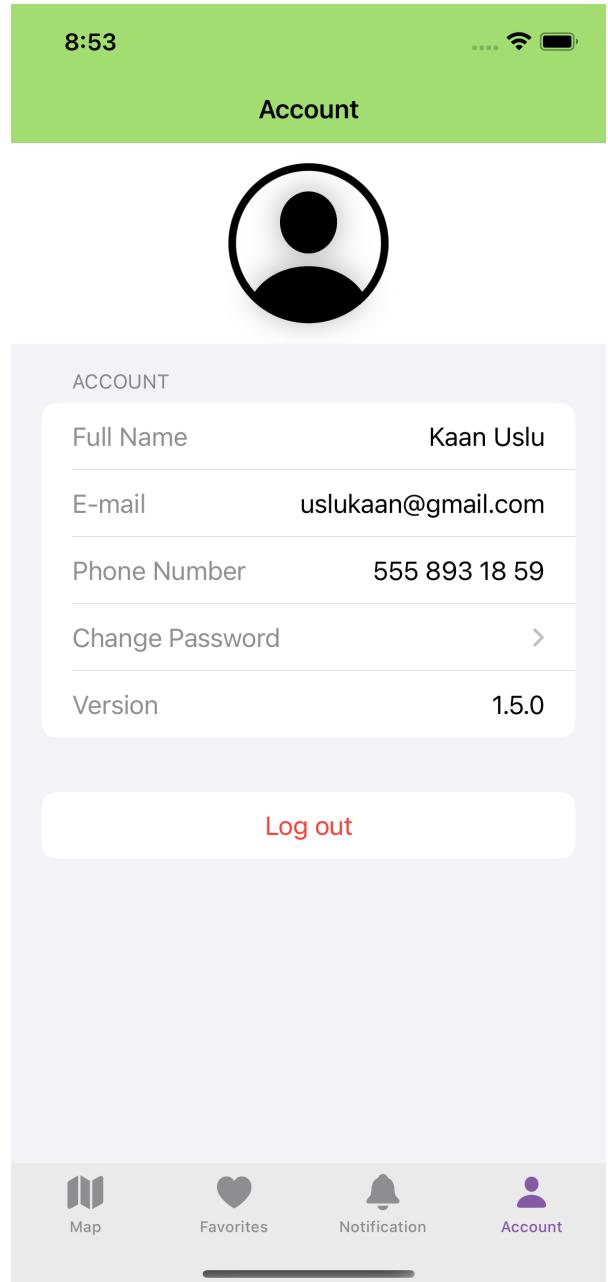
Favorites Screen



Notification Screen



Profile Screen



Restaurant Detail



Wiggling Plants heart icon

This is vegan restaurant

Menu chevron icon

kaan ★★★★★ 02.11.2022 23.43

this is the best restaurant i have ever gone

mete ★★ 02.11.2022 23.43

some weirdo things

Add Comment

baris ★★★★★ 12.11.2022 12.34

Restaurant Detail with Menu



9:17 < Back Wi-Fi icon

Wiggling Plants heart icon

Vegan Restaurant

Menu chevron icon

¶ V'egg Harissa Bap	80₺
¶ Tempeh Blt Bap	70₺
¶ Wild Blueberry Hotcakes	85₺
¶ Toasted Coconut Bread	75₺
¶ Home Fries	45₺
¶ Brunch Bowl	130₺

Map Favorites Notification Account

4.2 Backend Services

User-Authentication Data

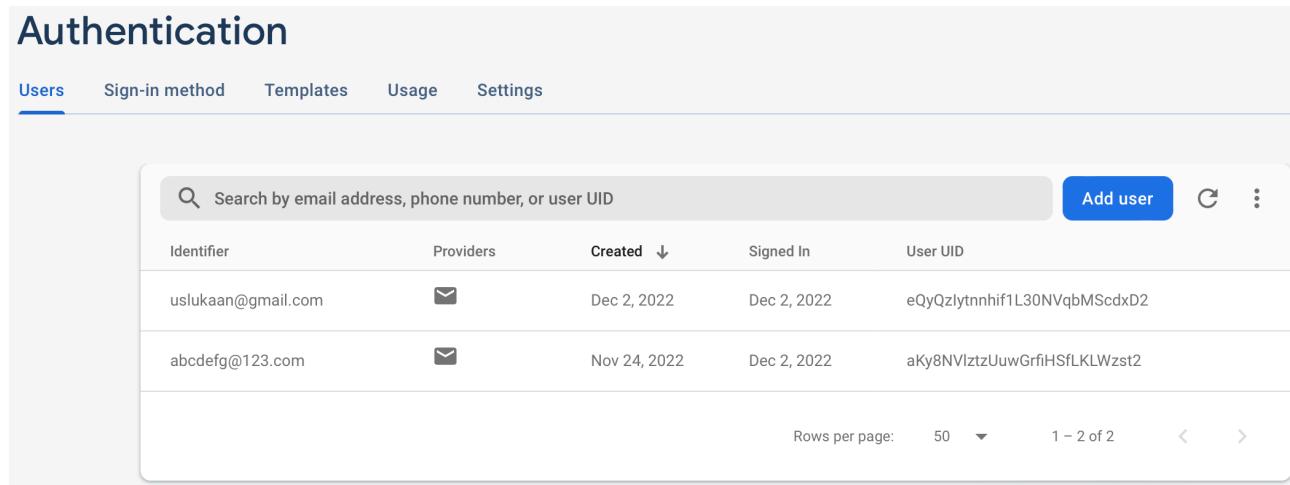
Authentication

Users Sign-in method Templates Usage Settings

Search by email address, phone number, or user UID Add user ⌂ ⋮

Identifier	Providers	Created ↓	Signed In	User UID
uslukaan@gmail.com	✉️	Dec 2, 2022	Dec 2, 2022	eQyQzlytnnhif1L30NVqbMScdxD2
abcdefg@123.com	✉️	Nov 24, 2022	Dec 2, 2022	aKy8NVlztzUuwGrfiHSfLKLWzst2

Rows per page: 50 ▾ 1 – 2 of 2 ⌂ ⌂ ⌂



User Data

vegatas

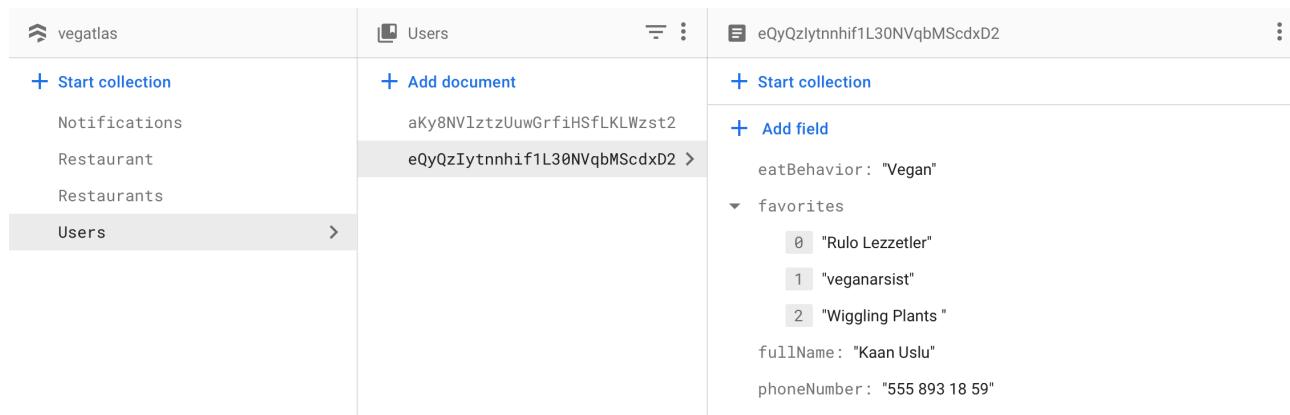
+ Start collection
Notifications
Restaurant
Restaurants
Users >

Users

+ Add document
aKy8NVlztzUuwGrfiHSfLKLWzst2
eQyQzIyttnnhif1L30NVqbMScdxD2 >

eQyQzlytnnhif1L30NVqbMScdxD2

+ Start collection
+ Add field
eatBehavior: "Vegan"
favorites
0 "Rulo Lezzetler"
1 "veganarsist"
2 "Wiggling Plants"
fullName: "Kaan Uslu"
phoneNumber: "555 893 18 59"



Notification Data

vegatas

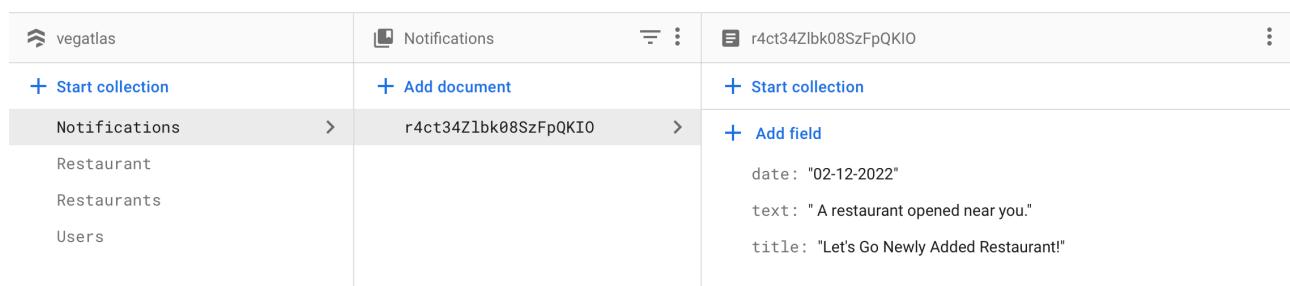
+ Start collection
Notifications
Restaurant
Restaurants
Users >

Notifications

+ Add document
r4ct34Zlbk08SzFpQKIO >

r4ct34Zlbk08SzFpQKIO

+ Start collection
+ Add field
date: "02-12-2022"
text: "A restaurant opened near you."
title: "Let's Go Newly Added Restaurant!"



Restaurant Data

The screenshot shows the Firebase Firestore interface. The left sidebar has collections: vegatlas, Restaurants, and Users. The Restaurants collection is selected. The main area shows the 'Restaurants' collection with documents: Kahve Dunyasi (selected), Kofteeci Yusuf, Rulo Lezzetler, Wiggle Plants, pepe rosso, and veganarsist. The 'Kahve Dunyasi' document details are displayed:

- + Start collection
- + Add document
- + Add field
- menu
- description: "Kahveci"
- lat: 40.9813791661535
- lon: 29.04589320425949
- name: "Kahve Dunyasi"
- restaurantType: "Vegan Kahve"
- workingHours: "09.00 - 23.00"

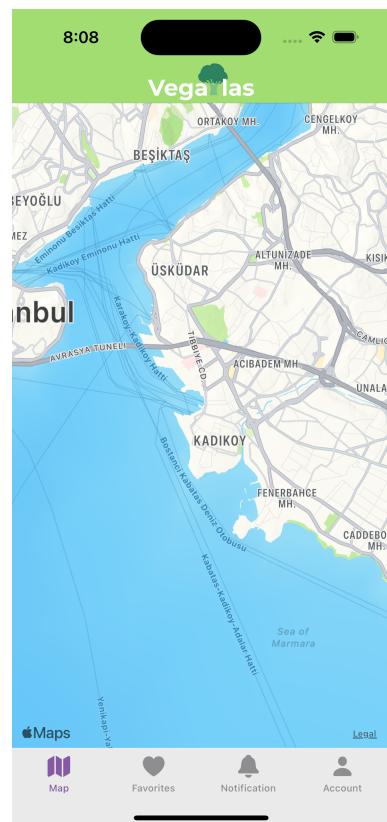
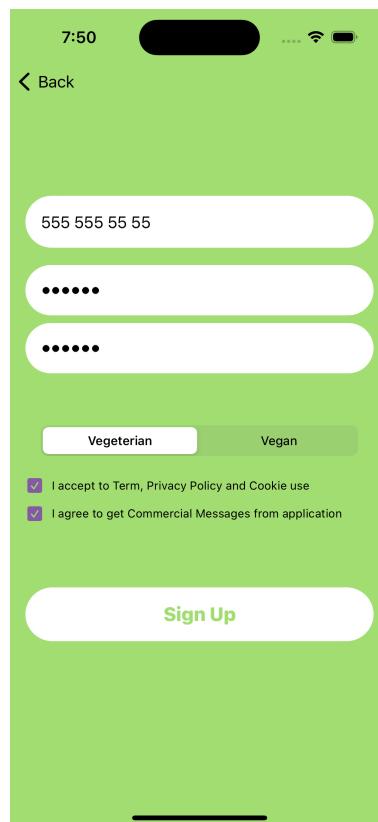
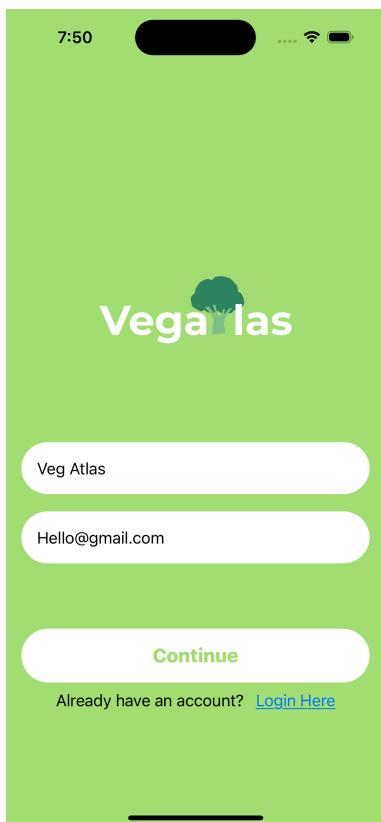
5.TEST CASES

5.1 Test Case (Signing Up Successfully)

Description: An user can sign up successfully.

Test Steps:

1. Enter full name.
2. Enter a valid e-mail address and click continue.
3. Enter valid phone number.
4. Enter valid password.
5. Select vegetarian / vegan option.
6. Accept terms and click sign up.



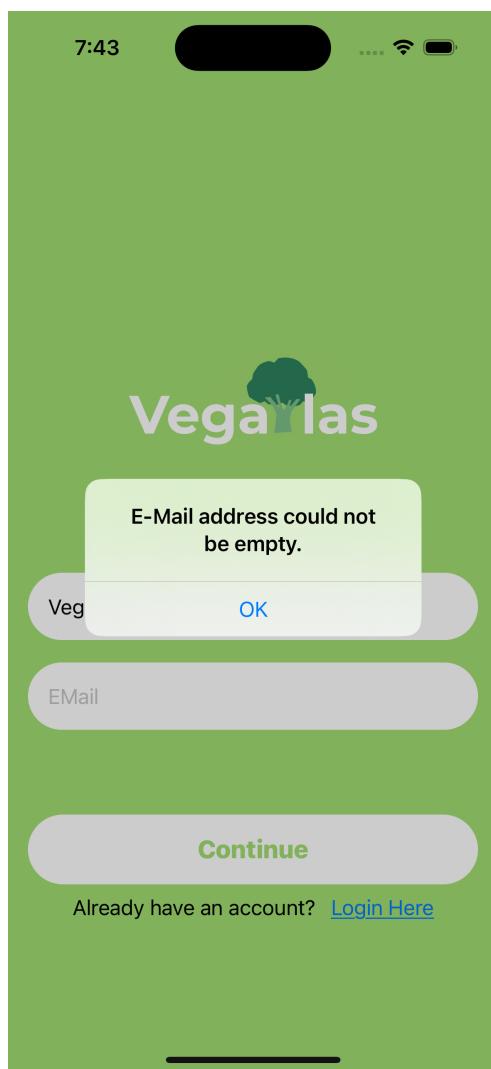
Result: An user signed up successfully.

5.2 Test Case (Signing Up with Empty E-mail)

Description: An user tries to sign up with empty e-mail.

Test Steps:

1. Enter an empty e-mail address.
2. Enter full name.
3. Click continue.



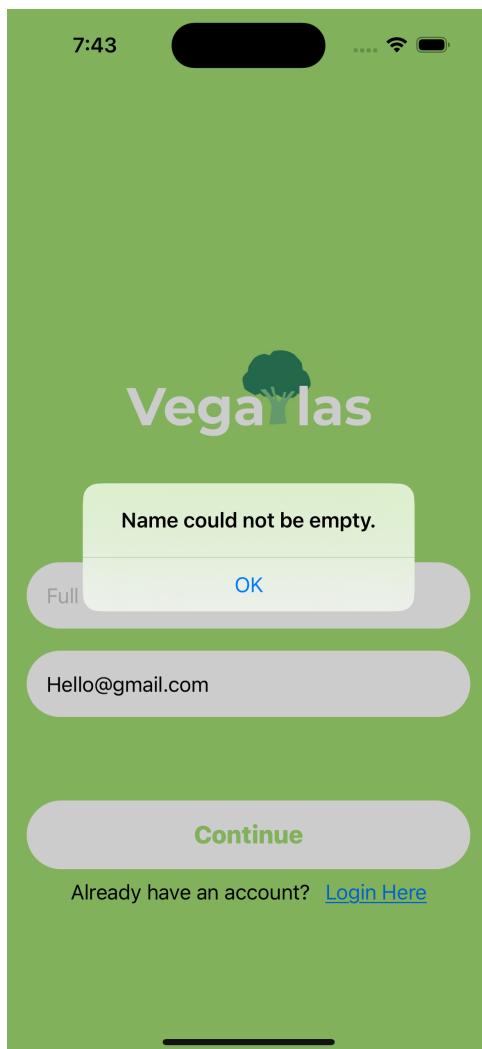
Result: An user can't signed up successfully and sees "E-Mail address could not be empty." alert.

5.3 Test Case (Signing Up with Empty Name)

Description: An user tries to sign up with empty name.

Test Steps:

1. Enter an e-mail address.
2. Enter empty full name.
3. Click Continue.



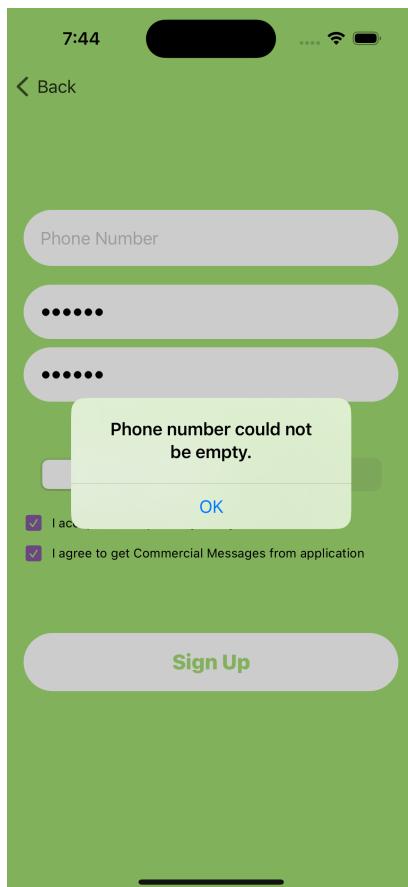
Result: An user can't signed up successfully and sees "Name could not be empty" alert.

5.4 Test Case (Signing Up with Empty Phone Number)

Description: An user tries to sign up with empty phone number.

Test Steps:

1. Enter an e-mail address.
2. Enter full name.
3. Enter empty phone number.
4. Enter passwords.
5. Select eat behavior.
6. Accept terms and click sign up.



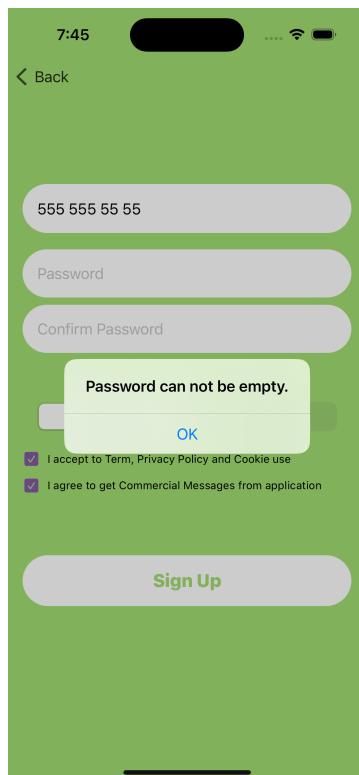
Result: An user can't signed up successfully and sees "Phone number could not be empty." alert.

5.5 Test Case (Signing Up with Empty Password)

Description: An user tries to sign up with empty password.

Test Steps:

1. Enter an e-mail address.
2. Enter full name.
3. Enter phone number.
4. Enter empty password.
5. Enter empty confirm password.
6. Select eat behavior.
7. Accept terms and click sign up.



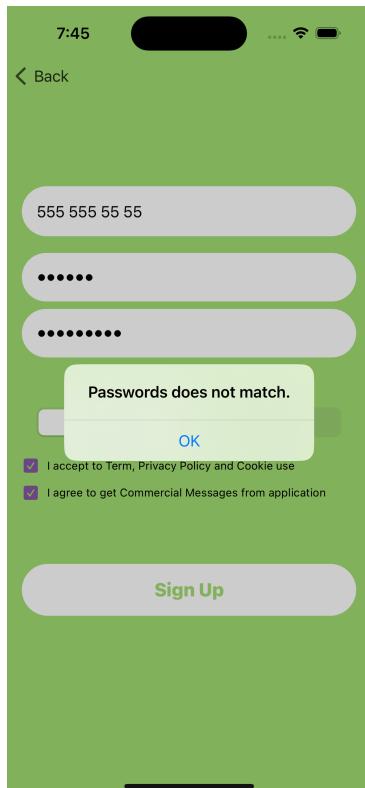
Result: An user can't signed up successfully and sees "Password can not be empty." alert.

5.6 Test Case (Signing Up with unmatched password)

Description: An user tries to sign up with unmatched password.

Test Steps:

1. Enter an e-mail address.
2. Enter full name.
3. Enter phone number.
4. Enter password.
5. Enter confirm password does not match.
6. Select eat behavior.
7. Accept terms and click sign up.



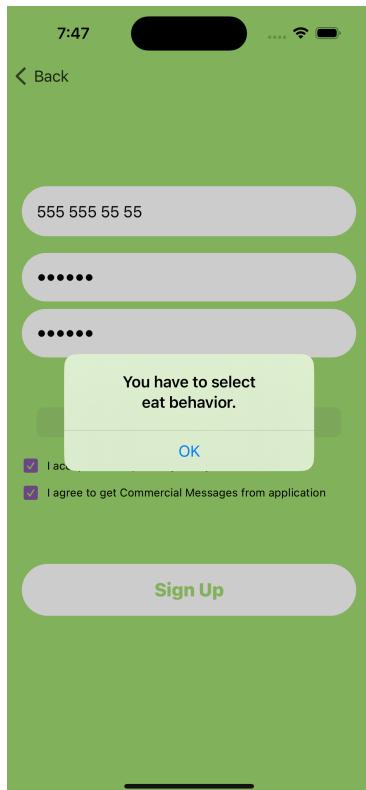
Result: An user can't signed up successfully and sees "Password does not match." alert.

5.7 Test Case (Signing Up with unselected eat behavior)

Description: An user tries to sign up with unselected eat behavior.

Test Steps:

1. Enter an e-mail address.
2. Enter full name.
3. Enter phone number.
4. Enter password.
5. Enter confirm password does not match.
6. Does not select eat behavior.
7. Accept terms and click sign up.



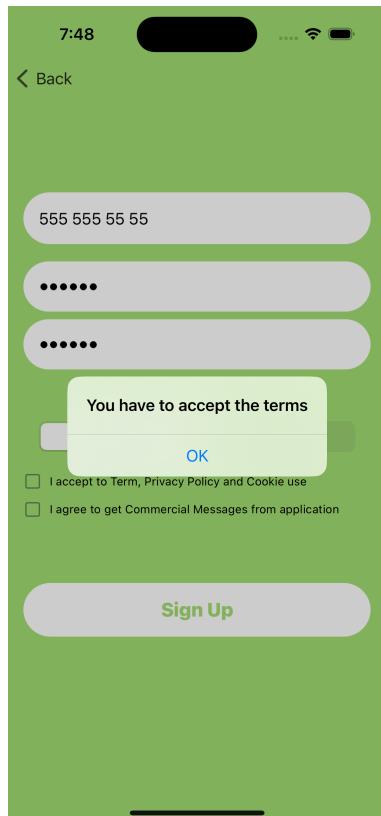
Result: An user can't signed up successfully and sees "You have to select eat behavior." alert.

5.8 Test Case (Signing Up with unselected terms)

Description: An user tries to sign up with unselected terms.

Test Steps:

1. Enter an e-mail address.
2. Enter full name.
3. Enter phone number.
4. Enter password.
5. Enter confirm password does not match.
6. Select eat behavior.
7. Does not accept terms and click sign up.



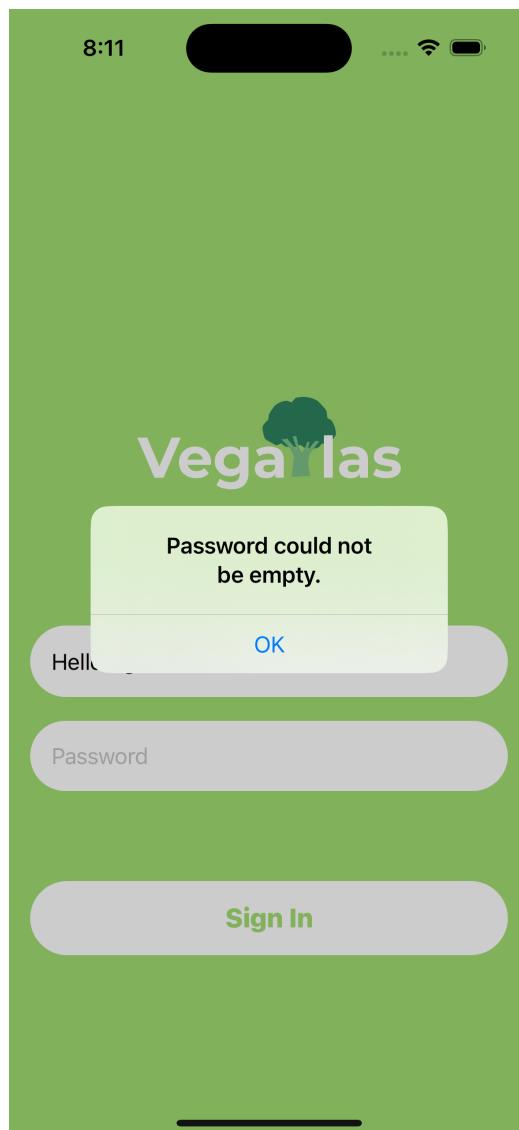
Result: An user can't signed up successfully and sees "You have to accept the terms." alert.

5.9 Test Case (Logging up with empty E-mail)

Description: An user tries to login up with empty E-mail.

Test Steps:

1. Enter an empty e-mail address.
2. Enter password.
3. Click sign in.



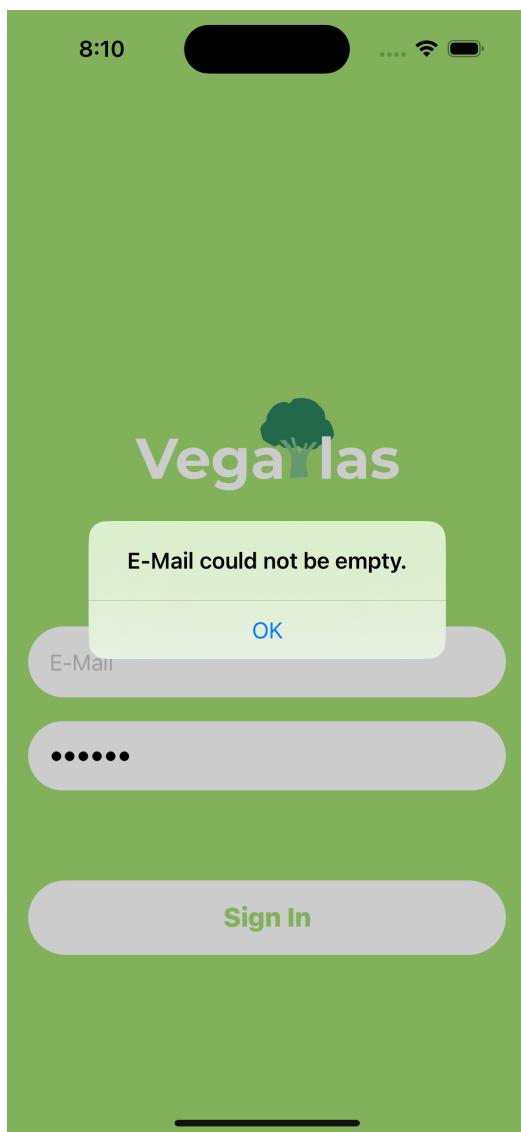
Result: An user can't signed in successfully and sees "Password could not be empty" alert.

5.10 Test Case (Logging up with empty password)

Description: An user tries to login up with empty password.

Test Steps:

1. Enter an e-mail address.
2. Enter empty password.
3. Click sign in.



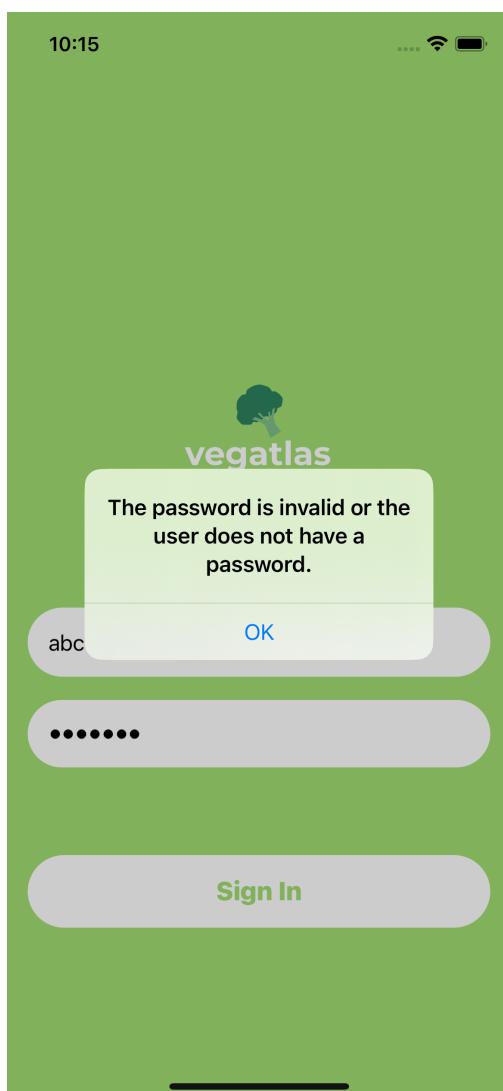
Result: An user can't signed in successfully and sees "E-Mail could not be empty" alert.

5.11 Test Case (Logging up with wrong password)

Description: An user tries to login up with wrong password.

Test Steps:

1. Enter an e-mail address.
2. Enter wrong password.
3. Click sign in.



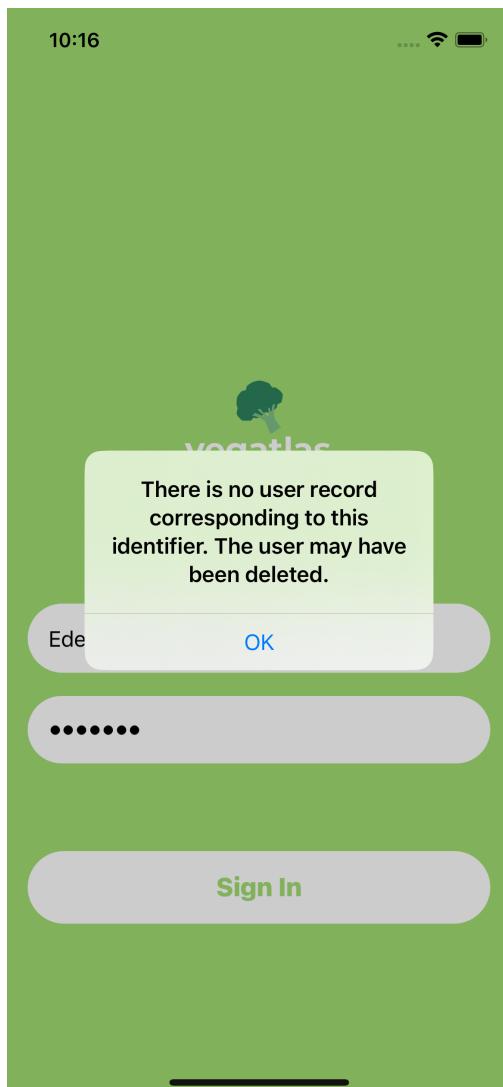
Result: An user can't signed in successfully and sees alert.

5.12 Test Case (Logging up with wrong email)

Description: An user tries to login up with wrong email.

Test Steps:

1. Enter an wrong e-mail address.
2. Enter password.
3. Click sign in.



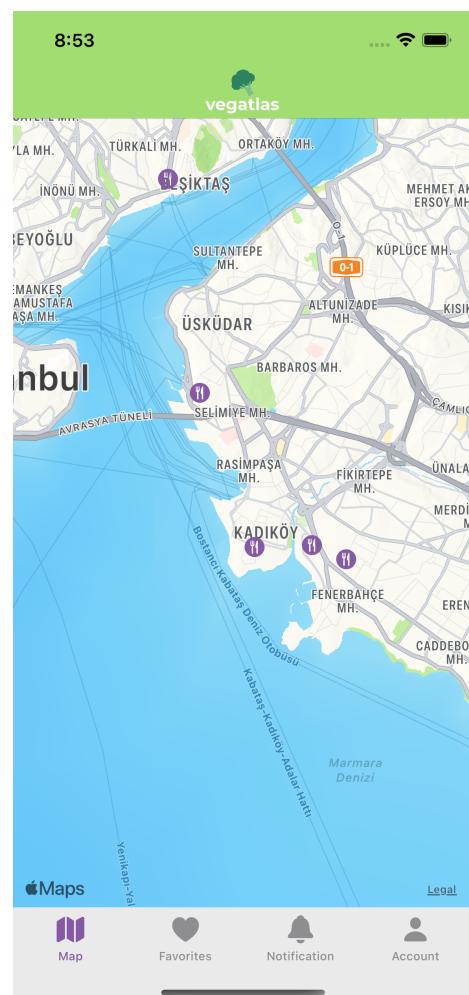
Result: An user can't signed in successfully and sees alert.

5.13 Test Case (Logging up successfully)

Description: An user tries to login up successfully.

Test Steps:

1. Enter an e-mail address.
2. Enter a password.
3. Click sign in.



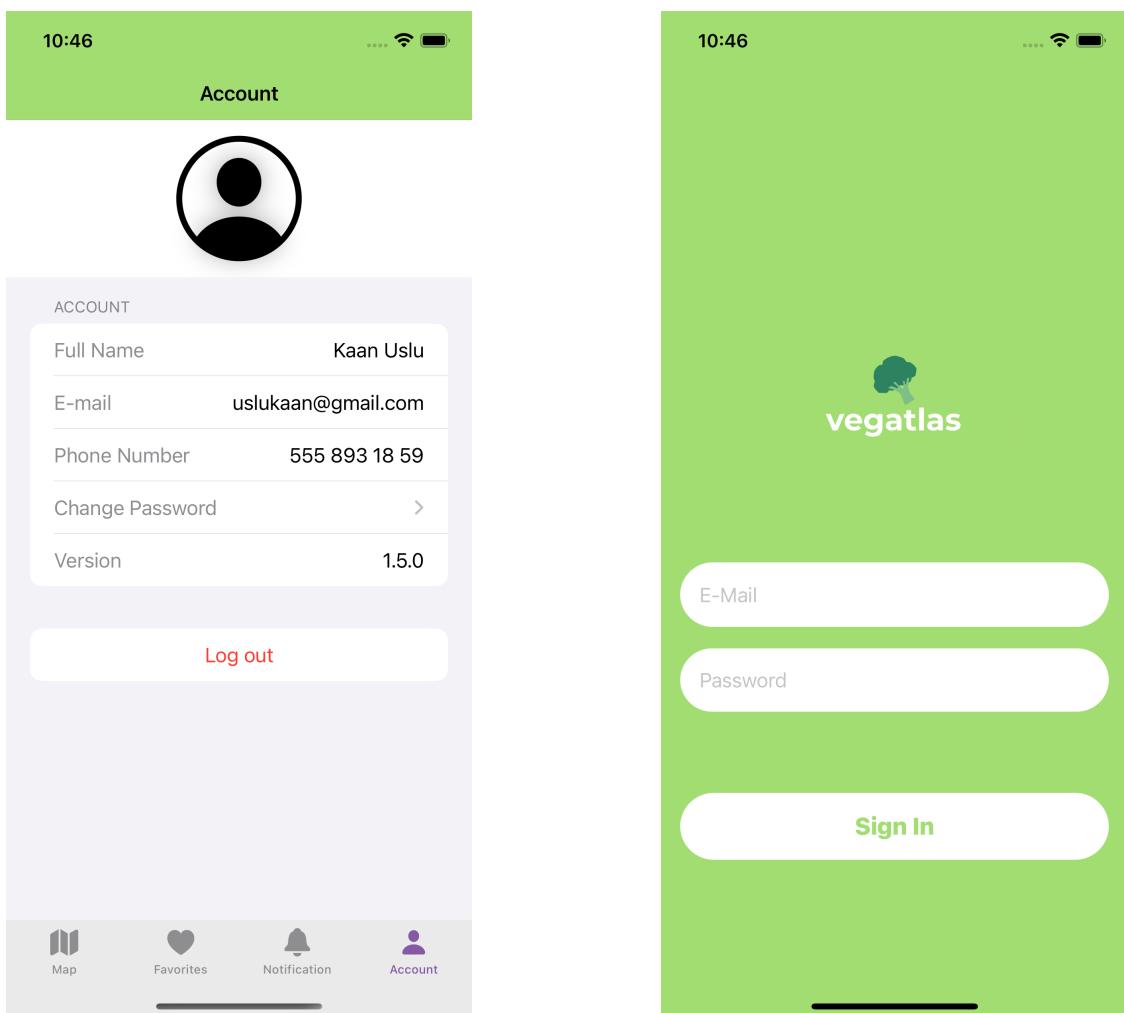
Result: An user signed in successfully.

5.14 Test Case (Logging out successfully)

Description: An user tries to log out successfully.

Test Steps:

1. Go to the account section.
2. Click log out button.



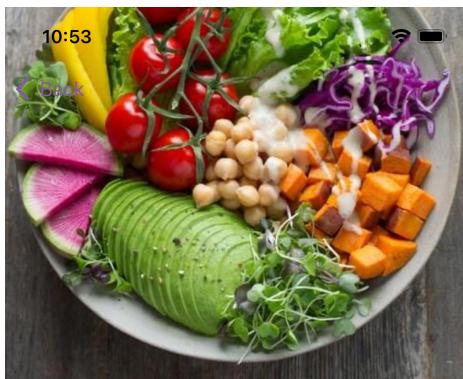
Result: An user logged out successfully.

5.15 Test Case (Make favorite restaurant successfully)

Description: An user tries to make restaurant favorite.

Test Steps:

1. Go to the map.
2. Click on one of the restaurants.
3. Click the heart button.(Button turns into filled heart)



Wiggling Plants 

Vegan Restaurant

Menu	
🍴	V'egg Harissa Bap
80₮	
🍴	Tempeh Blt Bap
70₮	
🍴	Wild Blueberry Hotcakes
85₮	
🍴	Toasted Coconut Bread
75₮	
🍴	Home Fries
45₮	
🍴	Brunch Bowl
130₮	

Map Favorites Notification Account

A screenshot of the mobile application showing the "Favorites" screen. It displays a card for "Wiggling Plants" with a rating of 0.0 and the text "Vegan Restaurant". The bottom navigation bar shows the "Favorites" tab is selected, indicated by a purple heart icon.

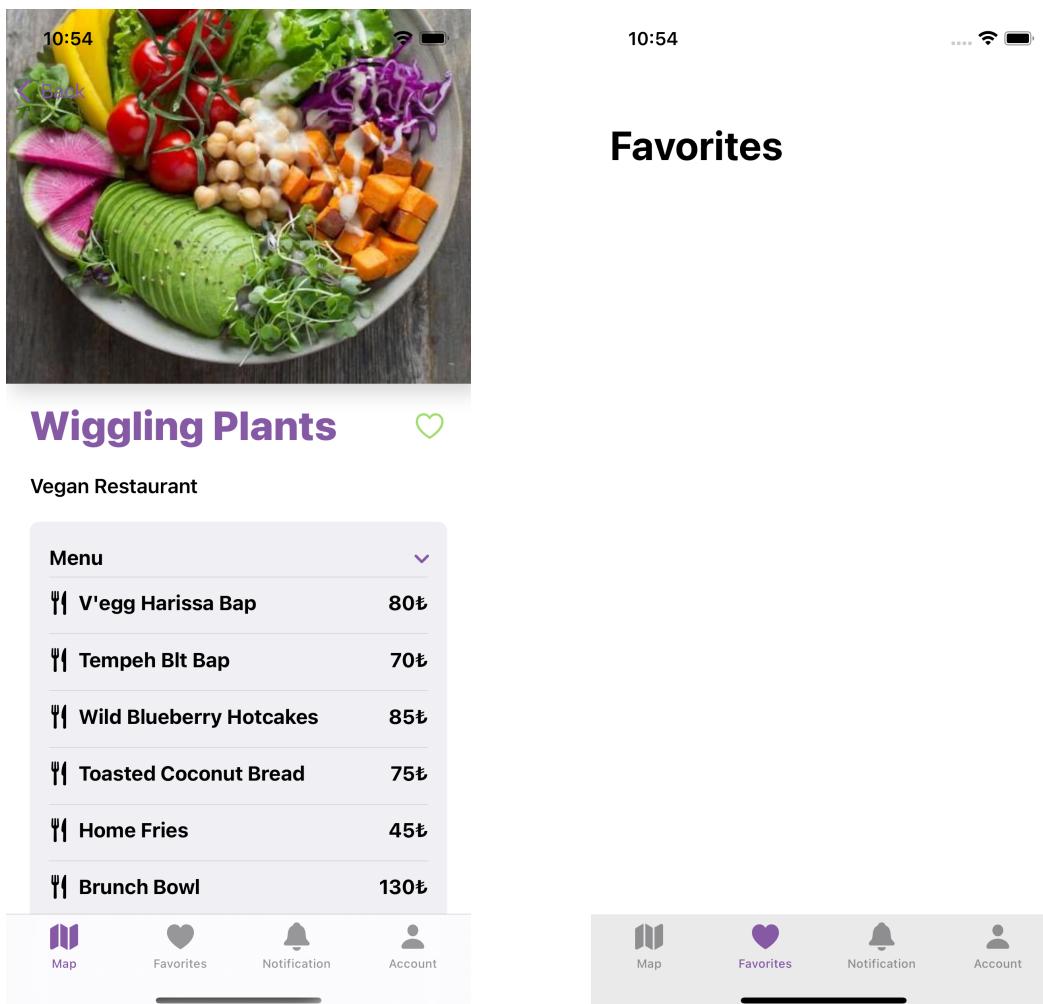
Result: An user make a restaurant favorite successfully.

5.16 Test Case (Make unfavorite restaurant successfully)

Description: An user tries to make restaurant unfavorite.

Test Steps:

1. Go to the map.
2. Click on one of the restaurants.
3. Click the heart button.(Button turns into non-filled heart)



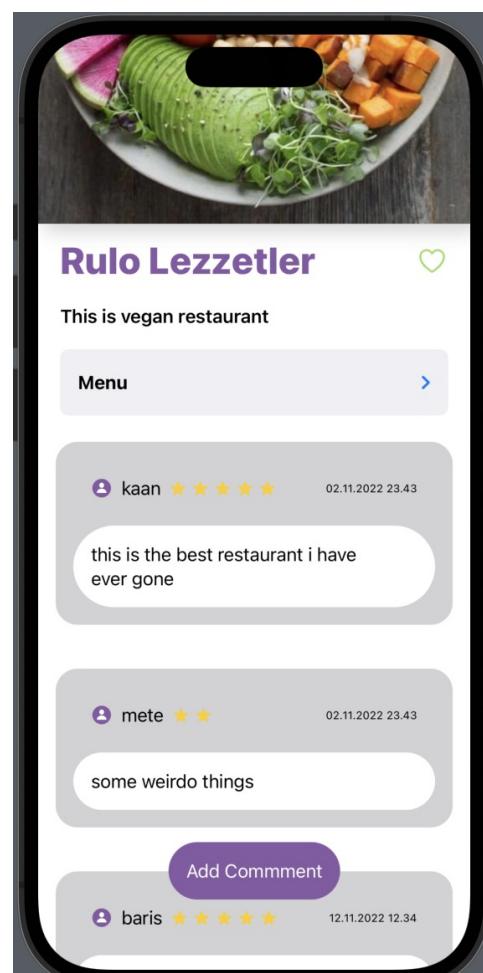
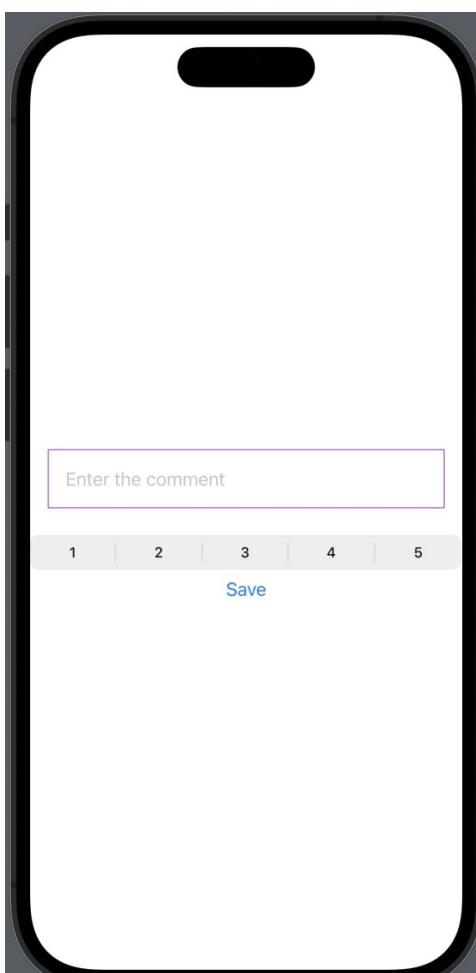
Result: An user make a restaurant unfavorite successfully.

5.17 Test Case (Make comment successfully)

Description: An user tries to make comment to a restaurant.

Test Steps:

1. Go to the map.
2. Click on one of the restaurants.
3. Click the add comment button.
4. Make comment and save it.



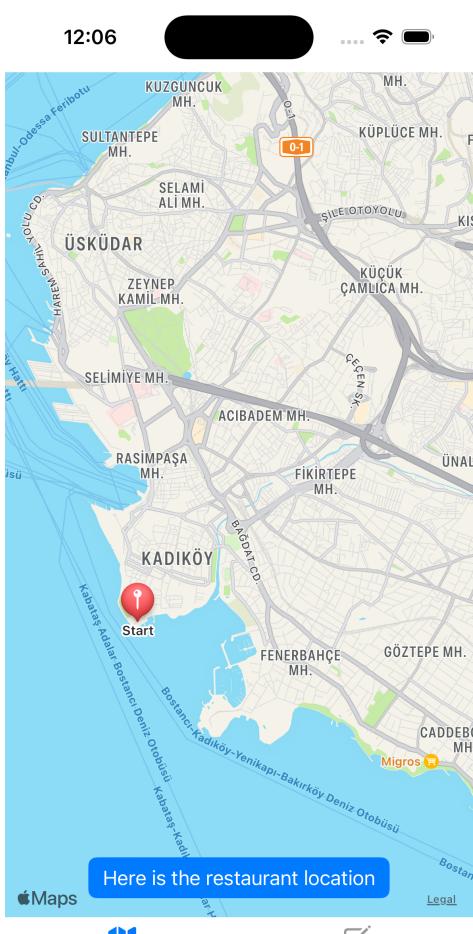
Result: An user make a restaurant unfavorite successfully.

5.18 Test Case (Add restaurant from admin panel)

Description: Admin tries to add restaurant to system.

Test Steps:

1. Go to the admin system.
2. Pin place and click here is the location.
3. Enter name, description, working hours and menu.
4. Click save.



A screenshot of a mobile application interface for adding a new restaurant. The screen displays several input fields:

- A text field containing "Tortilla".
- A text field containing "Tortilla vegan place.". This field has a red dotted underline, indicating it might be a required or invalid entry.
- A text field containing "9.00-22.00".
- A text field containing "Vegan".
- A text field containing "Tortilla durum".
- A text field containing "25".
- A text field containing "Nohut Tortilla".
- A text field containing "35".

At the bottom right is a blue "Save" button with a white plus sign icon above it.

 vegatlas	 Restaurants	 Tortilla	
<p>+ Start collection</p> <p>Notifications</p> <p>Restaurant</p> <p>Restaurants ></p> <p>Users</p>	<p>+ Add document</p> <p>Falafella</p> <p>Kahve Dunyasi</p> <p>Kofteeci Yusuf</p> <p>Rulo Lezzetler</p> <p>Tortilla ></p> <p>Wiggling Plants</p> <p>pepe rosso</p> <p>veganarsist</p>	<p>+ Start collection</p> <p>menu</p> <p>+ Add field</p> <p>description: "Tortilla vegan place."</p> <p>lat: 40.96838755675773</p> <p>lon: 29.035169134240302</p> <p>name: "Tortilla"</p> <p>restaurantType: "Vegan"</p> <p>workingHours: "9.00-22.00"</p>	

Result: Admin successfully adds restaurant.

5.19 Test Case (Add notification)

Description: Admin tries to add notification.

Test Steps:

1. Go to the admin system.
2. Enter title,description and date.
3. Click save.



 vegatlas	 Notifications	 	 9QAEe8yyZTry36A1BQ8d	
+ Start collection	+ Add document		+ Start collection	
Notifications >	9QAEe8yyZTry36A1BQ8d >		+ Add field	
Restaurant	r4ct34Z1bk08SzFpQKIO		date: "3.12.2022"	
Restaurants	text: "A great restauran opened in Kadikoy. Check thing out now!"		title: "A great place is opened."	
Users				

Result: Admin successfully adds restaurant.