Foodle

Software Requirements Specification

08.04.2022

Name	**-**	**ID**
Bekir Nazmi Görkem	-	150118017
Berk Kırtay	-	150118043
Burak Çağlayan	-	150118027
Erkam Karaca	-	150118021
Rasim Sadıkoğlu	-	150118009

Prepared for
CSE3044 Software Engineering Term Project

Contents

1. INTRODUCTION **3**

1.1 PURPOSE 3
1.2 SCOPE 3
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 3
1.4 REFERENCES 3
1.5 OVERVIEW 3

3. SPECIFIC REQUIREMENTS **5**

3.1 EXTERNAL INTERFACE REQUIREMENTS 5
3.1.1 User Interfaces 5
3.1.2 Hardware Interfaces 6
3.1.3 Software Interfaces 6
3.1.4 Communications Interfaces 6
3.2 FUNCTIONAL REQUIREMENTS 6
3.2.1 Nearby Restaurants 6
3.2.2 Filter Restaurants by Meal 7
3.2.3 Post your experience 7
3.3 NON-FUNCTIONAL REQUIREMENTS 8
3.3.1 Performance 8
3.3.2 Reliability 8
3.3.3 Availability 8
3.3.4 Security 8
3.3.5 Maintainability 8
3.3.6 Portability 8
3.4 INVERSE REQUIREMENTS 8
3.5 DESIGN CONSTRAINTS 8
3.6 LOGICAL DATABASE REQUIREMENTS 9

4. UML DIAGRAMS **9**

4.1 USE CASES 9
4.1.1 Use Case #1 9
4.1.2 Use Case #2 10
4.2 CLASSES / OBJECTS 10
4.3 SEQUENCE DIAGRAMS 11
4.3.1 Sequence Diagram for Use Case 1 11
4.3.2 Sequence Diagram for Use Case 2 12

5. ASSIGNMENT TABLE **13**

1. Introduction

The introduction part of the Software Requirements Specification document provides a brief overview of the rest of the document as follows:

1.1 Purpose

The purpose of this document is to gather up all the features that comes into mind and give a detailed description of the requirements needed for this software. It will also explain which technologies should use together, what will be the interactions between user and app. This will help the software engineer to create the program in her/his mind, what to use and not to use during the creation phase and what to pay attention to while doing it.

1.2 Scope

The purpose of "Foodle" is to create an online environment that receives feedback from users and filters them accordingly.

Foodle will be designed to help our users with their restaurant choices. There are several apps like Foodle, but they do not provide enough information. For instance, they use faked photos of food, outdated menu pricing. Thanks to the information provided by GPS and other users, our users will be able to easily access the places in the appropriate location and information about those places with the filters they choose. Then, like any other user, they can rate the restaurant by sharing their opinions about that place, photos of the food, prices, and other things.

1.3 Definitions, Acronyms, and Abbreviations

USER - Actor who uses application
FEEDBACK – Comments made by users
GPS - The abbreviation of Global Positioning System
HASHING - A one-way transformation of a string into a fixed length value
QUERY - A request against database for information retrieval
SESSION - User interaction with web service in a given time frame
NAVBAR – Navigation bar.

1.4 References

1) **IEEE Std 830-1998**

1.5 Overview

The remainder of this document provides a general description of the software, specific requirements, and UML diagrams. In the second part of document provides information about features, functionalities, and general constraints of software.

The third part explains the requirements of the system in detail. This section also will give the requirements that are used to guide the project's software design, implementation, and testing.

The fourth part will provide the uses cases, objects, and necessary diagrams of software.

2. General Description

Our system aims users to have prior knowledge about prices and quality of the service before they go to a restaurant. They can be seen the menus of restaurants, prices of the meals and reviews that has been written by other users. Our system mainly provides the frequently changing prices of meals as timely as possible and help users to decide where they want to go.

2.1 Product Perspective

There are some products that are opponent of our project. One of them is Zomato. The aim of this website is to give information about the places. This information consists of the price of meals, atmosphere of the place and photos of the meals. From the user's perspective two systems are mostly equivalent. The main difference is that on Zomato reviews are written by admins but in our system, users will be able to write reviews. The benefit of this feature is that if our app happens to be used by many users, majority of the reviews and prices will be up to date.

The other chance that users can use is some social media accounts. However, they are not well organized. Possibility of finding useful review at this point is nearly impossible. The information about food is insufficient because only one person or a small group of people make all the reviews. And as we stated above, those accounts do not solve our main problem which is having knowledge about frequently changing prices.

2.2 Product Functions

Some of the functions of our system is stated below:

- Users can search the nearby restaurants with GPS.
- Users can see the menus of the restaurants.
- Users can see the prices, photos, and reviews of the meals on the menu.
- Users can rate the meals and write their comments about meals and service.
- Users can list the restaurants by their rate.

2.3 User Characteristics

The mobile application users can use the application to find a restaurant, rate meal or write a review for them. This means that the user must be able to search for restaurants and choose a restaurant. He can look at the menu and see the prices. For the users to get a relevant search result there are multiple criteria the users can specify, and all results matches all of those. The restaurant owners will not use the mobile application as owner.

2.4 General Constraints

There are a lot of constraints that limits the developer on this project. Some of them are stated below:

- Memory limit of database to store information's about users and restaurants. Also, photos will take up large space.
- Using the RAM of device efficiently.
- The Internet connection is also a constraint for our project. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to run.
- Queing the requests if there are so many users.

2.5 Assumptions and Dependencies

Since Foodle is a mobile application, in coding process we may have some troubles to adjust user interface to different type of devices. For instance, if we have a navbar with height 80px and it seems good in a 1080x1920 device however it may not be seen as fit as previous device in another device that has 1440x2960 pixel ratio. Therefore we may have to change our navbar height according to some other ratios.

Other dependency we have to consider is RAM and memory usage of mobile devices. We have to store data in cloud based database as much as possible in order not to take up large amounts of space in device.

3. Specific Requirements

3.1 External Interface Requirements

This section has detailed explanation of user interface of this application.

3.1.1 User Interfaces

If a user is not logged-in in the mobile application, log-in page should be displayed to the user. If the user has not registered, he/she should be able to do that on the log-in page. If the user is logged-in, he/she should be able to see the search page directly when the application is opened. Every user should have a profile page where they can edit their credentials such as e-mail, password etc. The icon shows on top-right corner which the user can click to navigate to his/her profile page. At the top there is a search bar where you can search among the restaurants.

When a user searches, view should be sorted by ratings by default. The sorting header allows the user to sort the results according to ratings, price, distance, restaurant name and restaurant type. There is also a filtering option, where the user can choose to filter the results by increasing or decreasing the price or distance range. Each result item includes information about the restaurants, restaurant menu and pricing, a link to the restaurant's web page.

In the map view each restaurant is represented by a pin. Next to every pin there is an information link which provides a more detailed description of the restaurant, as mentioned for the list view.

If a user wants to leave a review to a restaurant, he/she must log-in to his/her account first. Afterwards, he/she rates the service of the restaurant and also optionally he/she can attach a photo to the review.

3.1.2 Hardware Interfaces

Since the mobile application doesn't have any designated hardware, it does not have any direct hardware interfaces. The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone.

3.1.3 Software Interfaces

The mobile application communicates with the GPS application in order to get geographical information about where the user is located and the visual representation of it, and with the database in order to get the information about the restaurants. The communication between the database consists of operation concerning both reading and modifying the data, while the communication between the database and the mobile application consists of only reading operations.

3.1.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the mobile application.

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 Nearby Restaurants

3.2.1.1 Introduction / Description

You can see the restaurants near you via the mobile app.

3.2.1.2 Inputs / Display

- Current Location

3.2.1.3 Processing

App calculates distance between the restaurants and you.

3.2.1.4 Outputs

Shows you the available restaurants on the map.

3.2.1.5 Constraints

None

3.2.1.6 Error/Data Handling

If GPS is not available or not working correctly you can enter your address manually.

3.2.2 Filter Restaurants by Meal

3.2.2.1 Introduction / Description

You can filter restaurants by your choice in different categories.

3.2.2.2 Inputs / Display

You must select an option.

3.2.2.3 Processing

App filters restaurants or meals that you didn't choose.

3.2.2.4 Outputs

Shows you the available restaurants or meals in a list.

3.2.2.5 Constraints

Categories are predefined.

3.2.2.6 Error/Data Handling

None

3.2.3 Post your experience

3.2.3.1 Introduction / Description

After you had a meal in restaurant you can post your experience and rate the meal and the restaurant. You can add photos of the meal and the restaurant.

3.2.3.2 Inputs / Display

Rating

Text message

Photos

3.2.3.3 Processing

App formats your post and sends it to the server.

3.2.3.4 Outputs

None

3.2.3.5 Constraints

None

3.2.3.6 Error/Data Handling

If there is no available internet connection, post will be saved on your device and will be posted after internet connection is available.

3.2.4 Check others' experience

3.2.4.1 Introduction / Description

Before going to any restaurant, you can check other people's post and decide whether to go or not. Also, you can rate other people's review by its correctness.

3.2.4.2 Inputs / Display

Select a post

Rate (optional)

3.2.4.3 Processing

Server sends you to the selected review.

3.2.4.4 Outputs

Review

3.2.4.5 Constraints

You cannot edit others' review, only rate them.

3.2.4.6 Error/Data Handling

None

3.3 Non-Functional Requirements

3.3.1 Performance

- UI Response time must be lower than 200ms.
- System downtime shall not exceed 5 minute per day.
- RAM usage must be under 5%.
- App shall not run-in background.

3.3.2 Reliability

- Users must be able to preserve their personal information if they want to.

3.3.3 Availability

- App must be available on the app stores.

3.3.4 Security

- Users must not access other user's information.
- Our app should hash user passwords before storing them into database.

3.3.5 Maintainability

- Programmers refactor source code continuously so that the code base will be easy to maintain.

3.3.6 Portability

- App must be cross platform.

3.4 Inverse Requirements

- Users cannot order any food or service by using our app.
- Restaurant owners neither can delete negative reviews nor can add fake reviews on their services deliberately.
- Our app will not provide any service to the restaurant owners like promotion of their service, fake reviews, advertising etc.
- Our app will not collect any user session data (i.e., cookies).

3.5 Design Constraints

Our app is supposed to be usable on most of the smartphones in today's standards. Since our app will not need a high usage of system sources, a very basic phone will be enough.

- This smartphone should have the following minimum hardware requirements: 1 GB RAM, A CPU which has 2.4 ghz clock speed and at least two cores.

- Our app will be cross-platform, so it will be usable on any of the modern smartphone operating systems (i.e., Android, IOS).

- Web server of the app must not use all the system resources on the host computer. Implementation of the server must include optimizations for that.

- There will be no payments of any kind. Also, on UI there will not be any advertisement towards users.

3.6 Logical Database Requirements

- Our app will use several databases. One for general data storage and one for user sessions, cookies etc.

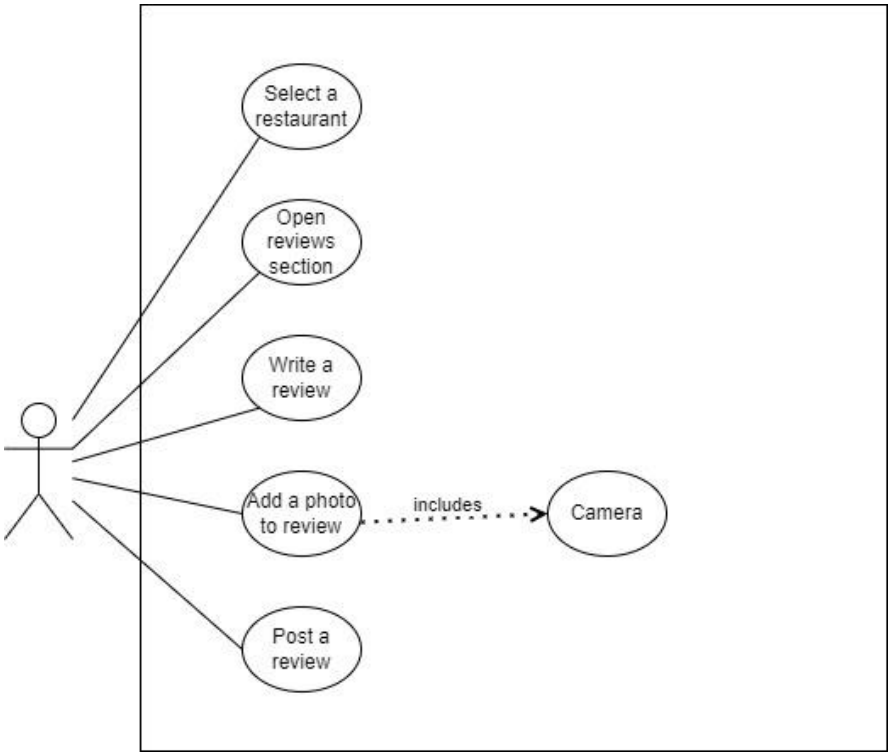
- Our app will also use a file system for binaries such as user photos, restaurant, and food photos. The app and database must correctly handle the file paths for every binary.

- Database should be optimized for high number of queries. Database tables must be in a normalized format and allow admins to access data with easy enough queries.

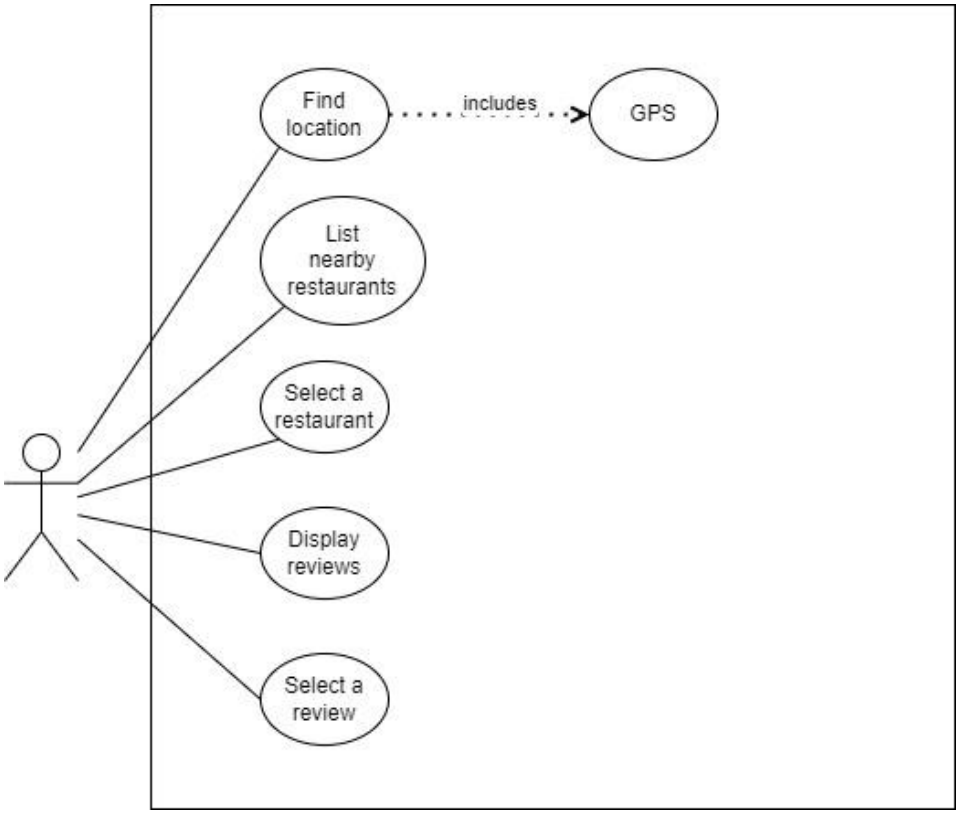
4. UML Diagrams

4.1 Use Cases

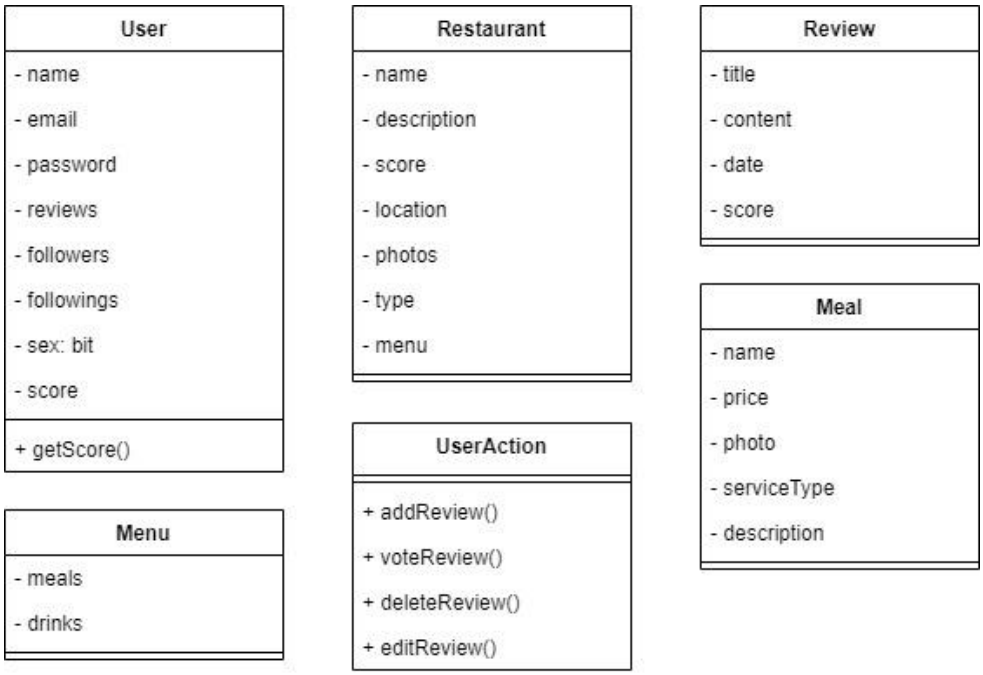
4.1.1 Use Case #1



4.1.2 Use Case #2

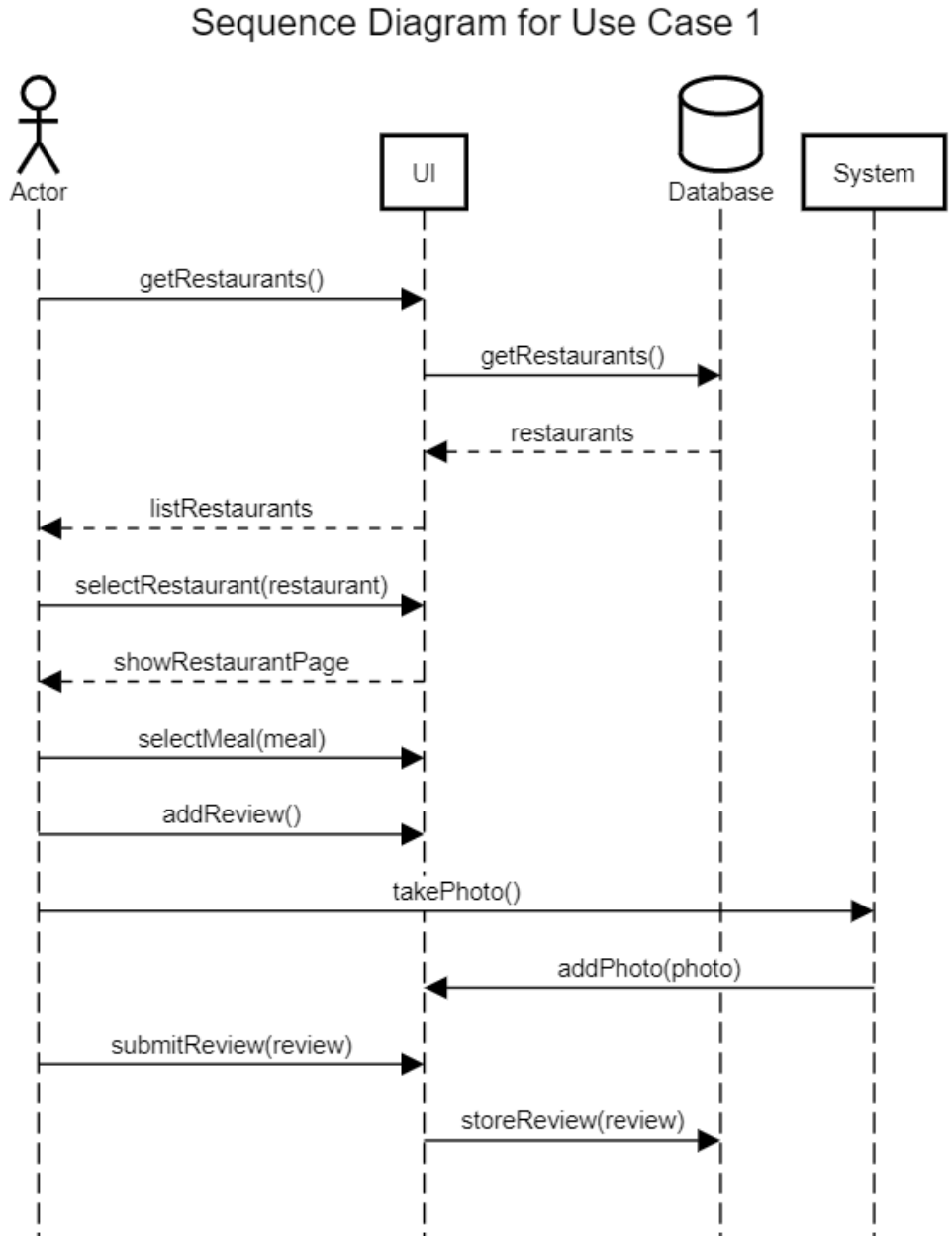


4.2 Classes / Objects

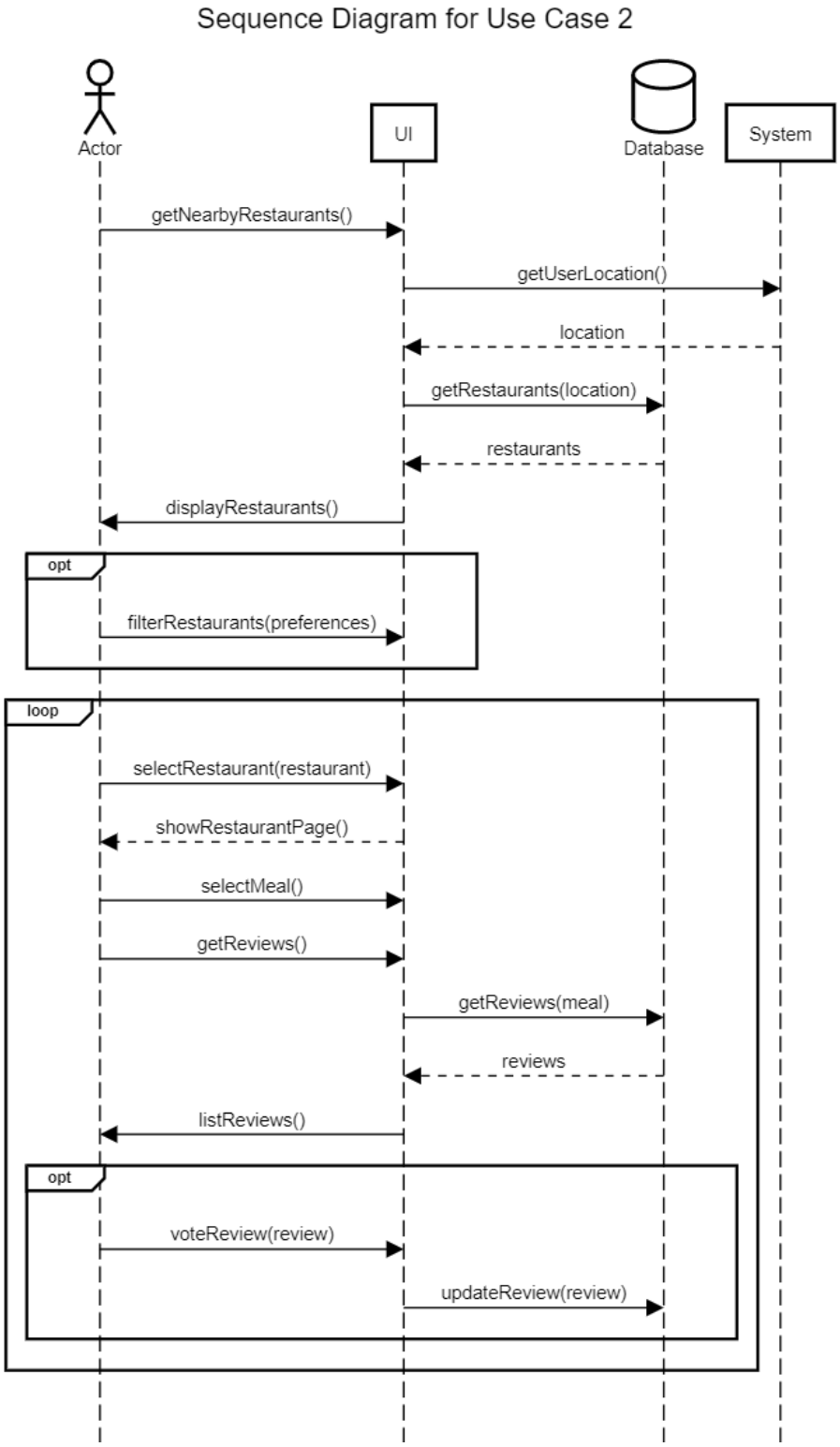


4.3 Sequence Diagrams

4.3.1 Sequence Diagram for Use Case 1



4.3.2 Sequence Diagram for Use Case 2



5. Assignment Table

Section 1 -> Burak Çağlayan
Section 2 -> Erkam Karaca
Section 3.1 -> Bekir Nazmi Görkem
Section 3.2, 3.3 -> Rasim Sadıkoğlu
Section 3.4, 3.5, 3.6 -> Berk Kırtay
Section 4 -> Group Work

Acknowledgment: Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984). The SRS templates of Dr. Orest Pilskalns (WSU, Vancover) and Jack Hagemeister (WSU, Pullman) have also been used as guides in developing this template.