

PROJECT PART 2

TEAM: Jeet Baru
Shalin Shah
Vihanga Bare

TITLE: Foodzie!

DESCRIPTION: Foodzie is an online restaurant discovery guide providing information on restaurant types, cuisines, reviews by critics and users, location and contact information. It is the one-stop-shop for all things food. Users will be able to search for outlets and customise results based on their liking. The application shall provide all-you-wish-to-know information about selected restaurant along with reviews by critics. Restaurant Owners get this platform to market their place by adding content to their restaurant's page.

ACTORS: Admin, Restaurant owners, Users, Critics

Project Requirements

➤ User Requirements:

User Requirement			
ID	Description	User	Priority
UR-01	As a new user, I should be able to create an account based on my role as user, food critic, restaurant owner, admin.	All Actors	Low
UR-02	As an existing user, I should be able to login into the system	All Actors	Low
UR-03	As an existing User or Food critic, I should be able to comment on and rate restaurants.	User, Food critic	Low
UR-04	As an existing user who is administrator, I should be able to delete objectionable comments.	Administrator	Low
UR-05	As a Food critic or Restaurant Owner, I should be able to reply to comments given by user.	Food critic, Restaurant Owner	Low
UR-06	As an administrator, I should be able to delete objectionable replies given by food critics or Restaurant Owner.	Administrator	Low

UR-07	As a Restaurant Owner, I should be able to create and add content for my own restaurant.	Restaurant Owner	Low
UR-08	As an administrator, I should be able to create and delete content for all restaurants.	Administrator	Low
UR-09	As a User or Food critic, I should be able to suggest content for any restaurant	User, Food critic	Low
UR-10	As a Food critic or User, I should be able to upload images for all restaurants	User, Food critic	Low
UR-11	As a Restaurant Owner, I should be able to upload images for my own restaurant	Restaurant Owner	Low
UR-12	As a User of Food critic, I should be able to delete my own comments, ratings and uploaded images.	User, Food critic	Low
UR-13	As a Restaurant Owner, I should be able to delete my own restaurant's content, replies and uploaded images.	Restaurant	Low
UR-14	As an Administrator, I should be able to create or delete any content across my application.	Administrator	Low
UR-15	All actors should be able to search restaurants based on different search filters in application	All actors	High
UR-16	As an administrator, I should be able to approve the suggestions to add content by User or Critic or Restaurant Owner.	User, Food Critic, Restaurant Owner	Low
UR-17	All actors should be able to delete their accounts	All actors	Low

➤ **Business Requirements:**

Business Requirements		
ID	Description	Topic Area
BR-01	Project must be implemented in Java using Object Oriented Design practices.	Implementation
BR-02	Must be a standalone application	Implementation

BR-03	The project development should be able to be tracked using GitHub	Submission
BR-04	Application data to be backed up and tracked using relational database	Data Storage
BR-05	The code should be refactored to follow industry recognized design patterns	Refactoring
BR-06	The critic ratings should have a higher weightage than a user's rating	Rating

➤ **Functional Requirements:**

Functional Requirement			
ID	Description	Topic Area	Priority
FR-01	Password based authentication for all users while login	Authentication	Low
FR-02	All users should be able to reset passwords for their accounts	Authentication	Low
FR-03	The rating displayed is the weighted average of the critics and user ratings	Rating	High
FR-04	The user's search should be filtered according to price, cuisine type, working hours or location.	Search	High

➤ **Non - Functional Requirements:**

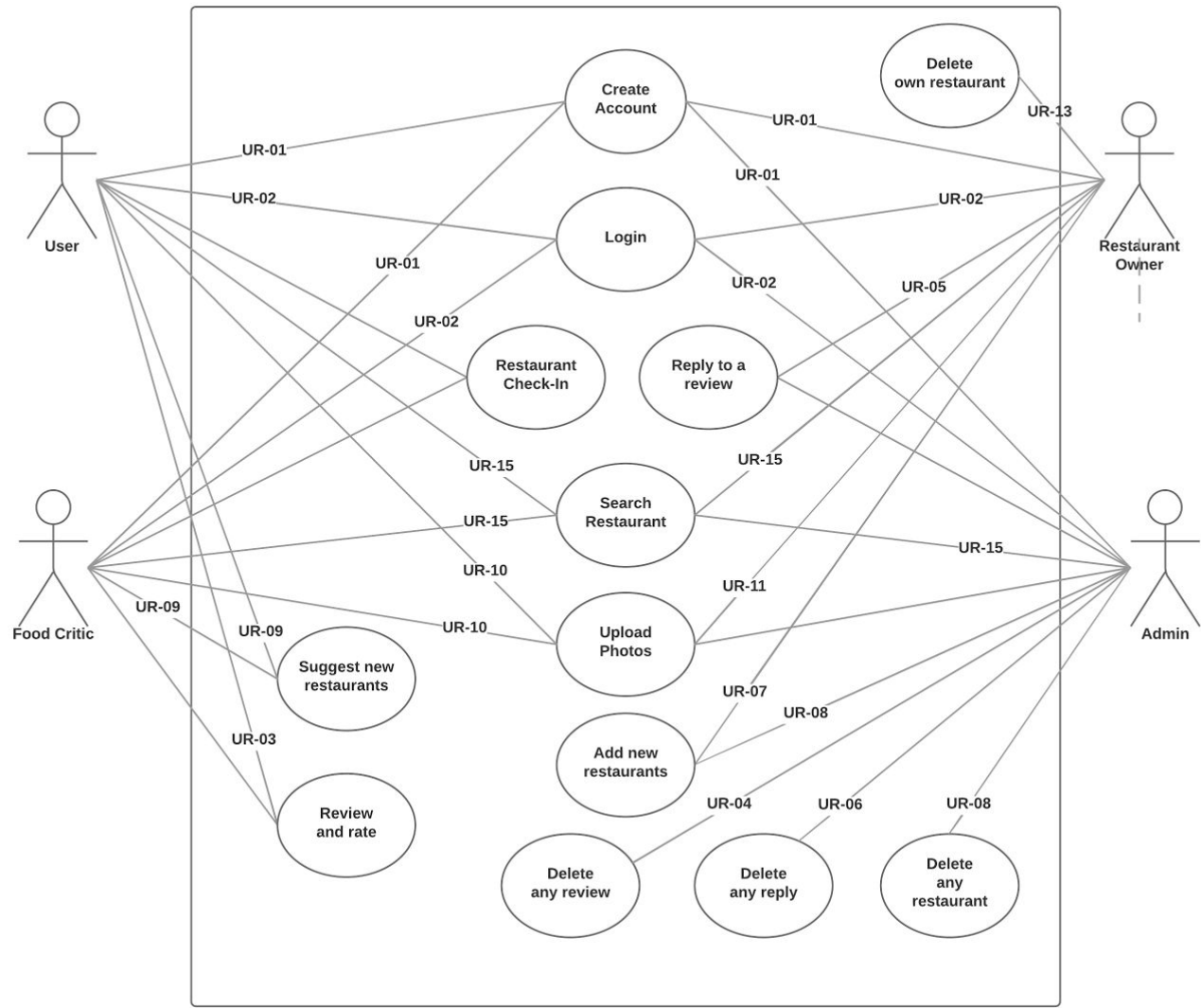
Non-Functional Requirement			
ID	Description	Topic Area	Priority
NFR-01	Encryption of passwords using hashing techniques.	Authentication	High
NFR-02	The system should be available to all users 24x7	Reliability /Performance	High
NFR-03	Application be able to support at least N users simultaneously using the application.	Performance	High
NFR-04	Application to generate daily, weekly, monthly and incremental data backups from database	Backup	Medium

NFR-05	Application to fetch weekly and monthly user activity reports from database for administrators to monitor the health or status of application	Sanity Check	Medium
--------	---	--------------	--------

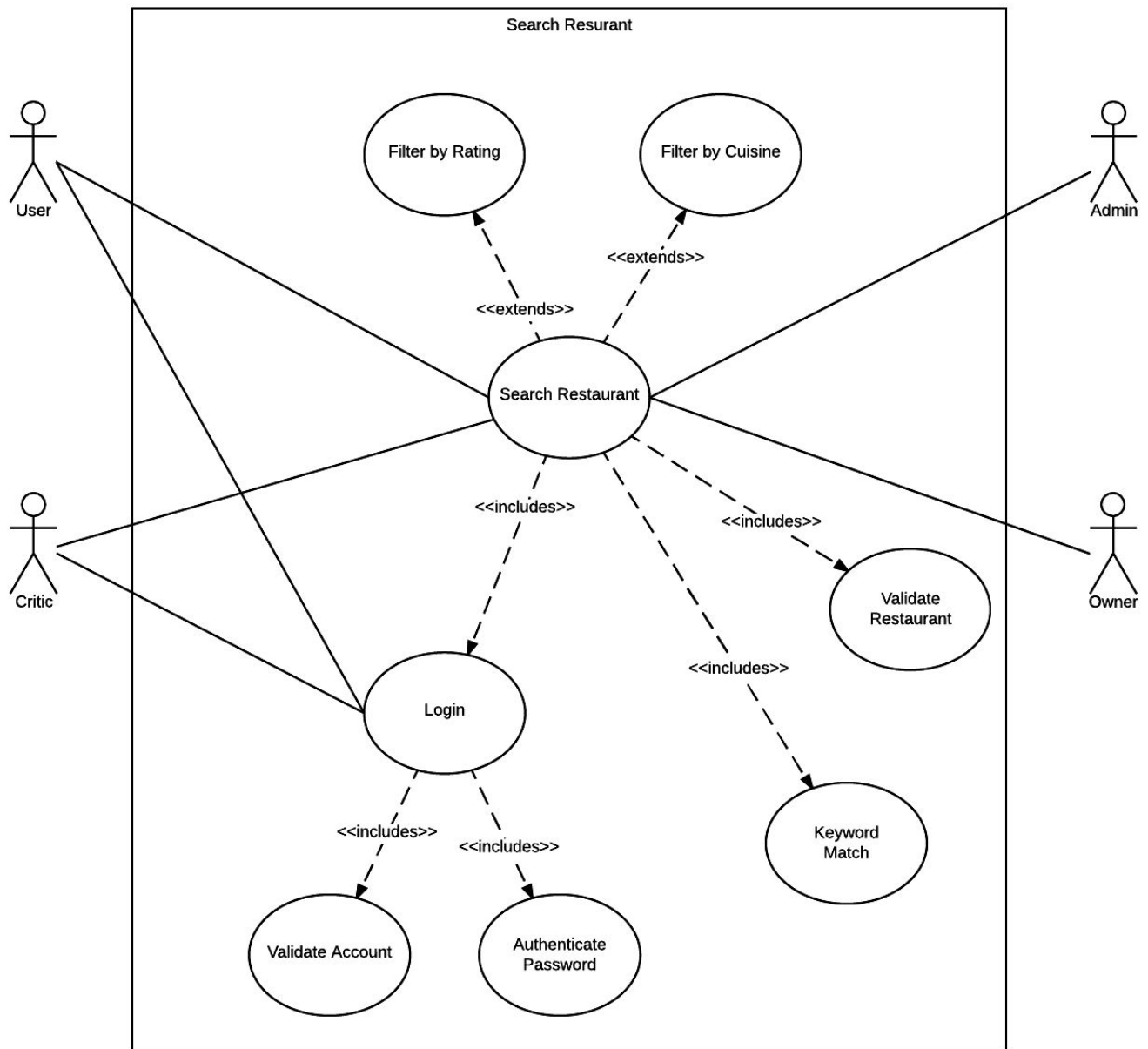
➤ Use Case Overview:

CASE OVERVIEW

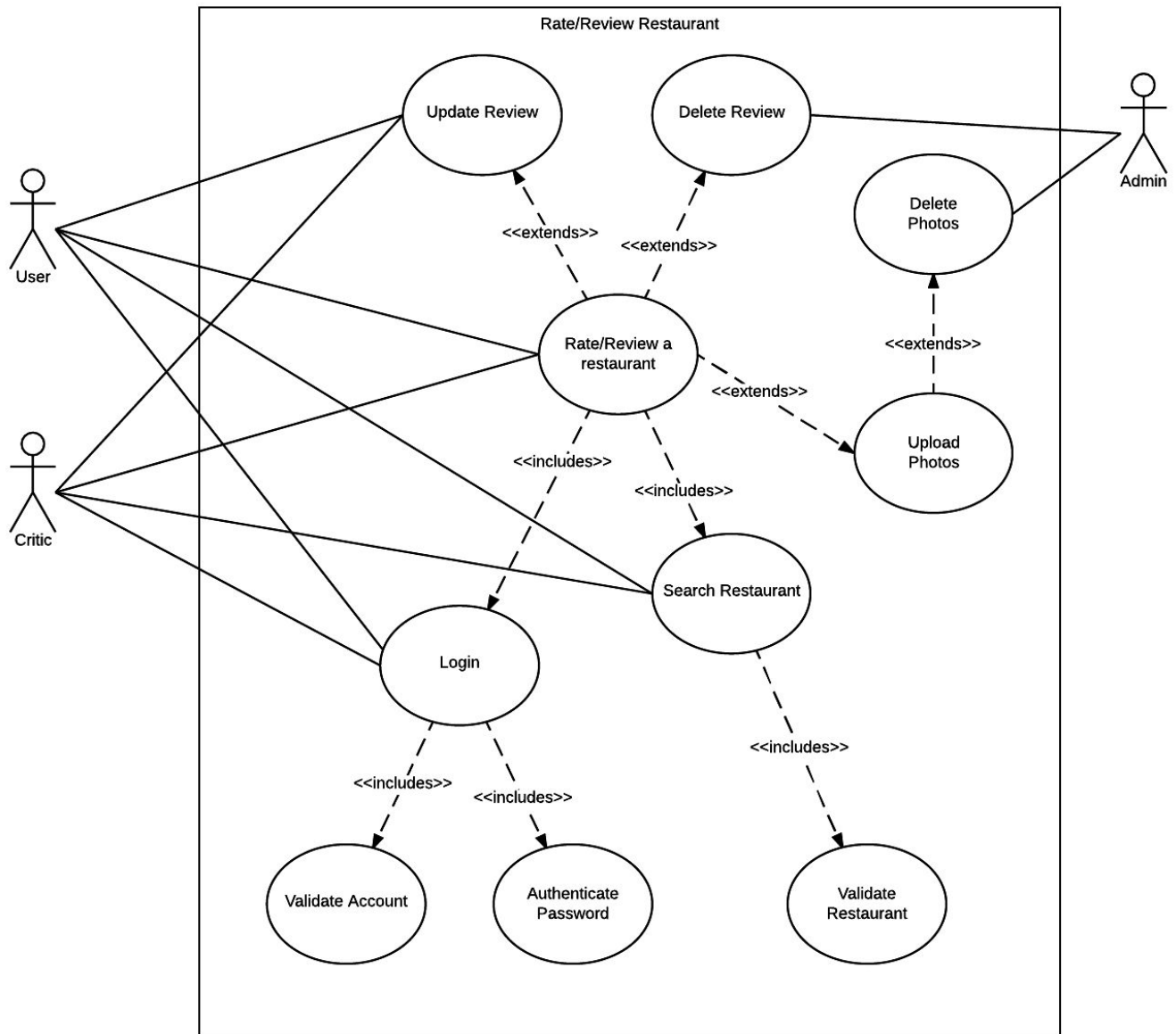
Foodzie! | October



➤ Use Case Sub Diagrams:
Use case - Search for a restaurant:



Use case - Rate/Review a restaurant:



➤ **Use Case Documentation:**

Use Case ID:	UC-01
Use Case Name:	Register A User
Description	Register a user and create a new login for this user on sign-up

Actors:	All Actors		
Pre-Conditions:	User does not have an account		
Post-Conditions:	User will have a new account		
Frequency of Use:	Every time a new user is added to the system		
Flow of Events:		Actor Action	System Response
	1	Select Register/Sign-up	Show the sign-up screen
	2	Enter user information, email address and select authentication using email address.	Store user information, email address.
	3		Authenticate user's email address. Notify user.
	4	User is signed-up in system	
Variations:			
Exceptions:	New account creation will fail if account already exists		
Developer Notes:			

Use Case ID:	UC-02
Use Case Name:	Login
Description	User can login into the system

Actors:	All Actors		
Pre-Conditions:	User is not logged into the system		
Post-Conditions:	User logged into the system and can see the search/list restaurants view.		
Frequency of Use:	Every time user logs into the system		
Flow of Events:		Actor Action	System Response
	1	User enters his username/email address and password	System authenticates username and password.
	2		If authentication successful, open the search /list restaurants view.
	3		If authentication not successful, throw a prompt saying password or username does not exist.
	4		Request user to re-enter username/password combination.
Variations:	3. Display option to handle case of forgot password. User selects forgot password. System requests user to enter email address for sending password reset link. User enters email address. System sends across the password reset link. User resets password through link received in email. New password recorded into system.		
Exceptions:	Block user if user enters wrong username/password combination for 3 consecutive attempts in past half an hour.		

Developer Notes:	
-------------------------	--

Use Case ID:	UC-03
Use Case Name:	Add Rating
Description	User or a Critic can provide a rating write a review for a restaurant

Actors:	User, Critic		
Pre-Conditions:	User or Critic should be logged in		
Post-Conditions:	The User or Critic's rating and review is added to the restaurant. The ratings and review section of the restaurant is updated.		
Frequency of Use:	Anytime of the day		
Flow of Events:		Actor Action	System Response
	1	Select a listed restaurant	Open the details page for the restaurant
	2	Select Add rating	Popup a window where user can provide ratings and review
	3	Provide a rating, write a review and click submit	The rating and review is added to the database for the restaurant.
	4		The overall rating and reviews view for the restaurant is updated
Variations:	2. If the User/Critic has already provided a rating and review, the window would allow to edit it		
Exceptions:	3. User/Critic does not provide any rating or review and clicks cancel		
Developer Notes:	The updated rating should be weighted accordingly for a user or a critic		

Use Case ID:	UC-04
Use Case Name:	Add new restaurant
Description	New restaurant information can be added to the database

Actors:	Admin, Restaurant owner		
Pre-Conditions:	A restaurant with same information(name, address, telephone number) should not exists in the database		
Post-Conditions:	A restaurant with provided information will now be added to database, available for users to review and check-in		
Frequency of Use:	Admin may use this when a new restaurant is suggested, restaurant owner wishes to add his restaurant.		
Flow of Events:		Actor Action	System Response
	1	Select add new restaurant	A form shall be presented for the actor to fill.
	2	Must fill the compulsory fields of the form	Create a new entry in the database with the information provided.
	3	Select confirm	Sections for reviews and ratings for the restaurant are created in the database
	4		A new restaurant details page for the restaurant is created
Variations:	2. If the actor does not have information for the compulsory fields in the form the new entry cannot be added.		
Exceptions:			
Developer Notes:	Verification of information provided must be done, if it is done after adding restaurant to database, the unverified restaurant must be deleted.		

Use Case ID:	UC-05
Use Case Name:	Search Restaurant/ List Restaurant
Description	Allows user to search restaurant based on keywords and applied filters

Actors:	All actors		
Pre-Conditions:	User must be logged into the system		
Post-Conditions:	User gets a list of restaurants filtered by the search settings		
Frequency of Use:	Every time any actor searches for a restaurant		
Flow of Events:		Actor Action	System Response
	1	The actor searches restaurant based on restaurant name as keyword	The system displays a list of matching restaurants
	2	Actor might narrow down the search based on filters related to location, cuisine type, cost.	The system creates a list of restaurants using the search filter
	3		The listed is sorted based on overall ratings of the restaurants in the list
	4		The list of restaurants based on the search filter is displayed
Variations:			
Exceptions:	1. No restaurant in the database matches the actor's search Systems prompts the User/Critic to suggest to add the restaurant. System prompts the Owner/Admin to add the restaurant.		
Developer Notes:			

Use Case ID:	UC-06
Use Case Name:	Delete Review
Description	User/Critic can delete his own review for a restaurant

Actors:	User/Critic		
Pre-Conditions:	User must be logged into the system User is already on the review page where he wishes to remove his review		
Post-Conditions:	User's review is removed. Rating is re-calculated and updated		
Frequency of Use:	Every time any actor deletes his review for a restaurant		
Flow of Events:		Actor Action	System Response
	1	Click delete review button	Display confirmation dialog box
	2	Confirm deletion.	Delete the review. Update the ratings based on new calculations.
	3		Display new sorted list of reviews based on new calculations.
	4		Display sorted list on the review page
Variations:			
Exceptions:	Review does not exist in database. User selects not to delete request during confirmation dialog box		
Developer Notes:			

Use Case ID:	UC-07
Use Case Name:	Suggest to add a Restaurant
Description	Critic/User can suggest to add a restaurant in the database

Actors:	Critic/User		
Pre-Conditions:	Critic/User must be logged into the system		
Post-Conditions:	The approval request is send to the Admin		
Frequency of Use:	Every time Critic/User suggests a restaurant		
Flow of Events:		Actor Action	System Response
	1	The actor searches restaurant based on restaurant name as keyword	The system does not find any matching results
	2	Actor selects suggest a new restaurant	Display suggest a new restaurant with required fields
	3	Provide the details and submit the request	Check if all the required details are added. Check if the restaurant exists. Notify the admin about the request
Variations:	3. If all the required details are not entered, return to suggest restaurant page 3. If the restaurant already exists, return to the search page and show an error to the actor		
Exceptions:	3. If cancel is selected instead of submit , return to the search results.		
Developer Notes:			

Use Case ID:	UC-08
Use Case Name:	Delete Account
Description	Registered user must be able to delete account

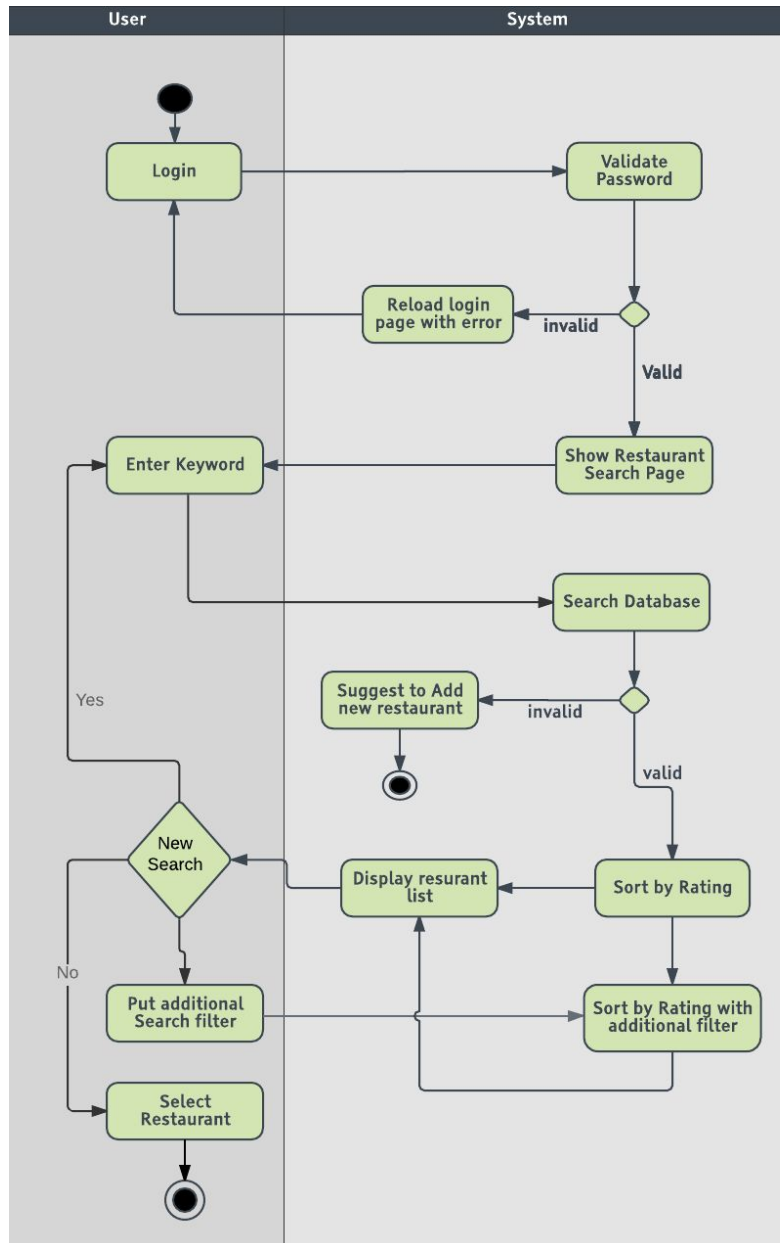
Actors:	All Actors		
Pre-Conditions:	User has an account		
Post-Conditions:	User will have deleted his account		
Frequency of Use:	Every time an existing user requests for account deletion		
Flow of Events:		Actor Action	System Response
	1	Select Delete Account	Ask for confirmation
	2	User confirms to delete his account	All reviews and ratings contributed by the user must be deleted
	3		User personal information must be deleted from the database
	4	User account is deleted	
Variations:	User ratings and reviews remain in the system and contribute to database.		
Exceptions:			
Developer Notes:			

➤ Activity Diagram:

Use Case Name: Search Restaurant

Use Case ID: UC-05

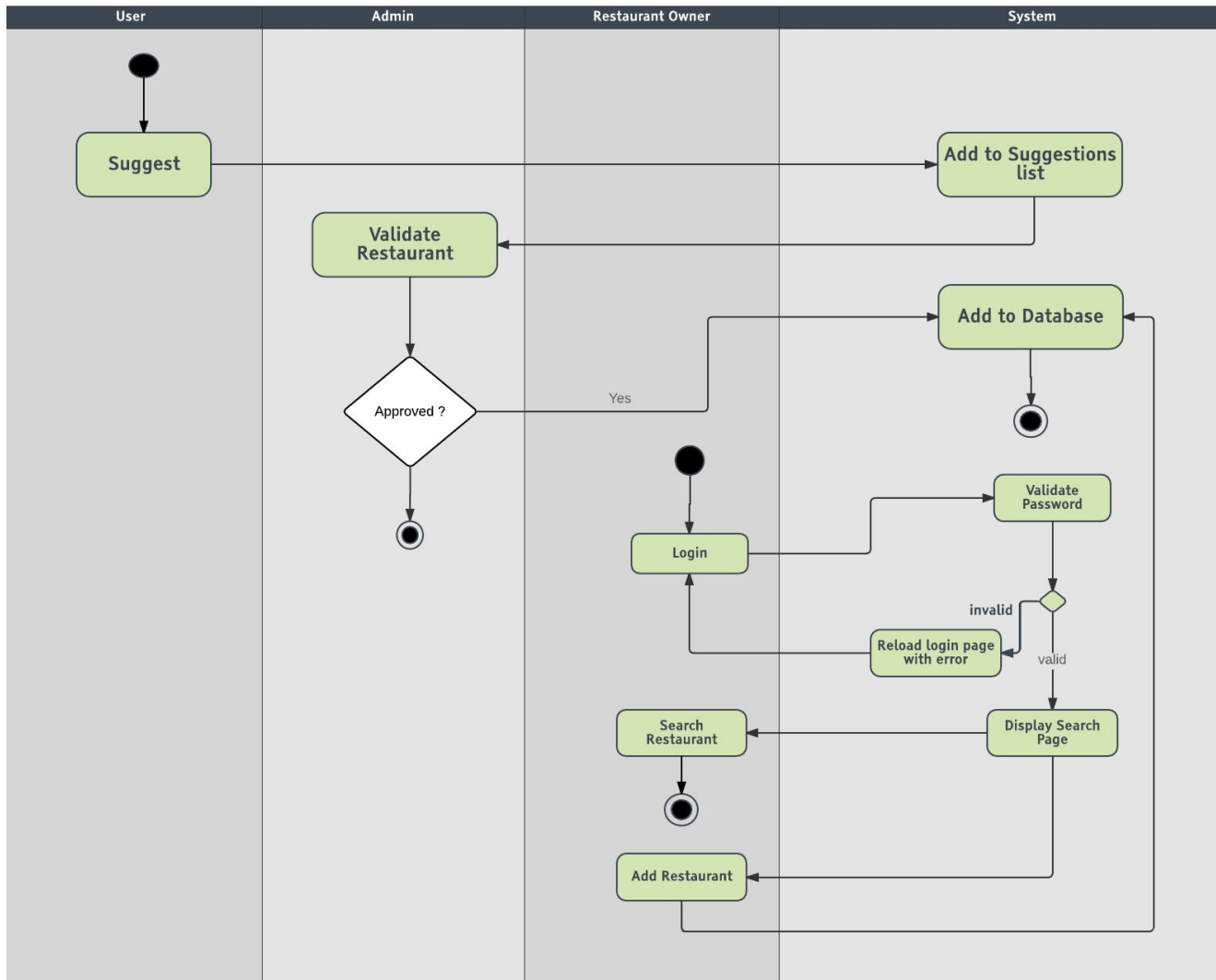
Requirement ID: UR-15



Use Case Name: Add Restaurant

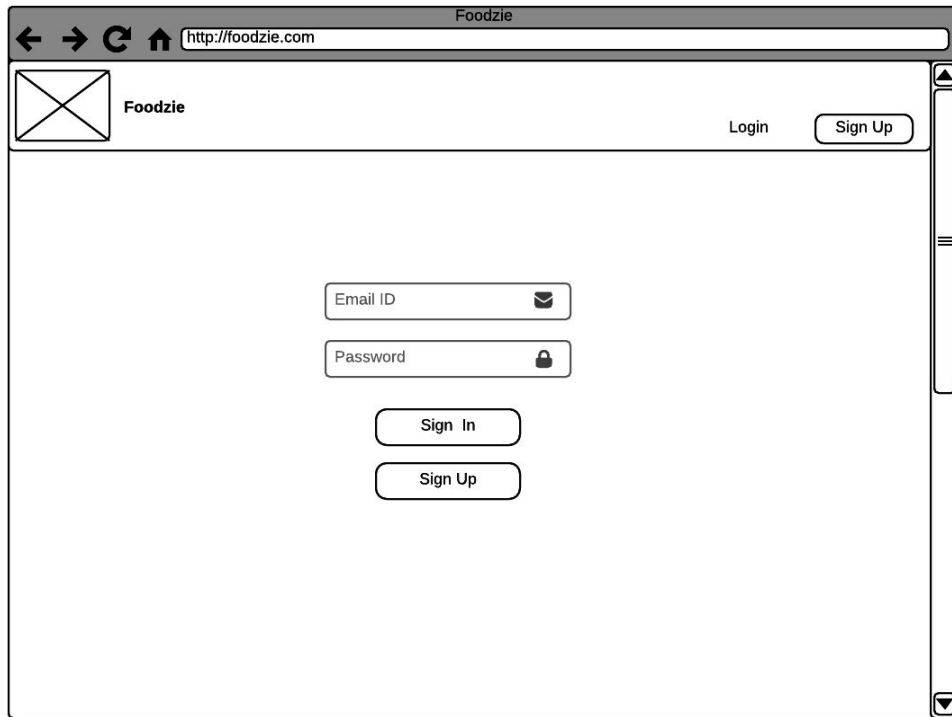
Use Case ID: UC-04

Requirement ID: UR-7



➤ UI Mockups:


Landing/First Page:



SignUp Page:

Foodzie

http://foodzie.com/signup

Foodzie

Login

Welcome,

First Name

First Name

Last Name

Last Name

Address Line 1

Address Line 1

Address Line 2

Address Line 2

City

City

State

State

Country

Country

Phone

Phone

Date of Birth

DD/MM/YYYY

Email ID

Email ID

Password

Password (min 6 characters)


Re-enter Password

Password (min 6 characters)


Sign Up

Display Outlet/Restaurant Details Page:

Foodzie.com

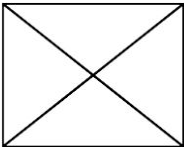


Foodzie!



Search New

Logout



Outlet Name

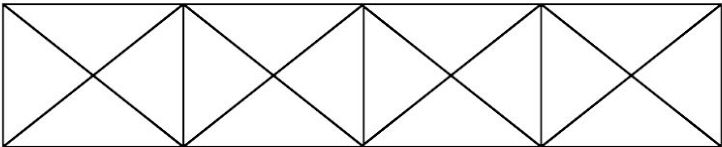
Cuisine Type

Address

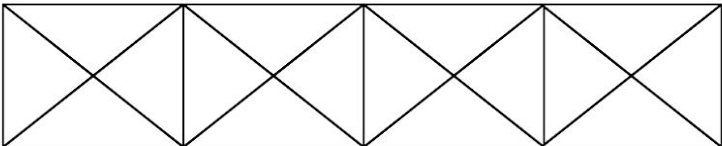
Phone Number

☆☆☆☆☆

Menu



Pictures



Reviews

Write Review

Edit your Review

Reviewer 1

☆☆☆☆☆

Comment 1


Reviewer 2

☆☆☆☆☆

Comment 2

Search Restaurant Page:


Foodzie.com



Foodzie!

Welcome, User!

Logout

 search

Select Cuisine/ Outlet Type

Cuisine 1

Cuisine 2


Cuisine 3

Cuisine 4

Cuisine 5

Search Results Page:

Foodzie.com

Foodzie!

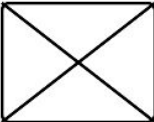
Welcome, User!

Logout

Outlet Type

Add Filter

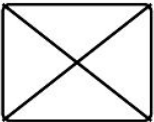
Search Keyword



Outlet 1

☆☆☆☆☆


Address of Outlet 1 | Phone Number



Outlet 2

☆☆☆☆☆


Address of Outlet 2 | Phone Number



Outlet 3

☆☆☆☆☆

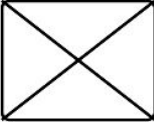
Address of Outlet 3 | Phone Number



Outlet 4

☆☆☆☆☆


Address of Outlet 4 | Phone Number



Outlet 5

☆☆☆☆☆

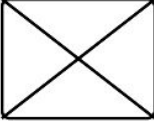
Address of Outlet 5 | Phone Number



Outlet 6

☆☆☆☆☆


Address of Outlet 6 | Phone Number



Outlet 7

☆☆☆☆☆

Address of Outlet 7 | Phone Number



Outlet 8

☆☆☆☆☆

Address of Outlet 8 | Phone Number

32_Foodzie!_Part2.pdf

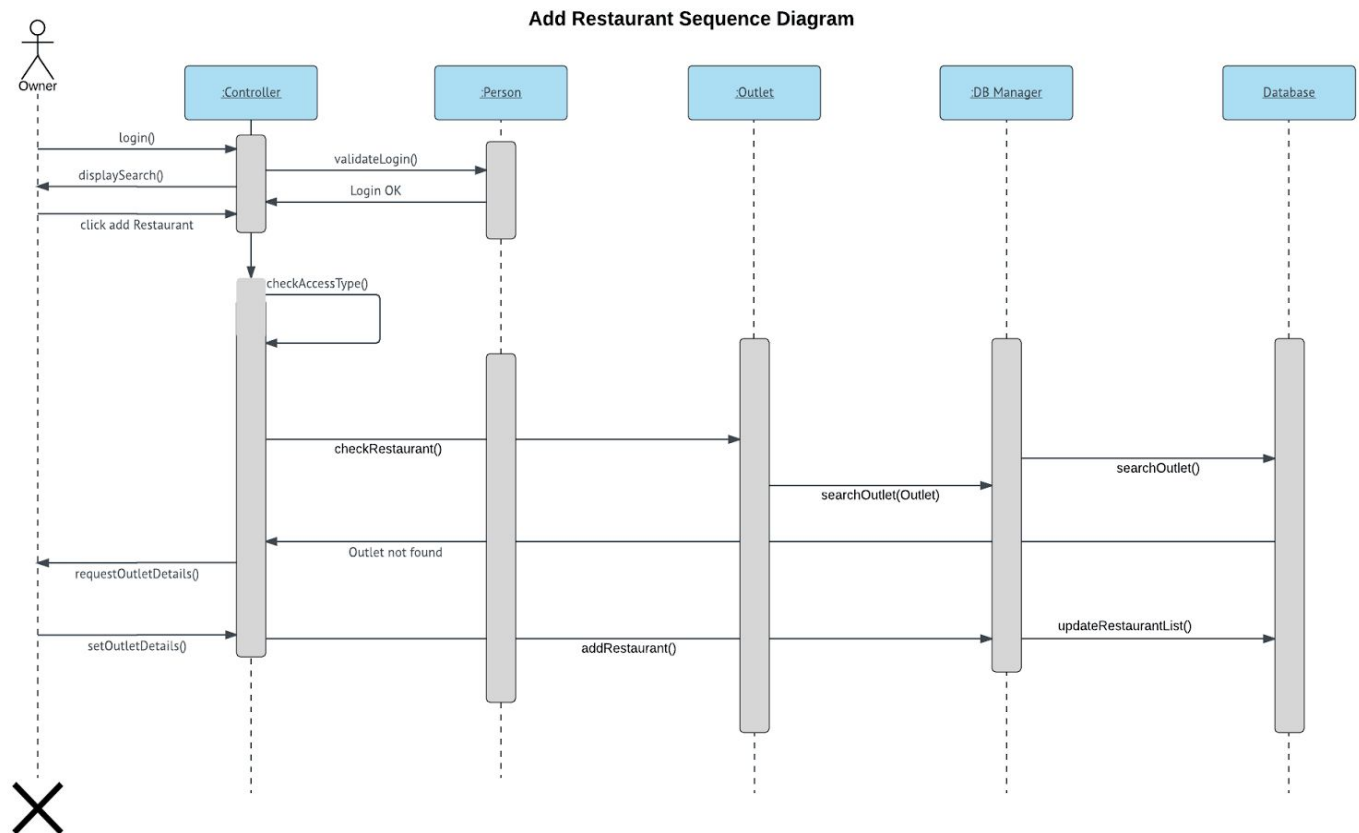
Page 20

➤ User Interactions / Sequence Diagrams::

Use Case Name: Add Restaurant

Use Case ID: UC-04

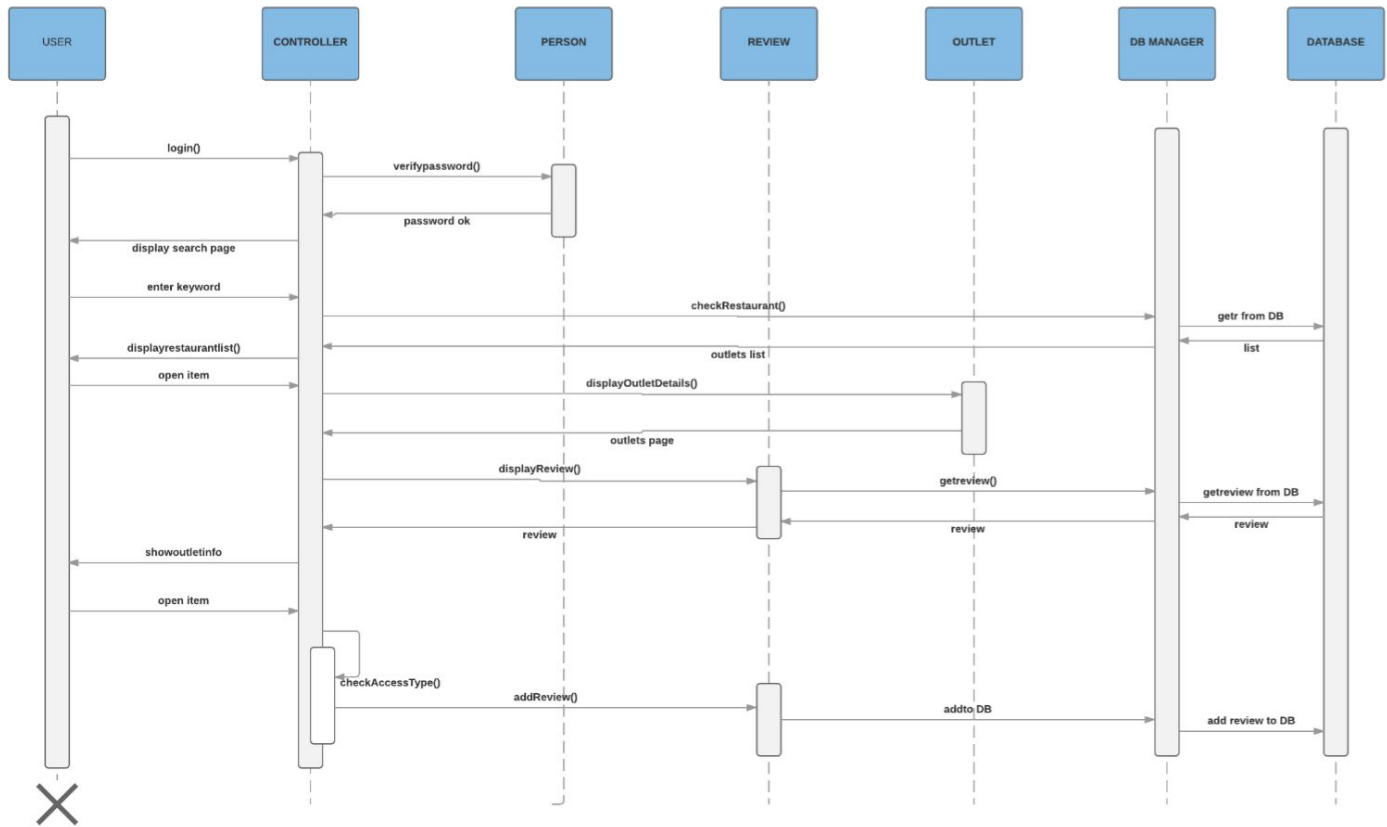
Requirement ID: UR-7



Use Case Name: Rate/Review Restaurant

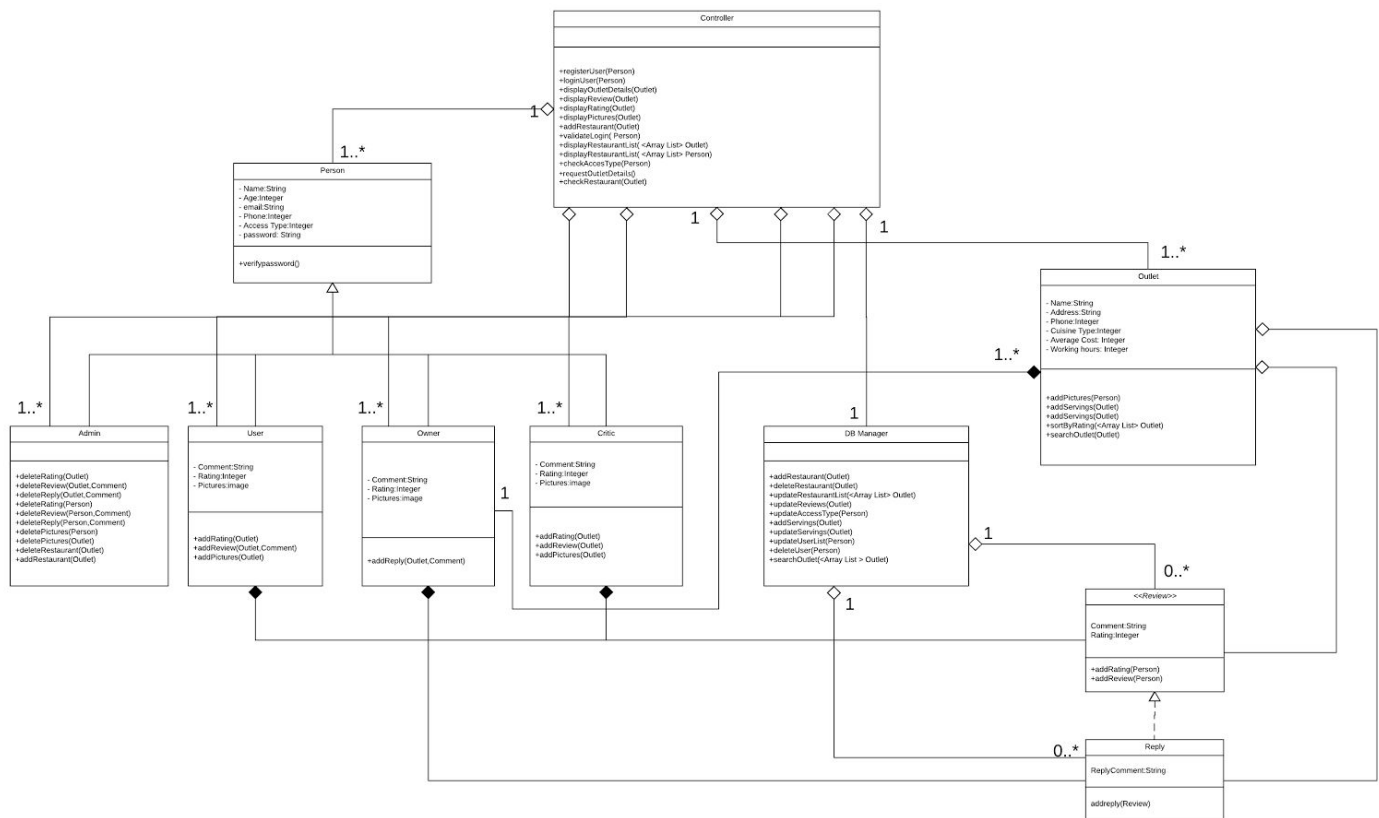
Use Case ID: UC-03

Requirement ID: UR-03



➤ Class Diagram:

Class Diagram



Data Storage:

For connecting to database, we are using Hibernate with MySQL database.