ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 407
Innovative System Design and Development I

Restaurant Reviews According To Geographical Location

Beste Şankaynağı 201711061

Baran Yiğit 201711071

Coşkun Oruç 201811046

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this assignment has been obtained and presented in accordance with academic rules and ethical conduct and the work I am submitting in this document, except where I have indicated, is my own work.

Date: 07.01.2022

Group Members' Name and Surname   **Signatures**

Beste Şankaynağı

Coşkun Oruç

Baran Yiğit

Table of Contents

Table of Figures

ABSTRACT

Most of us have thought about where to go while eating, but we may not be able to decide exactly what we want to choose. Especially if we are going to eat from a place we have not been to before and we have no idea whether this place is good or bad, it can become an inextricable task. We decided to make an application in order to prevent this situation. Today, people spend an average of 248 minutes a day looking at their mobile devices and they aim to reach the information they want to learn quickly and easily. After knowing this information we decided to make a mobile application. This project sorts the restaurants around the users according to their scores and distance from the user, and shows other comments made to the chosen restaurant by the other users. At the same time, it uses a captcha-style for Sign in to prevent bots from commenting.

ÖZET

Çoğumuz yemek yerken nereye gideceğimizi düşündük ama tam olarak ne seçmek istediğimize karar veremeyebiliriz. Özellikle daha önce gitmediğimiz bir yerden yemek yiyeceksek ve bu yerin iyi mi kötü mü olduğu hakkında hiçbir fikrimiz yoksa bu içinden çıkılmaz bir iş haline gelebilir. Bu durumun önüne geçmek için bir uygulama yapmaya karar verdik. Günümüzde insanlar günde ortalama 248 dakikalarını mobil cihazlarına bakarak geçirmekte ve öğrenmek istedikleri bilgiye hızlı ve kolay bir şekilde ulaşmayı hedeflemektedir. Bu bilgileri öğrendikten sonra bir mobil uygulama yapmaya karar verdik. Bu proje, restoranları kullanıcıların puanlarına ve kullanıcıya olan uzaklıklarına göre sıralar ve seçilen restorana diğer kullanıcılar tarafından yapılan diğer yorumları gösterir. Aynı zamanda, botların yorum yapmasını önlemek için Oturum aç için bir captcha stili kullanır.

1.INTRODUCTION

The Restaurant Reviews according to Geographical Location project report consists of Introduction, Literature Review, Software Requirement Specification, Software Design Description, Discussion, Conclusion and References sections. Details on the content of the project are available in our report.

As a result, it can be said that a literature review for the system was made and the methods to be used were decided according to this research. After the literature review, the software requirements specifications document is prepared. This document provides information about the functionality and performance criteria of the systems. The final document is the software design disclosure document, which contains information about the architecture of the system as well as the UI design information. This report covers all the theoretical aspects of recommendation systems and the next step in the project is the implementation of a system that covers the theoretical knowledge prepared in this document.

2.LITERATURE REVIEW

2.1 INTRODUCTION

Thanks to this project, users will be able to learn how the restaurant food is, whether they comply with the hygiene rules, without going to certain restaurants, they will be able to post pictures to support what they say, and they will be able to decide whether the comments made by others are useful or useless. The number of active internet users has increased by 1036% in the last 20 years (413 million to 4.66 billion)[6,7] and according to the documents published by Science, false stories and documents are shared faster than the truth [8]. Since we estimate how much a user who uses a commenting site or application can trust this information in an environment where misinformation spreads so quickly and comfortably, we will first ask a commenter to share his/her comment with a picture that supports his/her comment. Since we think that it will not be completely reliable, we will authorize users to evaluate other users' comments (as useful and useless). Comments with a certain amount of useless evaluations higher than useful will be automatically deleted by the system. Likewise, there may be business owners who want to increase their restaurant's score, so they can log in to comment. Required and to login a small captcha approval is required. While thinking about this project, we will add an option to show restaurants on the map, considering that there may be people who have just come to the city (There will be a certain score with the distance to the user and the points given by other users. Restaurants will appear on the map in different colors according to this score (from green to red).) so that the user will be able to read the reviews of the restaurant that is more attractive to him and choose whether he wants to go there.) Likewise, if the user knows what he wants to eat but does not know which restaurant he wants to go to, we can filter certain types of restaurants by searching with certain keywords. We will chart a path.

2.2 RELATED WORKS

Mobile apps are the easiest way to order food. It has been observed that there is an exponential growth in online food ordering due to the rapid increase in mobile users. Consumers find it very easy to search websites or apps to select dishes from restaurants they like. Reviews and ratings given by consumers on websites are helpful in making purchasing decisions for new customers. Filters are also added to websites and mobile apps to categorize and customize the order according to the consumer's needs. Everyone is a stakeholder in business. However, the most important stakeholder is the customer. Delivery time is the most important and decisive factor in customer retention. Company employees also know that delaying an order means consumers are more likely to switch to other food ordering and delivery services. Food delivery companies understand the importance of delivery time very well, so they provide live tracking order facility to know the arrival time of their orders. Order tracking is completely dependent on GPS (global positioning system), meaning the delivery agent needs to enable GPS service on mobile or vehicle so that

customer and consumer can track the cargo on mobile phone. Relationships are the key to success in the service industry, so companies focus more on building strong relationships with customers. In food delivery services, the company does not take ownership of the food flavor. This responsibility is taken by the restaurant owners. The main concern of the meal delivery service is to provide the food to the door of the consumer and this within the promised time frame (William R. King, Jun He, 2006) [4].

The development of technology in the field of information and communication, especially the Internet, facilitates consumers' access to kitchen information, and plays an important role in enabling kitchen entrepreneurs to market their products and develop a wider market. The development of technology brings along the need for mobility, especially the high use of smartphones. The rapid increase in smartphone usage has led to two billion active smartphone users worldwide in 2016, with Indonesia one of the countries with the third largest growth in smartphones, according to the Emarketer report. Indonesia is also expected to be the fourth largest smartphone user, with nearly 100 million active smartphone users in 2018. Culinary developments in Indonesia are accompanied by advances in mobile device technology and increased the mobility of consumer activities, thus eliminating the need for locations and restaurants. Information emerges quickly and easily. This study aims to develop applications that can help consumers find information on mobile devices using Location Based Service (LBS) technology. Current technology also makes it easy for programmers to make this application, one of which is Location Based Service (LBS). Location-Based Service (LBS), is a service accessible via mobile devices that displays information connected to a network and can indicate the geographic location of the device or the location of a place like to find the nearest restaurant. Location Based Service (LBS) will facilitate users to search for remote areas and also what information is available at user-acquired location (Layona & Yulianto, 2016) [1].

Online food ordering has been adopted by most of the restaurants that offer food delivery. Customers using online food ordering showed gratitude to the technology and stated that online ordering met expectations. The advantages of ordering online are improved order certainty, high throughput, and improved customer relationship management. These will likely eliminate costs and operational threats for most restaurants. The survey found that when a consumer decides to purchase food online, it is influenced by multiple factors. The main key factors identified are time savings and convenience. People compare prices on the online meal delivery website and apps, and then review all product feedback and ratings before making their final choice of dish. Therefore, restaurants should make appropriate strategies to increase consumer confidence by receiving feedback, to encourage customers to share reviews about their food, and also to raise awareness of their online food market presence by displaying food products online (Girish Deore1 & Pranav Shete, 2016) [2].

Online food delivery services in urban areas rely heavily on urban transport due to the rather heavy traffic in cities. These services use user-generated content to encourage collaborative consumption among their members. The researcher evaluated the impact of traffic conditions (using the Google Maps API), key performance indicators of online food ordering and delivery services. From the overall research, it was found that although early deliveries showed a quality issue association with the number of comments made by customers after receiving orders at the door, traffic conditions had no practical effect on throughput and delivery time fulfillment (Juan C. Correa, 2017) [3].

If online food ordering services offer a delivery option, the customer will be more satisfied and a confirmation email will be sent to the customer regarding the order status. In the current scenario, every online food ordering and delivery service company has a mobile app in various app stores, and almost many people carry smartphones in urban areas. When an order is placed through the mobile application, the customer can track their orders via the mobile application, thanks to the GPS installed on each smartphone. The app also shows the estimated time of delivery (Shantashree Das, Debomalya Ghose, 2019) [5].

In the article named A Mobile Location-based Information Recommendation System Based on GPS and WEB2.0 Services they used GPS systems to draw a map around user and show restaurants around them in map we will also use this but we will make restaurants clickable that would show user reviews for that restaurant and show their points, also according to article using Tagging(Tagging means binding a keyword that is not a part of coding with information within computer[11]) would help user to reach what is he looking for faster so after looking at this article we decided to add a little bit of tagging to our data.they also used coloring to change tag colors according to least used and most used tags we used this in our project's map function which shows closer restaurants with good score in green and far or bad scored ones with red[10].

Yemeksepeti is actually a food ordering app but in Yemeksepeti you could also look at comments made for any restaurant, but in Yemeksepeti you can't add images to food you ordered or you can't vote any comment done for any restaurant, unlike in Yemeksepeti in our project we could upvote, downvote other reviews add images of food and you could even have road map to the destination [9].

2.3. Background

2.3.1. Location-Based Services (LBS)
LBS is the term that describes the technology used to locate the device. LBS is basically finding users Location by taking signal from users telephone catching that signal from satellites and resending that location in mathematical location to user. After finding the user's location, use that location to give services. LBS are composed of two parts; first part is Location Manager second part is Location Providers [12,13] .

2.3.2. Location Manager (API Maps)
Location Manager is something like a middleman. It takes data from Location Providers and helps the user to process that data like getting mathematical values turning them into maps or changing maps or changing the view of the user like satellite views etc. This package is located at com.google.android.maps.

2.3.3. Location Providers (API Locations)
By detecting displacement, users can determine their location, track movement, and determine proximity to a particular location. Currently, it is built from new information and communication technologies (New Information and Communication Technologies / NICTS), mobile telecommunications systems and handheld devices, data from the Internet and Geographic Information Systems (GIS) using spatial databases. Location Based Services has five key components, including:

Mobile Devices: A tool used by the user to request the required information. Information can be provided in the form of sound, images, and text.

Communication Network: A communication network is needed to first encapsulate the users data and the information that is requested and send that to the service provider. Service providers would send the requested information to the user by user data. Communication Network could be one of the three; Cellular network(GSM, CDMA), Wireless Wide Area Network(WWAN), Wireless Local Area Network.

Positioning Component: When a service is running it needs the user's current position.

Service and Application Provider: Service and Application providers present their services to users and wait for user feedback to improve their works.

Data and Content Provider: As Service providers are not responsible for saving all the required data. Data and content providers are needed to request data.

2.3.4. Captcha

Captcha is a test that stops automated sign in process. Captcha is a test that gives a text in what is wanted is written and different photos and ask user to choose which photos are related to text. For average human it takes 10 sec[14] to solve a captcha as it is easy for a human brain to read a text and create an image that corresponds to that text but for a computer it firstly has to know what the text means than check for images that corresponds to that and find images closer to the image the computer finds so it is harder for computer than human.[15]

2.4. Conclusion

The results that can be obtained from the application of a geographic information system to use the location search of a restaurant in the province of Ankara with an Android-based local-based service method are:

The closest restaurant using a smartphone as this media application.

This system can make it easier for users to find the nearest restaurant, so it is easier for users to determine which restaurant to visit.

The generated Google Maps road data is more complete to support a more diverse route selection.

There are a lot of Review systems and GPS systems but there are not many applications that combine those two systems to create a new one. In our system we hope to create an application that would help users to find new places to go when they are hungry, help them locate hidden nice restaurants in hard to find places. Users don't even need to go to that location to know if that place is good or bad as other reviewers would score that place already. User's could also get achievements for scoring a restaurant that doesn't have any

reviews or scores. In conclusion, we hope to create a Restaurant reviewing application that would help users to find new locations to eat food and enjoy.

3. Software Requirements Specification

3.1. Introduction

"Restaurant Review App according to Geographic Location" is a project that helps users find restaurants based on their needs. The main goal of the project is to provide users with an easy-to-use application that helps them find restaurants according to their taste and convenience. Users can also give feedback about restaurants by rating and writing reviews. The Google Maps API shows the restaurants in the current range. The application also makes it easy for the user to send invitations to friends/colleagues for events.

3.1.1. Purpose

The purpose of this document is to give a detailed description of the requirements for the "Restaurant Review App according to Geographic Location" project. It will indicate the purpose and complete statement for the development of the system. It will also explain system constraints, interface and interactions with other external applications. Through this app, users can share their experience of visiting restaurants and provide reviews of foods reach the restaurants they want to go to. Basically, this app will help build a social community for food lovers.

3.1.2. Scope of Project

"Restaurant Review App according to Geographic Location" is a GPS-based mobile application that helps users find the closest restaurants based on the user's current location and other features such as top rated restaurants, restaurant type, and more.
Restaurant owners can provide restaurant information using this application. This information will form the basis for the search results displayed to the user. An administrator also uses this application to manage the system and keep information accurate. For example, the manager can verify reviews posted in the restaurant's comments section.
This information will form the basis for the search results displayed to the user. An administrator also uses this application to manage the system and keep information accurate. The administrator can, for example, update restaurants and manage user information.
Also, the software needs both an Internet and GPS connection to receive and display results.
 All system information is kept in a database located on a web server. The software also interacts with the GPS Navigator on the user's mobile phone. Using the GPS Navigator, users can view the restaurants they want on the map and have access to a real-time map that will help them access the location of the restaurants. The app is also capable of presenting both summary and detailed information about restaurants.
Thanks to this navigator, users can easily find the closest and most perfect place to eat according to their needs and needs. This app will definitely boost restaurant business. On the other hand, this app guides food lovers to meet their restaurant requirements. This app will be a common platform for users and restaurant owners to learn the latest details about restaurants.

3.1.3. Glossary (Definitions, Abbreviations, Acronyms)

Android Studio: A mobile coding environment for creating a mobile application for Google Play.

Real Time Maps App (Application): Mobile application software which is prepared to work on mobile devices.

API (Application Programming Interface): API helps two different software to use each other's functions.

GPS (Global Positioning System): GPS, or the Global Positioning System, is a global navigation satellite system that provides location, velocity and time synchronization.

SRS: Software Requirements Specification

2D: 2-Dimensional

UI: User Interface

IDE: Integrated Development Environment

UC: Use Case

DB: Database

IEEE: Institute of Electrical and Electronics Engineers

3.1.4. Overview of Document

This SRS document is generated by using the IEEE STD 830-1998. Also, the rest of the document contains five important parts which are Introduction, Overall Description, Specific Requirements and References.

Introduction part describes an overview of the whole SRS. Overall description part contains general information that is not too specific and is provided as a background for the following sections. Specific Requirements contains more detail and presents the requirements with many diagrams that illustrate the functional requirements of the project.

3.2. Overall Description

3.2.1. Product Perspective

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

3.2.1.1. Development Methodology

This system will consist of two parts: one mobile application and one web server. The mobile application will be used to find restaurants and view information about them while the web server will be used for managing the information about the restaurants and the system as a whole. The mobile application will need to communicate to a GPS application within the mobile phone, which in turn communicates with a physical GPS device to find the location of the user. The GPS will provide the mobile application with locations of both the user and the restaurants and the distance between them, but it will also provide maps and the functionality to display the application's data on the map. The functionality provided by the GPS will be embedded into the application in order for the user to be able to use the functions in the application in a seamlessly manner. Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. Both the mobile application and web server will communicate with the database, however in slightly different ways. The mobile application will only use the database to get data while the web server will also add and modify data. All of the database communication will go over the Internet.

3.2.2. Product Functions

Getting user position: After receiving permission from the user we receive their coordinates from their mobile phone.

Getting desired position: After receiving a position name from the user, get the desired location's mathematical coordinates from the database and send that to the receiver.

Creating a Map: After getting position and desired position create a 2D map using road map creator(Open for change) to wanted position using the most ideal way.

Sending a mail: If a user gets banned or one or more of the user's reviews are getting deleted send a notification to the user(also used for signing up).

Update Restaurants/Reviews: As we don't want our information of restaurants or reviews to be old we constantly update our reviews or restaurant information.

CheckImage(Optional): Check if the image entered is a food or food related or other image, if not food ignore image, else receive image.

Authentication Services: Making sure user is not bot and User has entered from legal ways.

3.2.3. User Characteristics

In particular, the users who use this project are the people who have difficulty in deciding what to eat and where to go, and they need options for this. Users should have this information for functions such as search, GPS information and filtering. They should also be aware of the use of mobile phones.

3.2.4. Constraints

The mobile application is constrained by the system interface to the GPS navigation system within the mobile phone. Since there are multiple system and multiple GPS manufacturers, the interface will most likely not be the same for every one of them. Also, there may be a difference between what navigation features each of them provide. The Internet connection

is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function. Both the web server and the mobile application will be constrained by the capacity of the database. Since the database is shared between both application it may be forced to queue incoming requests and therefore increase the time it takes to fetch data.

3.2.5. Assumptions and Dependencies

3.2.5.1. Assumptions

-User expected to know how Google Maps is used.

-User should have bare minimum knowledge regarding smart phone usage.

-Mail services is expected to work.

-Google services should be running.

-We hope users upload images relevant to their comments.

3.2.5.2. Dependencies

-User should give access to GPS services.

-User should have stable internet connections.

-User's device should support Android services.

-This application is not made for specified users so it doesn't have text-to voice services so User is expected to have the capacity to see.

3.3. Specific Requirements

3.3.1. External Interface Requirements

3.3.1.1. User Interfaces

 User Interface must be in a clear and user-friendly structure. UI should provide the users consistency of users' information. Also, possible errors should be minimized because of the situation of target user group. If user meets any error during using the system, error messages should help the user in positive, actionable and professional approach. Also, error messages should be appropriate for user knowledge and situation. The user interface should be successfully designed not to encounter any difficulties.
 The target user group in the project is the people who are undecided about where to go and what to eat and need options. So the UI filtering option should be good and the options should be well listed.

3.3.1.2. Hardware Interfaces

The project is a mobile based application. Client side application will be developed as an Android application. To use this system, User should have an Android based operating system. In addition, a camera may be needed for this project in order to comment on

restaurants by adding photos. There will also be a server. And on this server, information such as restaurant type, score, location and user information will be stored here.

3.3.1.3. Software Interfaces

To develop the Android application, Eclipse will be used as a client-side IDE. Also, Java is the main programming language to develop Android applications in Eclipse. Android Studio will be used and Google libraries will be used.

In addition, in order to develop this project, Google's own database, especially the Google API, will be provided by Başarsoft, and information will be gained about roads and a map library. This library will help us to define successful and minimum error.

3.3.2. Use Cases

3.3.2.1. Opening Page
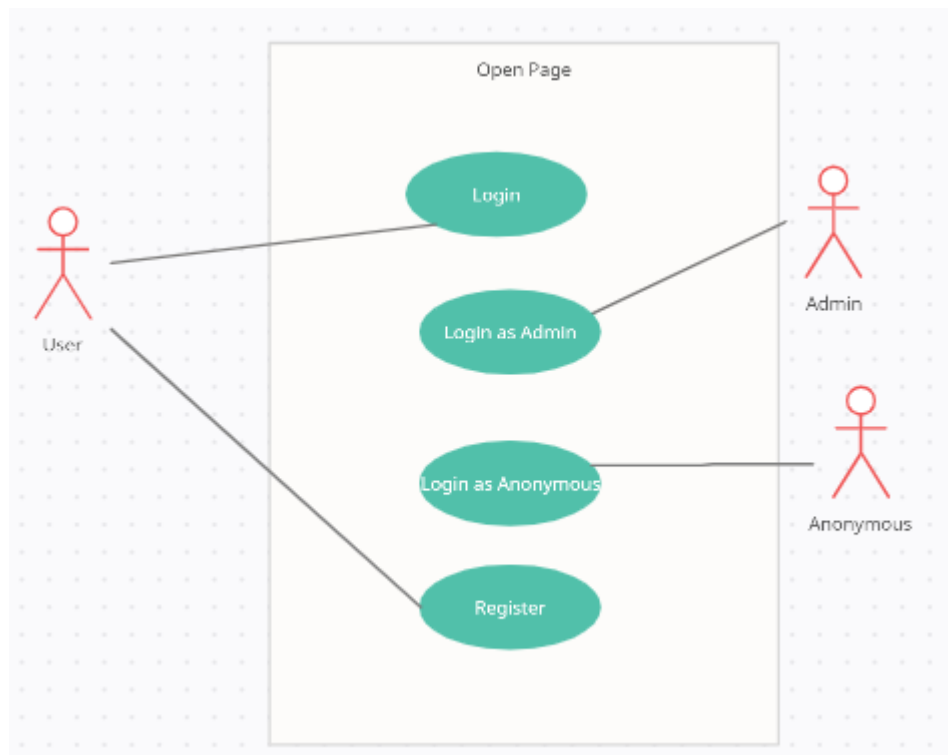
Use Case: Opening page
Register
Login as Admin
Login as Anonymous
login

Use Case Diagram:

Use Case Name: Opening Page

Use Case Number: UC-01

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: User, Admin, Anonymous

Overview:

This use case is the page with the options to login to the application.

References: [16], [17], [18]

Related Use Cases: -

Typical Flow Description:

      Pre-Condition:
      1. Internet connection must be available.
      Post-Condition:
      1. The user is logged into the system.
      Typical Flow Description:
      1. The user clicks the "Register" button to register to the application, or the "Login", "
Login as Admin", " Login as Anonymous" button if he / she is registered.
      2. The user is transferred to the "Register" or "Login" screen.
      Alternative Flow Description:
      1. If the person entering the application is "Admin", he / she clicks the
      "Login as Admin" button.
      2. If the person entering the application is " Anonymous ", he / she clicks the
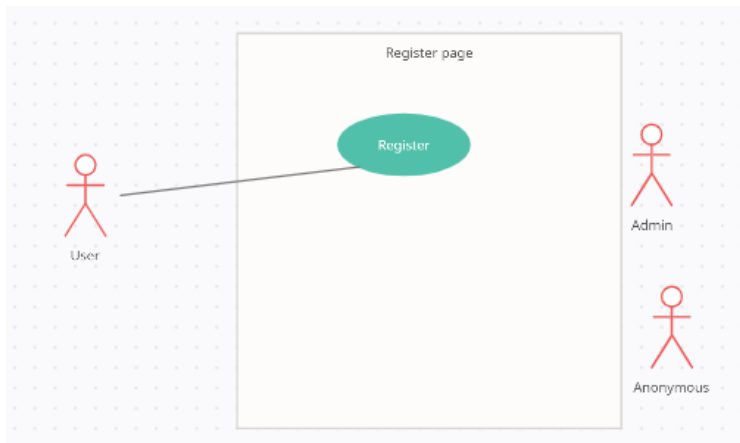   "Login as Anonymous " button.

3.3.2.2. Registration Page

Use Case Name: Registration Page

Use Case: Register Page

Register

Use Case Diagram:

Use Case Number: UC-02

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: User

Overview:

This use case allows users to register with the system.

References: [16], [17], [18]

Related Use Cases: UC-01

Typical Flow Description:

      Pre-conditions:
      1. User must switch from Open Page to Register Page.
      2. The user must not be registered to the system before.
      Post-conditions:
      1. As a result of the correct actions, it is registered in the system.
      Typical Flow Description:
      1. The user defines "User Name", "E-mail" and "Password" for himself/herself, fills in the fields and clicks the register button.
      2. The system checks whether the Username and E-mail are already registered.
      3. If there is an error condition, it is indicated on the screen. Re-enters user information.

3.3.2.3. Member Login Page

Use Case Diagram:
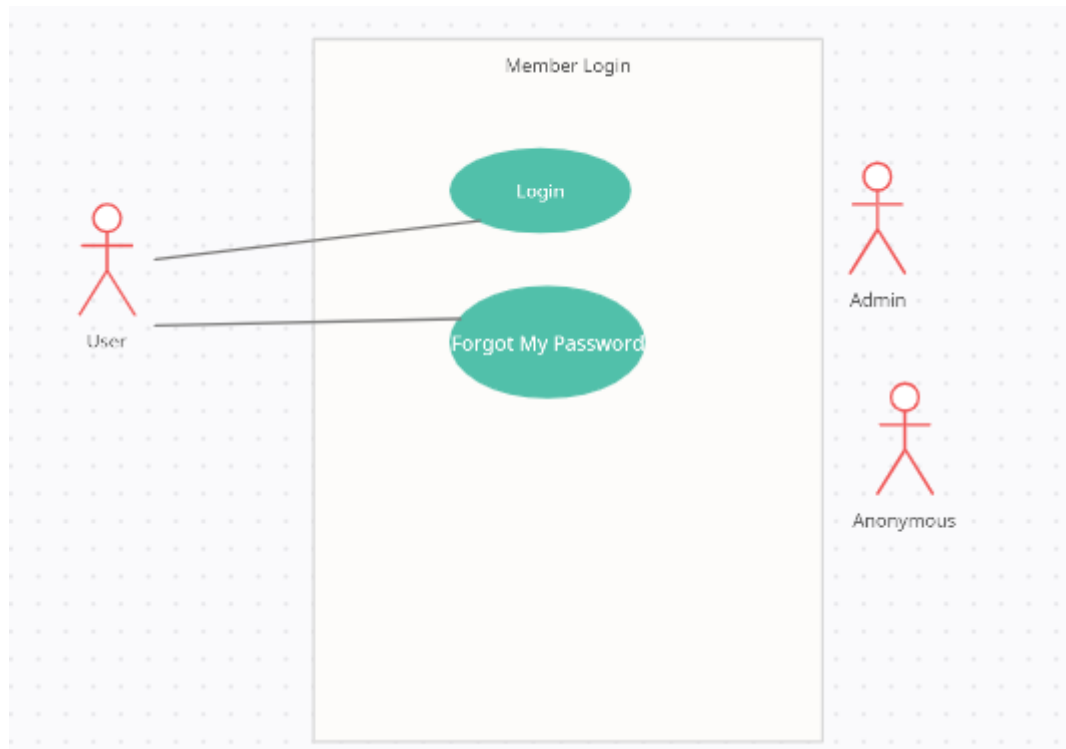
Use Case: Member Login Page
Login
Forgot My Password

Use Case Name: Member Login Page

Use Case Number: UC-03

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: User

Overview:

This use case allows users to log into the system.

References: [16], [17], [18]

Related Use Cases: UC-01

Typical Flow Description:

    Pre-Conditions:
    1. User must switch from Open Page to Login Page.
    2. It must enter the Username and Password correctly.
    Post-Conditions:

1. As a result of the correct operations, it has entered the system.
Typical Flow Description:
1. User clicks the "Login" button by filling in the "Username" and "Password" information.
2. The system checks the User Name-Password match.
3. If there is an error condition, it is indicated on the screen. Re-enters user information.
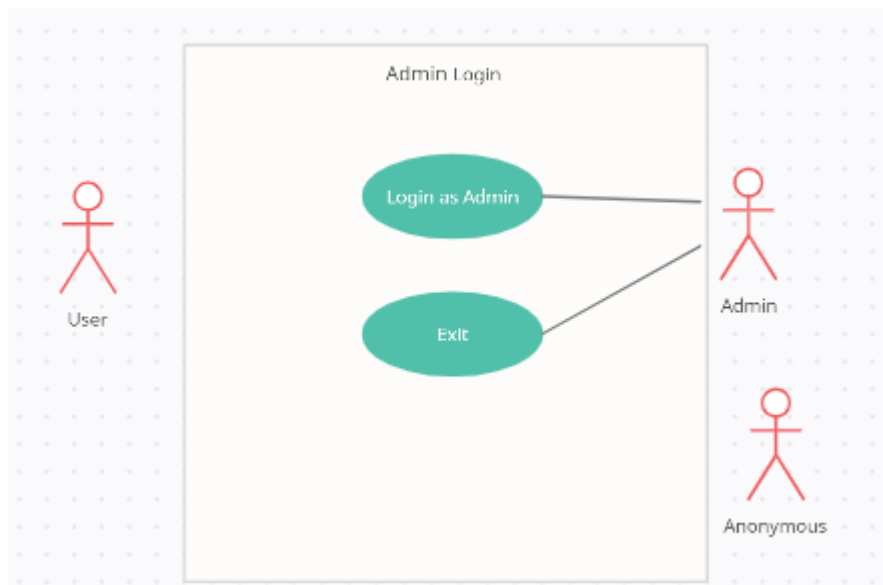Alternative Flow Description:
1. If the user has forgotten his password, click the "Forgot My Password" button.
2. The system sends a password reset mail to the e-mail address.
3. The user can reset his password in line with the incoming mail.

3.3.2.4. Admin Login

Use Case Diagram:

Use case name: Admin Login

Use case number: UC-04

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Users, System

Overview:

The role of an admin is important to manage the mobile application maintainability after the deployment. The potential actions that the restaurant admin
can perform. This is a super user role that can control or edit the roles of all the other actors in the system. The admin can add, delete or edit the menu items. The admin can edit the users and their roles in the system.

References: [16], [17], [18]

Related use cases: UC-01

Typical Flow Description:

     Precondition:
       1. User must have an Internet connection.
    Postconditions:
       1. User's login type is set to Admin.
    2. Accepted functions are set.
       Exceptional Situations:
    1. Admin forget his/her password.
    Successful Scenario:
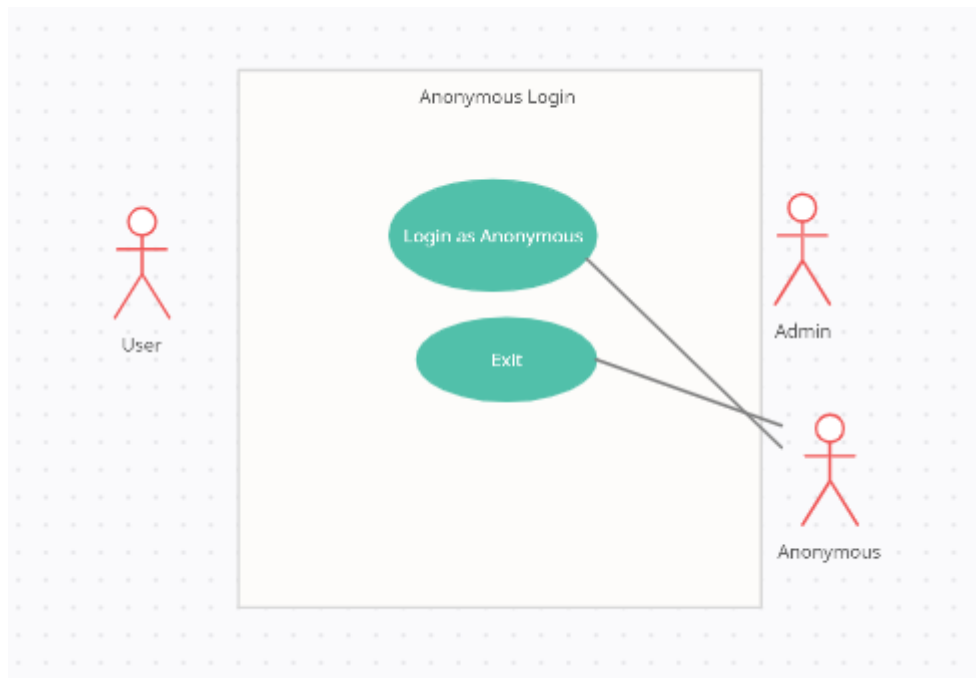   1) Admin enters id, password.
       2) System sets user's login type as Admin.
       3) Sets acceptable operations.
    Alternatives:
    1) Admin enters Reboot code.

3.3.2.5. Anonymous Login

Use Case Diagram:

Use case name: Anonymous Login

Use case number: UC-05

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Anonymous User, System

Overview:

When users do not want to waste their time to logging in and just search restaurants around them or look for particular restaurant's reviews. Calls login setter for system, sets value of user to anonymous.

References: [16], [17], [18]

Related use cases: UC-01

Typical Flow Description:

Preconditions:
-
Postconditions:
1. User's login type is set to Anonymous.
2. Accepted functions are set.
Exceptional Situation:
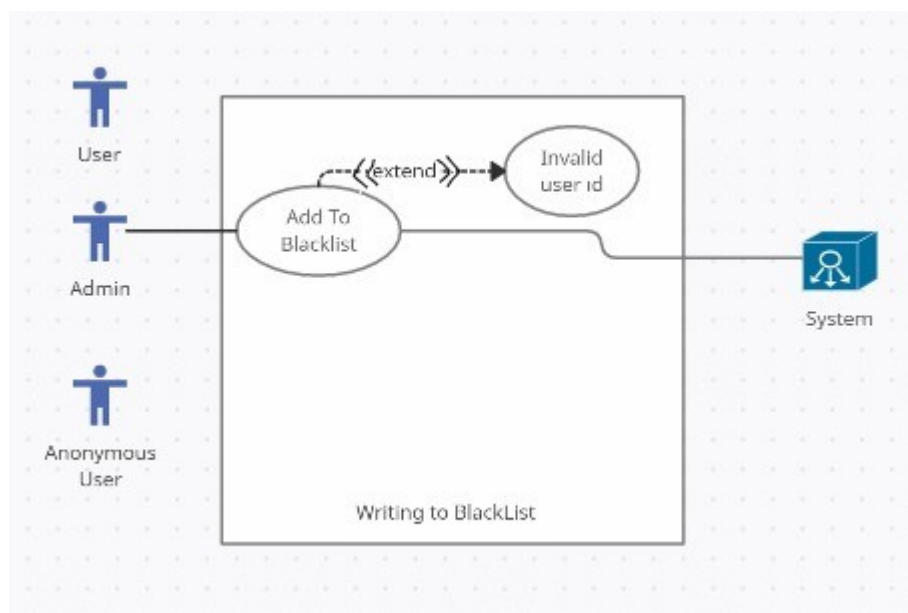1. User pressed around the button but not to the button.
Successful Scenario:
1) Presses the anonymous login.
2) System sets user's login type as Anonymous.
3) Sets acceptable operations.

3.3.2.6. Adding to the blacklist

Use Case Diagram:

Use case name: Adding to the blacklist

Use case number: UC-06

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Users, Admin, System, Mail System

Overview:

When a particular User constantly writes wrong reviews that deletes or adds inappropriate images below food images they are added to blacklist, so they can't re-enter this app.

References: [16], [17], [18]

Related use cases: -

Typical Flow Description:

      Pre-Conditions:
      1. Student DB should be accessible.
      2. Admin must select to add blacklist.
      3. There should be a user with given id.
      Post-Conditions:
      1. A mail is sent to the user that says she/he is banned.
      2. Users data is added to Blacklist Db.
      3. User's id is deleted from User database.
      4. User's reviews is not changed.
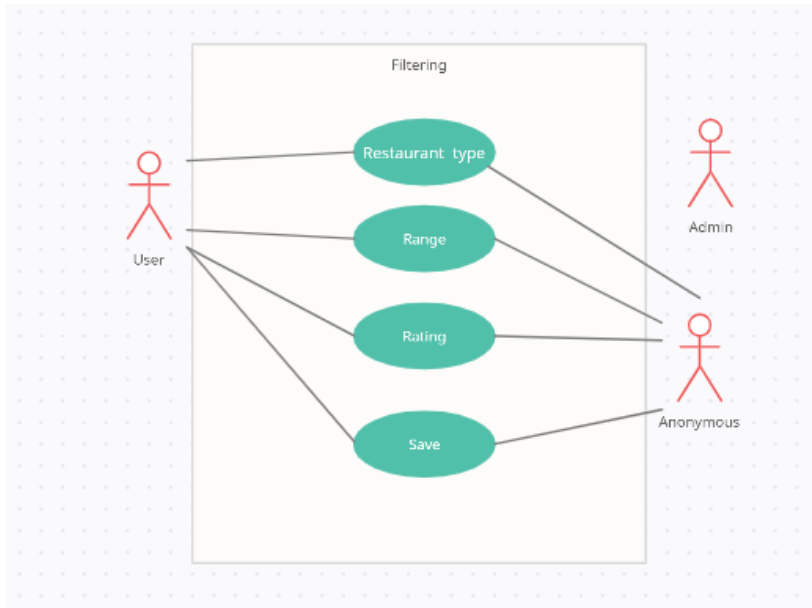   Exceptional Situation:
      1. Admin banned wrong person.
      Successful Scenario:
      1) Admin press send to blacklist.
      2) System adds User data to blacklist.
      3) System delete users data from User's database.
      4) System sends mail to the User that his/her account is suspended.

3.3.2.7. Filtering

Use Case Diagram:

Use Case:
Restaurant type
Range
Rating
Save

Use Case Name: Filtering

Usage Case Number: UC-07

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Anonymous, User

Overview: This use case allows users and Admin to filter by distance and meal options when choosing a restaurant.

References: [16], [17], [18]

Related use cases:

Typical Flow Description:

      Pre-Conditions:
1. The user must be logged in.
2. The user must decide what he wants to eat.
3. The user must decide the distance he wants to go.
Post-Conditions:
1. The restaurants that are suitable for the user's selection are listed for the correct actions.
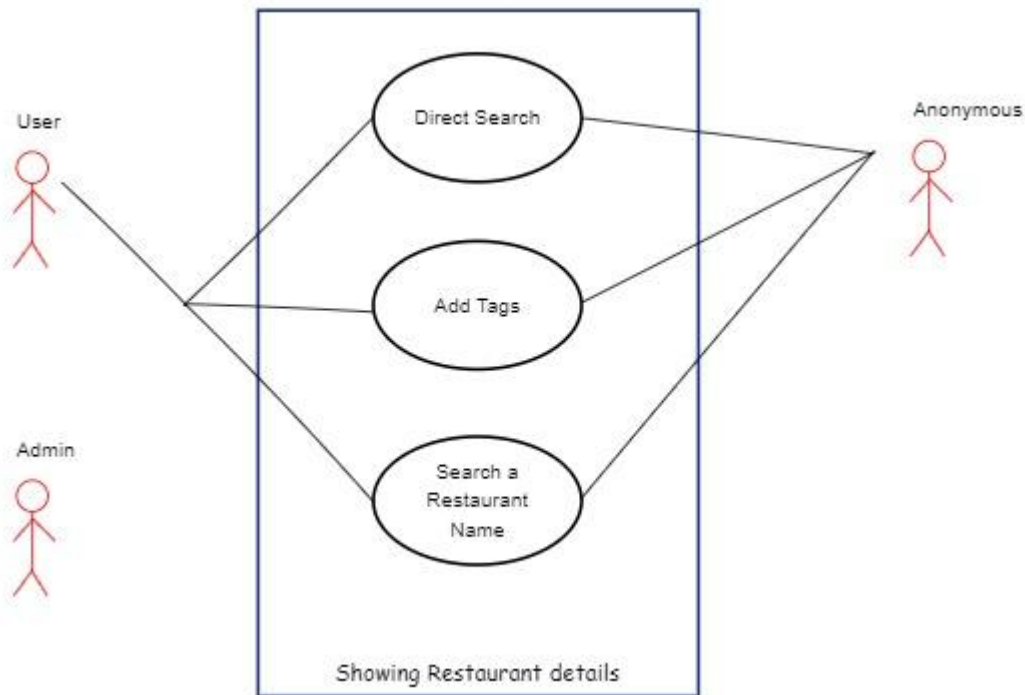   Typical Flow Description:

2. The user chooses the type of food they want from the food options.
3. The user selects the range.
4. The user clicks the save button.
Alternate Stream Description:
1. If there is no restaurant in the desired type or range, it will receive an error message.

3.3.2.8. Showing Restaurant Details

Use Case Diagram:

Showing Restaurant details

Use Case Name: Showing Restaurant Details

Use case number: UC-08

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: User, Anonymous, System, Google Maps

Overview:

When the show button is pressed Restaurant names, average points, location difference between user and restaurant would show if clicked show other reviews according to different user's reviews to that particular restaurant.

References: [16], [17], [18]

Related use cases: -

Typical Flow Description:

Pre-Conditions:
1. Show button should be pressed.
2. User should have a network connection.
Post-Conditions:
1. Restaurant information would be shown.
2. A cross would be in top right that would go back.
Exceptional Situations:
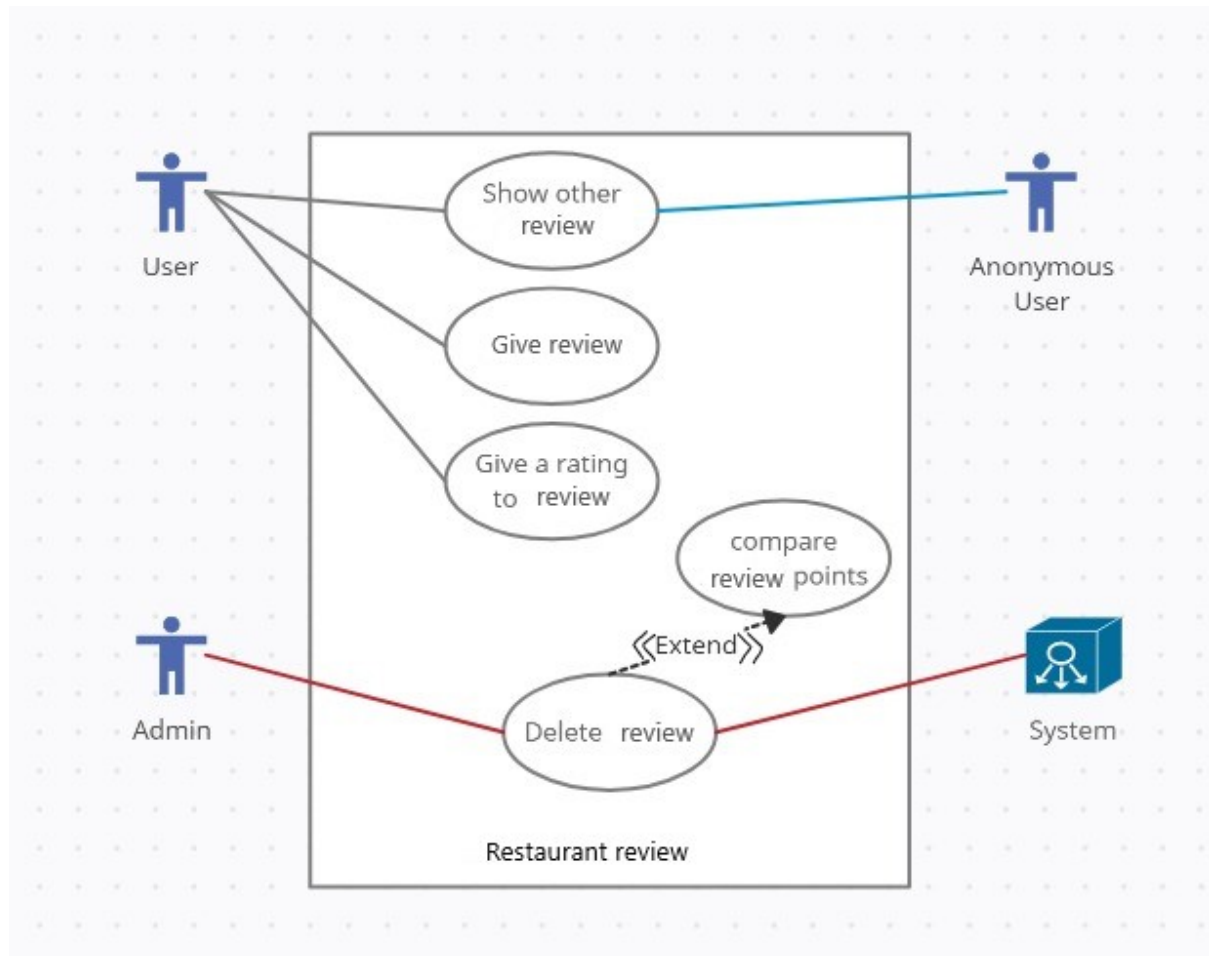-
Successful Scenario:
1) User clicks show.
2) User's current coordinate is sent to System.
3) Restaurant N meters close to the user is selected.
4) Restaurant details are added.
5) Restaurant points are added.
6) Restaurant are listed according to their location.
7) Restaurants are shown in search.

3.3.2.9. Reviewing Restaurants

Use Case Diagram:

Use case name: Reviewing Restaurants

Use case number: UC-09

Author: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Users, System, Anonymous

Overview:

When users are choosing the restaurants that they would like to eat they would like to make sure that the restaurant is good so reviewing restaurants uc
is used to give users an easier way to make sure that restaurant is good.

References: [16], [17], [18]

Related use cases: UC-10, UC-11, UC-12

Typical Flow Description:

      Pre-Conditions:

1. There should be restaurant details in the database.
2. User should have internet connection.
3. UC1, UCBa, UCBe should be accessible.
Post-Conditions:
1. If UC-11 or UC-12 is called, the Restaurant reviews page is refreshed.
2. If UC-10 is called check if any of the review is deleted refresh the page.
Exceptional Situations:
-
Successful Scenario:
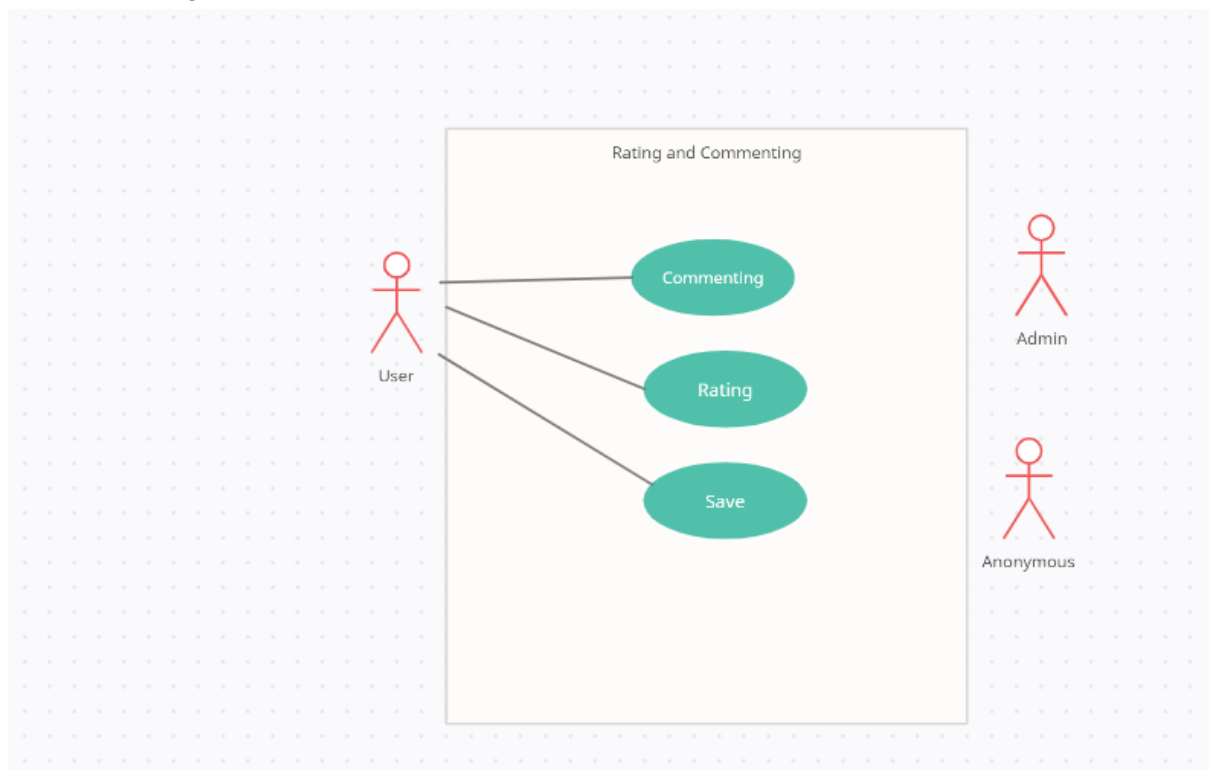1) Check for Any Use Case call.
2) If UC-11 is called change the restaurant reviews page.
3) If UC-12 is called refresh Restaurant page.
4) If UC-10 is called, the restaurant reviews page is refreshed.

3.3.2.10. Commenting And Rating

Use Case Diagram:

Use case name: Commenting And Rating

Use case number: UC-10

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Users, System

Overview:

The user can write a review and see all the reviews of a selected restaurant from these screens. The review includes the rating and an opinion about the restaurant. This use case also allows users to rate restaurants.

References: [16], [17], [18]

Related use cases:-

Typical Flow Description:

Preconditions:
- User should be logged in.
- User should have Internet connection.
- User should give access to GPS services.
- User's should open the application in Restaurant location.
Post-Conditions:
- Check location service.
If was in location
- Update database.
- Refresh is called.
- Exit activity.
If wasn't in location
- Send error message.
- Exit activity.
Exceptional Situations:
Successful Scenario:
1) User clicks give review.
2) System checks if User had been in location for past 2 hours.
If was in location:
3a) Get user review message.
4a) Send Review to Restaurant DB.
5a) Call refresh use case.
6a) Give Comment is given message.
7a) The user clicks on the rating button of the relevant restaurant.
8a) Gives points from 1-5.
If wasn't in location
3b) Stop review.
4b) Give error message.
5b) Exit page.
Alternate Stream Description:
If the 1st user has not been to this restaurant, he/she will receive an error message and cannot give points.

3.3.2.11. Deleting Reviews

Use case name: Deleting reviews

Use case number: UC-11

Author: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Users, System

Overview:

When other users find particular review useful for what they are searching for they would give it a positive point, or if it is fake, irrelevant to the food or not useful to the user's they could give it a negative point, when a review gets lot genitive points than positive it automatically gets removed by the system.

References: [16], [17], [18]

Related use cases: UC-10

Typical Flow Description:

      Pre-Conditions:
      1. Restaurant should be in database.
      2. User should have internet connection.
      Post-Conditions:
      1. Review's points should be changed.
      2. If negative is higher review should be deleted.
      Exceptional Situations:
      1. At the same time 2 reviews is given.
      Successful Scenario:
      1) User would press up or down.
      2) The system will check the difference between positive and negative points.
      3) If there is a difference Then Delete the review.
      4) If there is no difference Then Let the review be there.

3.3.2.12. GPS Services

Use case name: GPS services

Use case number: UC-12

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Users, Anonymous, System, Google Maps

Overview:

When the user clicks the scan button this use case is called this use case is the parent of UC4 and UC3.

References: [16], [17], [18]

Related use cases: UC12, UC13

Typical Flow Description:

Pre-Conditions:
1. User should click scan.
2. User should have a network connection.
3. Users should accept GPS services.
4. Google Maps should be accessible.
Post-Conditions:
1. User exits the application.
2. User reaches the destination, UC6 is called.
Exceptional Situations:
-
Successful Scenario:
1) User clicks Scan.
2) UC 4 is called in which user asked to select which restaurant does user would like to go.
3) After a restaurant is selected and accept is selected UC 3 is called which draws a map and reads users coordinates.
4a) If User has reached their destination User will be asked to review that restaurant.
4b) If User closes the app before the user reaches their destination nothing would happen.
5a) UC6 is called.

3.3.2.13. Show Restaurants

Use Case Diagram:
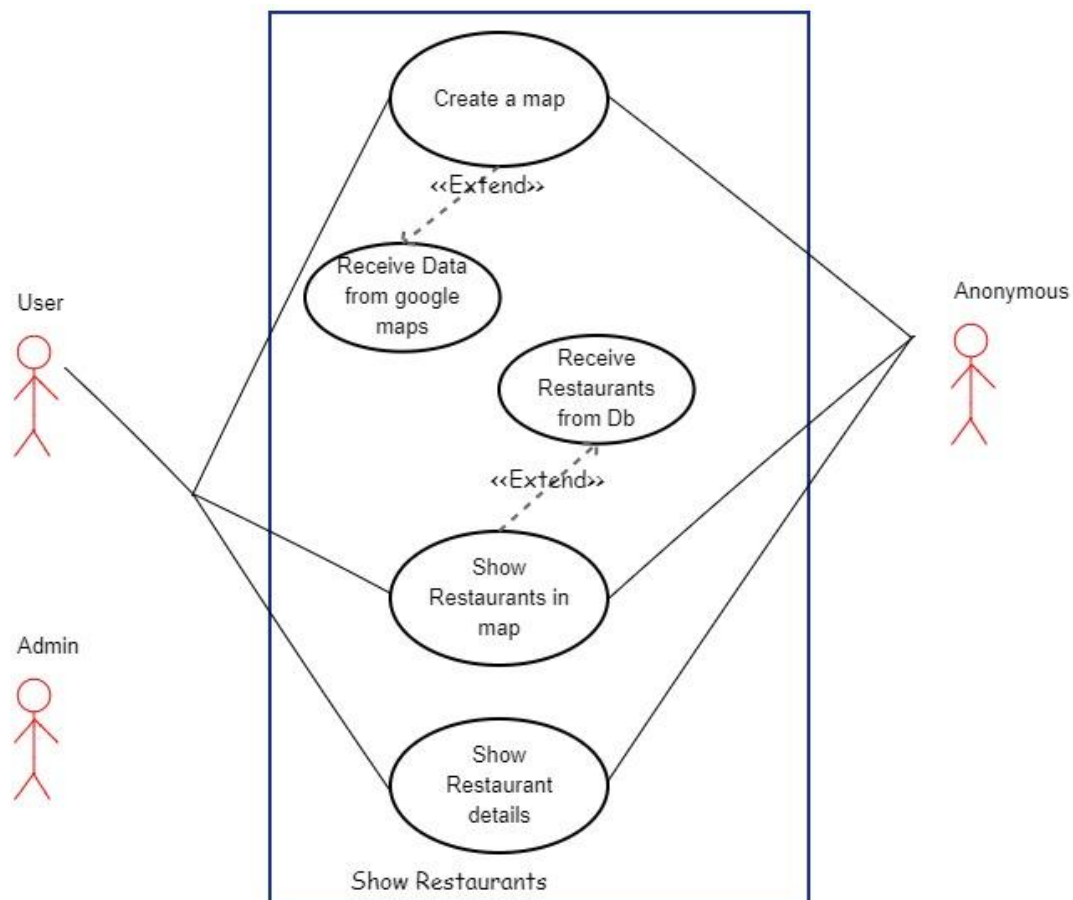
Use case name: Show restaurants

Use case number: UC-13

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: Users, System, Google Maps

Overview:

When user would like to choose a restaurant around him/her, he/she opens the map every restaurant around the user with different colors according to average points that restaurant has, would have a button that would send user to that location.

References: [16], [17], [18]

Related use cases: UC-14

Typical Flow Description:

    Pre-Conditions:
    1. User should have internet connection.

2. User should give permission to GPS services.

3. Restaurant coordinates should be in DB.

4. Restaurants should have reviews.

Post-Conditions:

1. Restaurants are listed.

2. Restaurants are printed according to their points average.

Exceptional Situation:

1. Restaurant is closed.

Successful Scenario:

1) User selects scan.

2) Other restaurants around user would have shown according to coordinates in coordinate select.

3) System checks their points according to their point average give them a color tag.

4) According to their color tag show restaurant in different colors.

5) Map is done user would choose.

3.3.2.14. Send To Restaurants

Use Case Diagram:
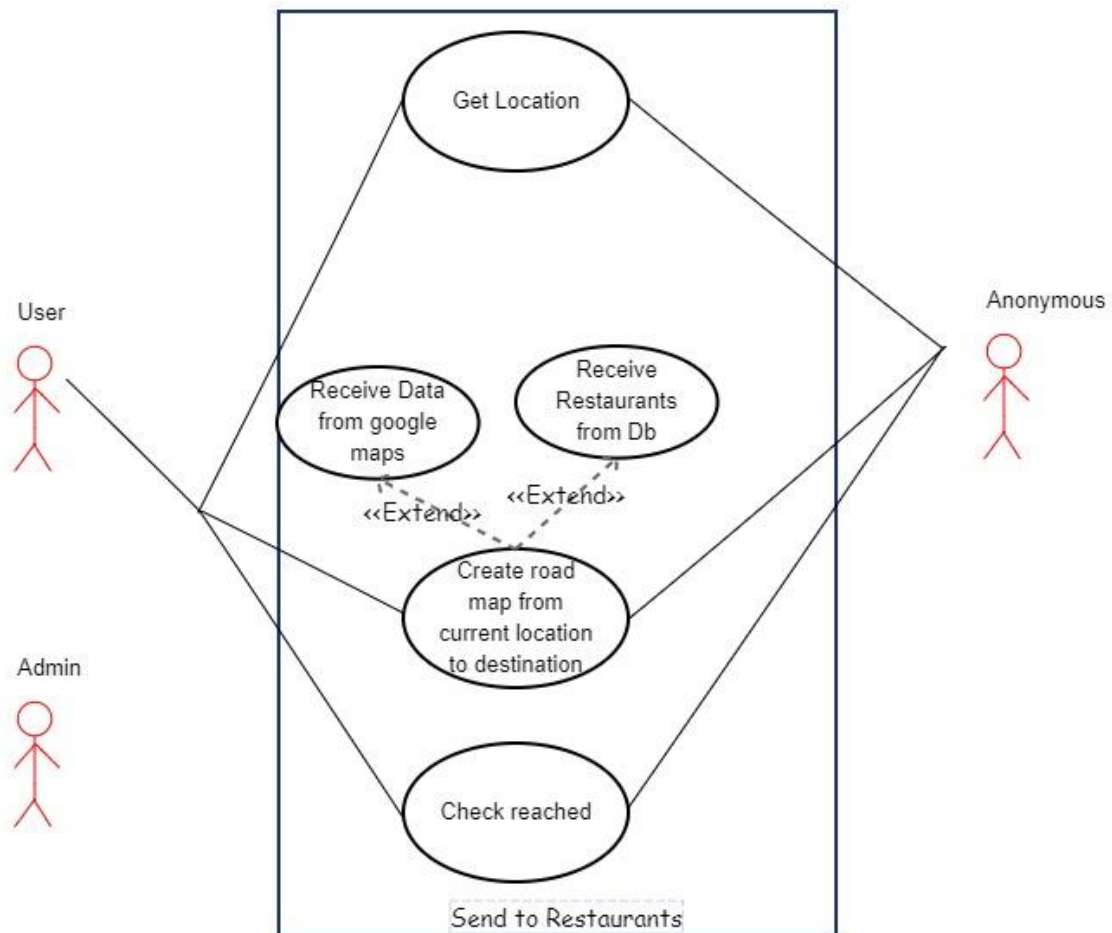
Use case name: Send to restaurants

Use case number: UC-14

Authors: Beste Şankaynağı, Baran Yiğit, Coşkun Oruç

Actors: User, Google Maps, System

Overview:

When User would choose the restaurant that he/she would like to go he/she press the button and the location would open in Google Maps with given coordinates.

References: [16], [17], [18]

Related use cases: -

Typical Flow Description:

        Pre-Conditions:
        1. User should click send me.
        2. User should have network connection.
        3. User should accept GPS services.
        Post-Conditions:
        1. Google Maps will open.
        2. A map is drawn according to location.
        Exceptional Situations:
        1. Restaurant is Closed.
        Successful Scenario:
        1) Users clicks take me there.
        2) User's location is packaged with restaurant location.
        3) Package is sent to Google Maps.
        4) Created the shortest route to the restaurant.
        5) A map is drawn as an app.

3.3.3. Performance Requirements

For the best performance user's should have updated Android services in their phone and Network connection to have stable and up to date info regarding restaurants.

3.3.4. Security Requirements

For security, we are holding our user's data in web database so there won't be any data of any user in another user's mobile device. Also, we are going to add encryption to our database when uploading data so even when someone manages to get user's data in their hands they could not read or change that data as they want.

3.3.5. Design Constraint

The design shall be according to IEEE Standard 1016-1998 Recommended Practice for Software.

3.3.6. Software System Attributes

3.3.6.1. Portability

We are creating a mobile application as around 97.2% of Turkey's population are using mobile phones [19]. After looking at this we could say that around 97% of Turkeys population might use our project on their phone so our project will be pretty portable.

3.3.6.2. Usability
This application is created for those who are not so close with mobile phones so this app will be pretty easy to use and user-friendly.

3.3.6.3. Reliability
First of all we are making sure no bots are entered to this application. Also, we are making sure users are logged in to make reviews for any restaurant so our program will be quite reliable also for our user's comfort. We are also going to use store user data in separate database which only admin can reach that database.

3.3.6.4. Availability
This mobile application is going to be available on Google Play Store so anyone who has Google Play in their phone, and has network connection can use this application.

3.3.6.5. Security
For security, we are going to put user data in separate database. Also, we could use encryption for changing their data in case of data stealing thieves can't reach users' data in a short time.

4. Software Design Description

4.1. Introduction
Software Design Description, is a document to detail implementation and design of the project in question and detailing it with visuals.

4.1.1. Purpose
This software design document provides information about the EATIE (Restaurant-Reviews-According-To-Geographical-Location) architecture and system design. The target audience of this document is customers who cannot decide what to eat and where to go. For better understanding, this SDD includes various diagrams of the project such as UML diagram, activity diagram and block diagram.

4.1.2. Overview
In the Introduction section, information was given about the purpose and design of our product. In the Approach section, the approach we will use while designing the product was highlighted. In the System Design section, the current class diagram, sequence diagram and flowcharts of the product were designed.

4.1.3. Definitions and Acronyms. Abbreviations
SDD: Software design document is a written description of the software product. Software design document is a document written and given by the software designer to guide the software developer team through the architecture of the software project.
UML: UML (Unified Modeling Language) is an emerging language for modeling business systems. Although it is generally used in the software industry, it is used by those who want to explain business systems, a process or any job with graphics.
DFD: Data flow diagram is a form of diagram that is closer to programming. The level system is used and as the level systems starting from zero increase, the detail is specified.

User: Any user character that is yet to register to the system.
Admin: The user character that supervised the system and a registered administration user.
Anonymous: A user who can use the system in a limited way without being a member of the system.
Database: The database where the registered users of the system are stored with their personal information.
Class Diagram: A modeling language that is used in the software development process.
IEEE: Institute of Electrics and Electronics Engineering
UML: Unified Modeling Language

4.2. Approach

The heart of the whole system is the database. Currently, the system is only considered for term project. For large restaurants, performance considerations should be taken into account in terms of Hardware/Software capacity, page load time etc. In addition, vulnerabilities should be evaluated for large-scale systems. In the future this may also be available as a Mobile app and integrated in App stores.

There is a module that provides functionality for the Administrator only. It will not be available to any other users. Using a graphical interface, it will allow an Administrator to manage the menu that is displayed to users. Also, Administrator can add & delete restaurants and update restaurant information.

There are two types of users who can use this application. One is User and the other is Anonymous User. User can be able to create his/her new profile with its real information. If a user wants to rate a restaurant, he/she must have to log in with his own authentication. User can create his/her ID and make dynamic ratings of any restaurant.

A Google Maps map loaded with the Google Maps API. A list of restaurants on the right side of the page that are within the area displayed on the map. The Google Maps map will focus immediately on the user's location. It shows restaurants on the map based on their GPS coordinates. Restaurants that are currently visible on the map will be displayed in list form. You will see the average reviews of each restaurant (ranging from 1 to 5 stars). When you click on a restaurant, the list of reviews will be shown. Also, it shows the Google Street View photo via the corresponding API! The map will be updated in real-time to show the relevant restaurants. Our visitors also want to share their opinions about restaurants! We will add a review part for an existing restaurant. For the moment, there are not many restaurants or reviews. Fortunately, Google Places offers an API to retrieve restaurants and reviews! We'll use the search API to find restaurants in a particular display area.

Briefly,

Admin can view/upload/replace/delete User, Anonymous User and Restaurant information.
Admin can view all the feedbacks published by the customers.
The User can log in to the system after submitting the registration information.
The User can search for restaurants based entirely by rating and restaurant type.
The User can can post descriptions about a particular food place.

For now, the language used to build this application is Java and on the client side, and the Oracle database on the back-end.
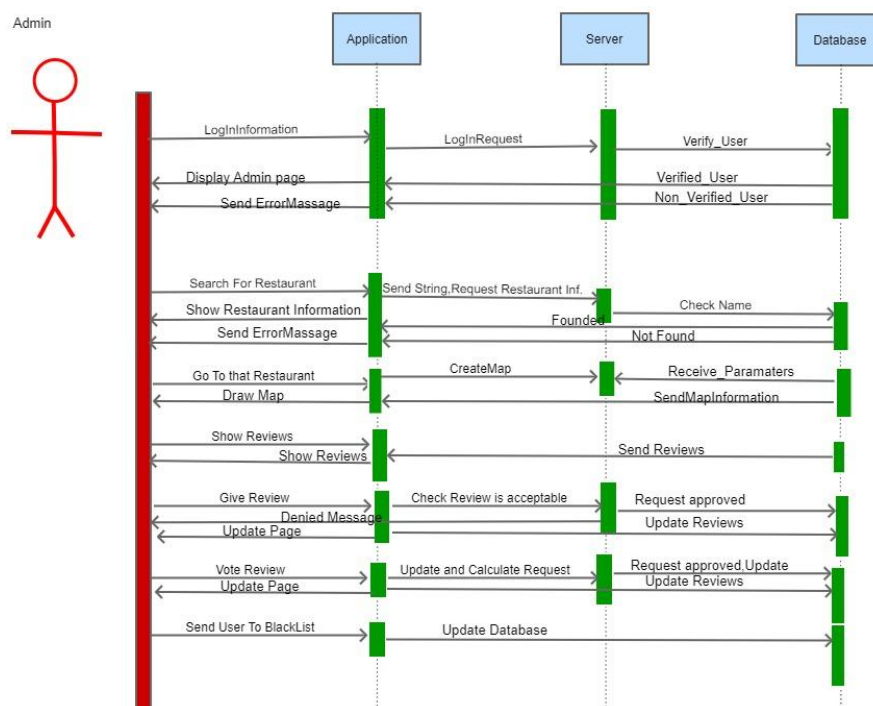
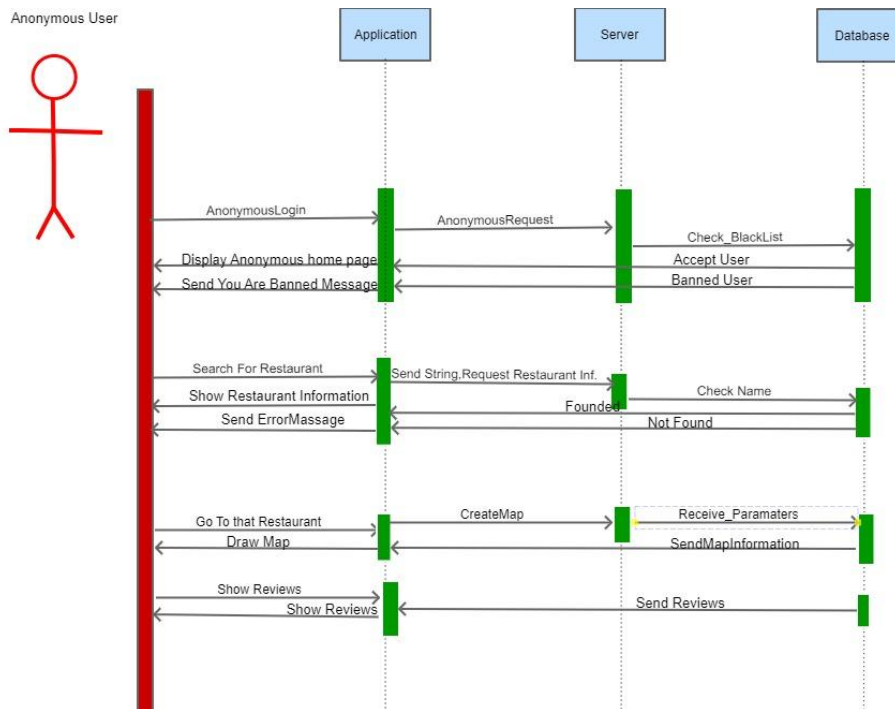## 4.3. System Design

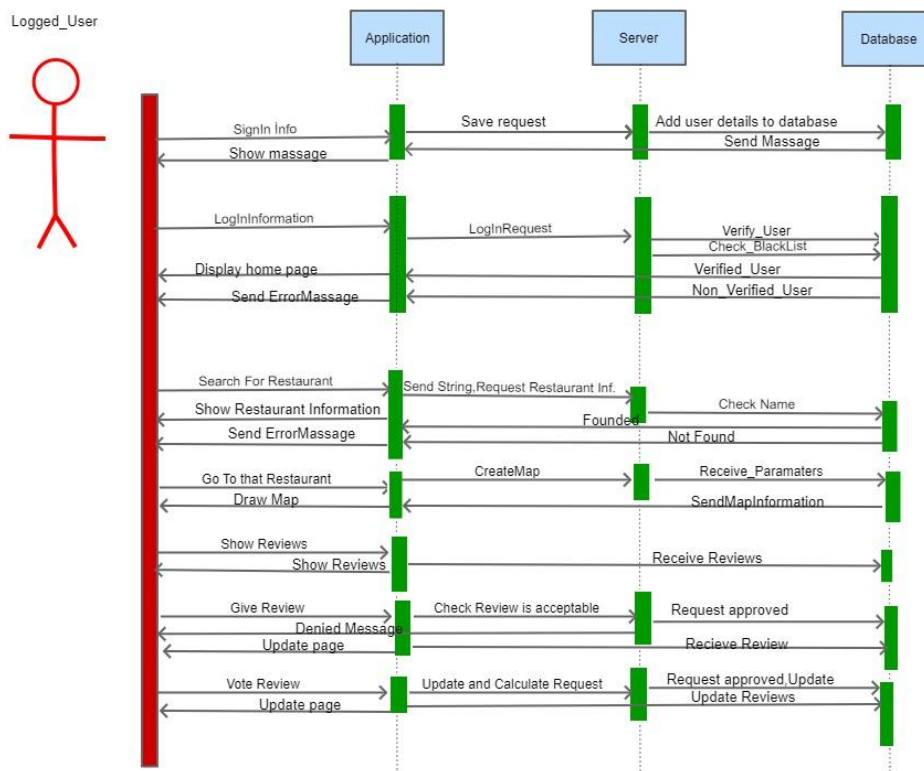### 4.3.1. Class Diagram [24]
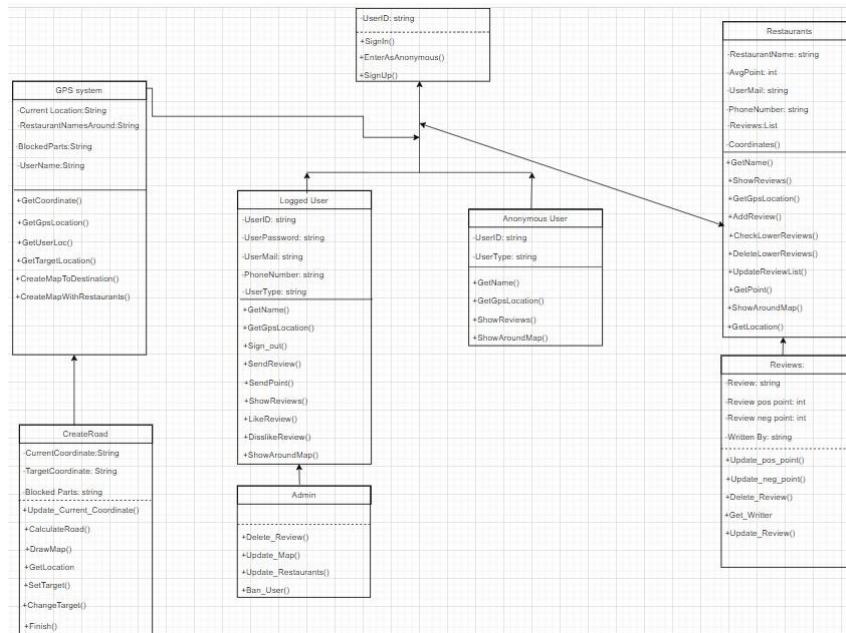
4.3.2. Sequence Diagrams[23]

4.3.2.1. Admin
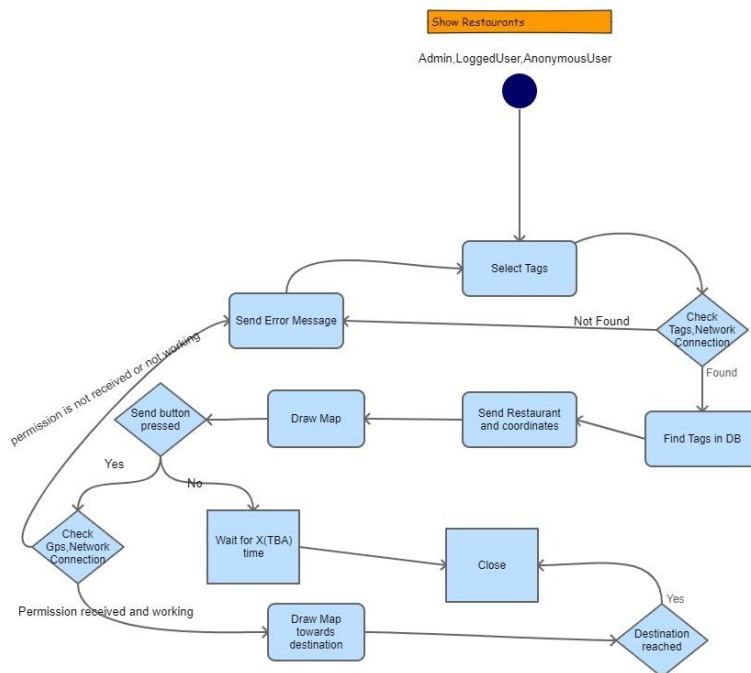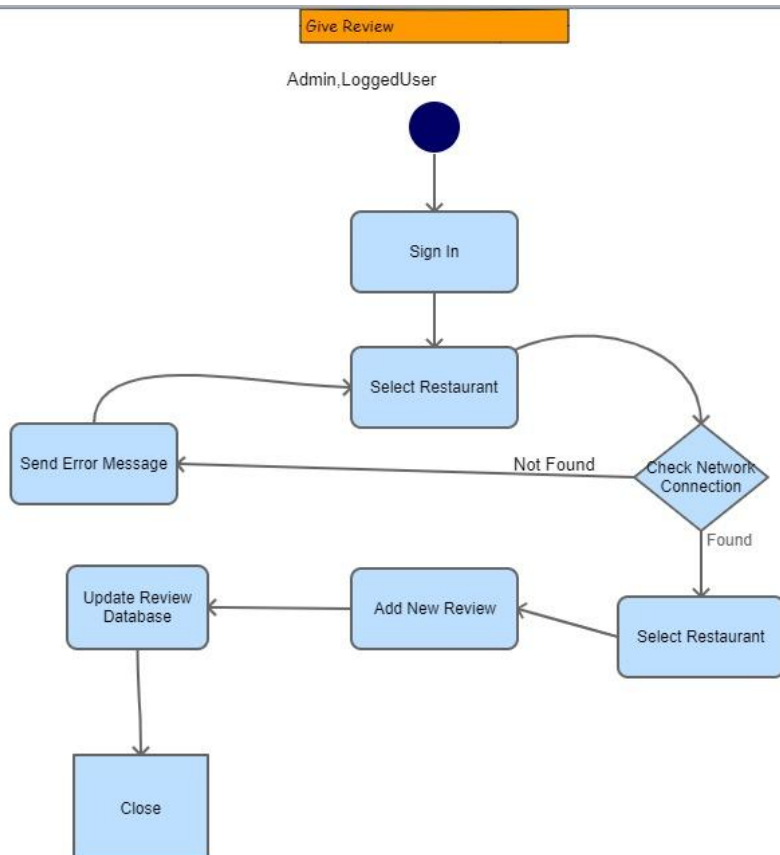
4.3.2.2. Anonymous

4.3.2.3. Logged User
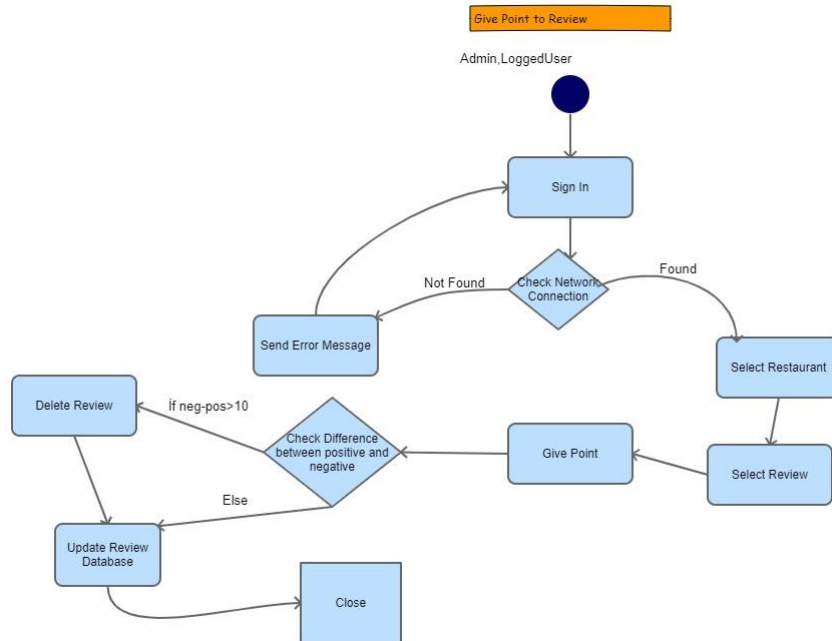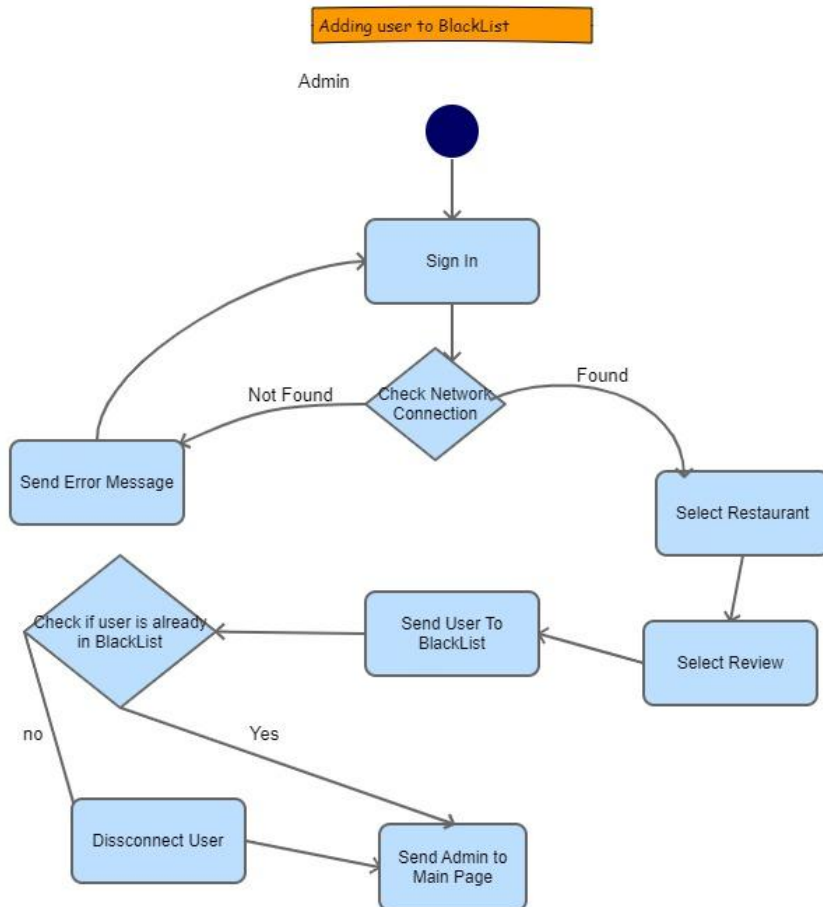
4.3.3. Flowcharts[23]
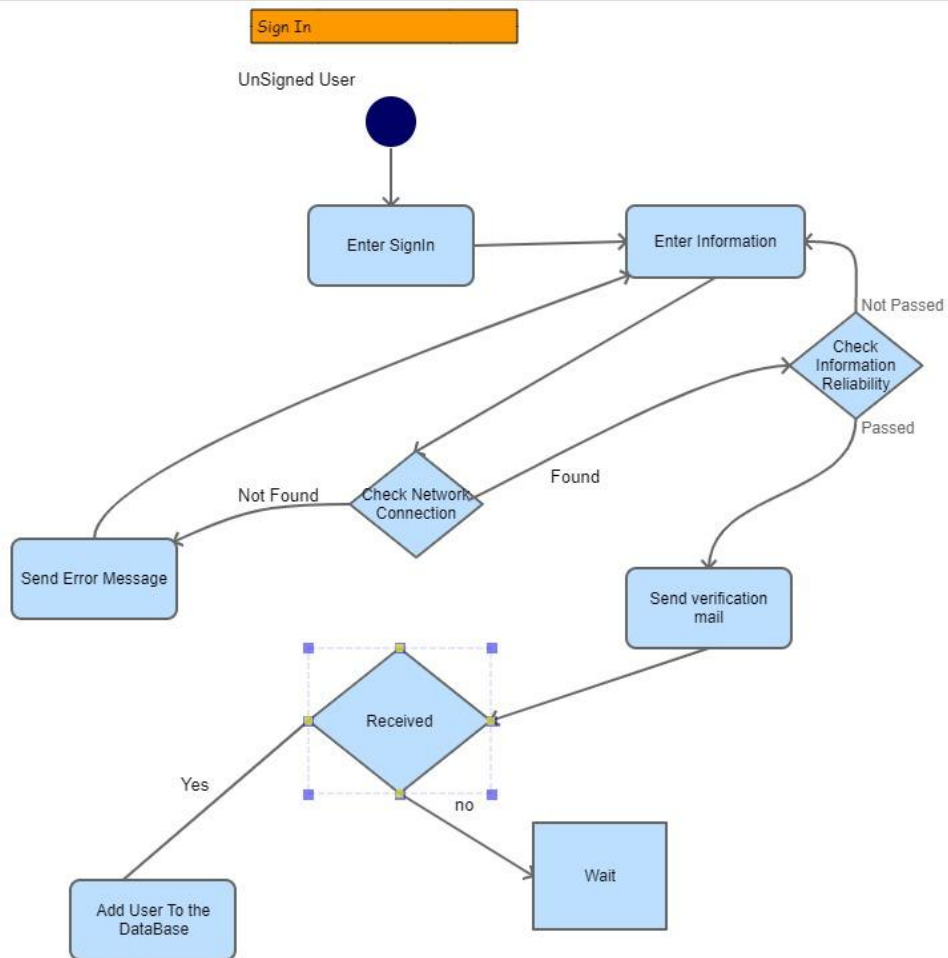
4.3.3.1. ShowRestaurants

4.3.3.2. GiveReview

4.3.3.3. GivePointsToReviews

4.3.3.4. AddingToBlackList

4.3.3.5. Sign-in

4.3.3.6. Database Diagram[25]
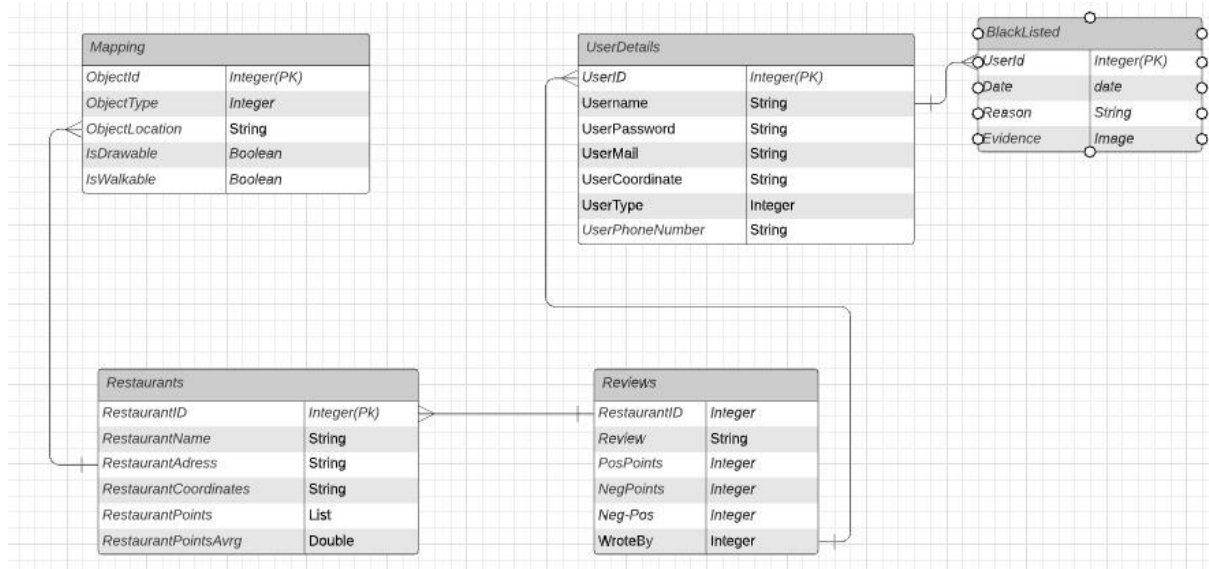
4.3.4. UI'S[23]

4.3.4.1. User Log-in

Members who have previously registered on this page, admin or anonymous accounts can log in to the site. They can log in by typing a user name and password in the relevant section

4.3.4.2. New User

Users who have not registered before can register from this page. It is necessary to enter e-mail, password and user name in the relevant sections.

4.3.4.3. Show Restaurants in app

In this section, we can view the restaurants. We can also view the average ratings of restaurants.

4.3.4.4. Show Reviews of Restaurant(For Logged User)

Users who have previously registered on this page can comment on restaurants and view restaurant reviews.

4.3.4.5. Show Reviews of Restaurant(For Anonymous User)

Anonymous people cannot comment on restaurants on this page, but can view restaurant reviews.

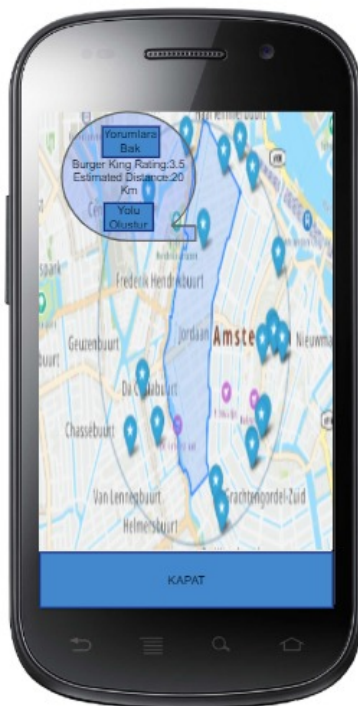Anonimlerde Yorumları Gösterme

4.3.4.6. New Review

This part is the part of commenting. After typing the comment in a certain field, the add comment button is clicked. At the same time, points are given to the restaurant from this part.
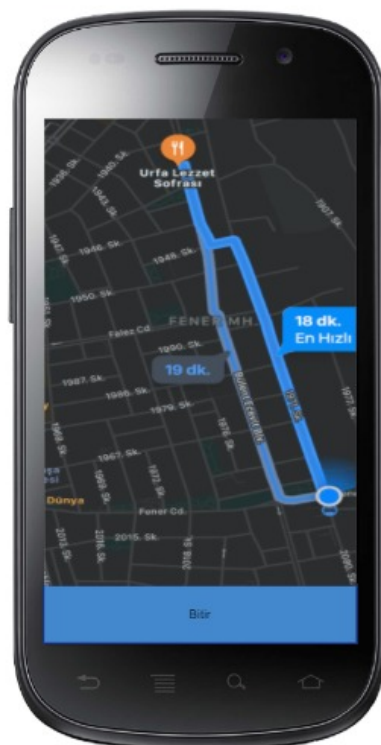

Yorum Ekleme

4.3.4.7. Show Restaurants in Map

Show Restaurants

4.3.4.8. Create Map

Create map

4.4. Constraints

4.4.1 Time

We got limited time to create this project, so we will use 3 steps to finish method(Which is just created by me that is dividing project to 3 major parts going step by step to completing them if it takes longer than it should have go past that step.);

-1st step is creating the general parts of project(Creating UI, Adding new user activating mail system etc..) which should take 1 week,

-2nd step is Adding details step which is Creating under layers of project(Update Reviews DB, Giving votes, calculating average, Banning user etc.) which should again take 1 week,

-3rd step is Completing the hardest part(which is Creating a map, Showing restaurants around user, etc.) and completing the project which should take around 2 weeks.

4.4.2. Performance

As our project is a mobile project that is going to run in Users phones our project's performance will be dependent on User's Internet, and Android phone's memory and their Android version.

4.4.3. Application Constraints

-User can only use approximately 5000 words when giving a review. Real value of word to be confirmed later.

-The maximum distance that a map can be drawed is approximately 20 kilometers. Real value of distance to be confirmed later.

4.4.4. Assumption and Dependencies

Assumption and Dependencies are written in the Software Specification Requirements.

5. Discussion

Information about the functioning of the system was shared in the SRS and SDD sections. In the discussion section, we talk about some potential problems that may occur in the system and their solutions (if any). Alternative improvements to the System Components (login, add restaurant, rating etc.) will be discussed in this section.

Multiple users sharing an account and rating simultaneously. We're not entirely sure where the problem might be, as we won't allow duplicate ratings (same user rated the same rating at the same restaurant). However, by comparing the browser-string/user-agent/geolocation information of the submission and the corresponding timestamp, we can use an algorithm to detect these cases and determine if there is a case where the same user account is rated from two different locations, or the browsers are within a very short time of each other.

Whether users will be allowed to withdraw their points is a topic that discussed in project. After the changes, it is aimed that the user will have the authority to withdraw or update their score.

There are ways to add a restaurant to the system and we discuss solutions on how it can be done. The system has an Administrator role which has the ability to add/modify restaurant details. Additionally, the system can allow registered users to "Add a Restaurant" but this addition is simply stored as "suggestions" which are evaluated by the administrator before they actually appear formally in the system.

6. Conclusion

As a result, it can be said that a literature review for the system was made and the methods to be used were decided according to this research. After the literature review, the software requirements specifications document is prepared. This document provides information about the functionality and performance criteria of the systems. The final document is the software design disclosure document, which contains information about the architecture of the system as well as the UI design information. This report covers all the theoretical aspects of recommendation systems and the next step in the project is the implementation of a system that covers the theoretical knowledge prepared in this document.

7. References

1. Layona Rita & Budi Yulianto, An Implementation of Location Based Service (LBS) for Community Tracking. Accessed on:
https://journal.binus.ac.id/index.php/comtech/article/view/3749/3129 .
2. Girish Deore1, Pranav Shete. To Study the Inclination of Consumers in Baner Area in Relation to the Online Food Ordering. Accessed on:
https://hmct.dypvp.edu.in/Documents/Girish-Paper.pdf .
3. Juan C. Correa, Phillip Broker, Gopal Sakarkar. Urban Mobility and Food Ordering Services: A Web Mining Perspective. Accessed on:
https://www.researchgate.net/publication/319493783_Urban_Mobility_and_Food_Ordering_Services_A_web_mining_perspective .
4. William R. King, Jun He. A meta-analysis of the technology acceptance model. Accessed on:
https://www.researchgate.net/publication/222297603_A_meta-analysis_of_the_Technology_Acceptance_Model#read .
5. Shantashree Das, Debomalya Ghose. Influence Of Online Food Delivery Apps On The Operations Of The Restaurant Business. Accessed on:
http://www.ijstr.org/final-print/dec2019/Influence-Of-Online-Food-Delivery-Apps-On-The-Operations-Of-The-Restaurant-Business-.pdf
6. Internet - Our World in Data. Accessed on:
https://ourworldindata.org/internet#:~:text=Globally%20the%20number%20of%20internet,online%20for%20the%20first%20time.
7. 10 Internet Statistics Every Marketer Should Know in 2021 [Infographic]. Accessed on:
https://www.oberlo.com/blog/internet-statistics.

8. The spread of true and false news online. Accessed on:
https://www.science.org/doi/10.1126/science.aap9559.

9. Yemeksepeti - Vikipedi. Accessed on: https://tr.wikipedia.org/wiki/Yemeksepeti.

10. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.590.1029&rep=rep1&type=pdf.

11. Tag(metadata) - Vikipedi. Accessed on: http://en.wikipedia.org/wiki/Tag_(metadata).

12. Koeppel, I. What are location services? From a GIS Perspective, ESRI white paper.2000.
Accessed on:
https://www.researchgate.net/publication/222679581_Location_Based_Services_and_GIS_i
n_Perspective.

13. Shiode, N., Li, C., Batty, M., Longley, P., & Maguire, D. The impact and penetration of
location-based services. In H. A. Karimi & A. Hammad (Eds.), Telegeoinformatics:
location-based computing and services, 2004, pp. 349–366, CRC Press. Accessed on:
https://www.researchgate.net/publication/32884883_The_impact_and_penetration_of_locati
on-based_services.

14. Bursztein, Elie; Bethard, Steven; Fabry, Celine; Mitchell, John C.; Jurafsky, Dan (2010).
"How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation" (PDF).
Proceedings of the 2010 IEEE Symposium on Security and Privacy: 399–413. CiteSeerX
10.1.1.164.7848. doi:10.1109/SP.2010.31. ISBN 978-1-4244-6894-2. S2CID 14204454.
Retrieved March 30, 2018. Accessed on:
https://web.stanford.edu/~jurafsky/burszstein_2010_captcha.pdf.

15. CAPTCHA - Wikipedia. Accessed on: https://en.wikipedia.org/wiki/CAPTCHA.

16. What Is a Use Case & How To Write One | Wrike. Accessed on:
https://www.wrike.com/blog/what-is-a-use-case/.

17. UML Use Case Examples of Common Scenarios | EdrawMax. Accessed on:
https://www.edrawsoft.com/article/use-case-diagram-examples.html.

18. Designing Use Cases for a Project - GeeksforGeeks. Accessed on:
https://www.geeksforgeeks.org/designing-use-cases-for-a-project/.

19. Akıllı telefon harcamaları yüzde 50 arttı - Son Dakika Haberleri. Accessed on:
https://www.trthaber.com/haber/ekonomi/akilli-telefon-harcamalari-son-bir-yilda-yuzde-50-artt
i-571161.html.

20. Google Maps Platform: Maps API nedir? | Başarsoft. Accessed on:
https://www.basarsoft.com.tr/google-maps-platform-api-nedir/.

21. API Nedir? Ne İşe Yarar? | ceaksan | Web Analisti & Veri Bilimi Meraklısı. Accessed on:
https://ceaksan.com/tr/api-nedir.

22. IEEE Std 1016-2009 (Revision of IEEE Std 1016-1998), IEEE Standard for Information
Technology--Systems Design--Software Design Descriptions. Accessed on:
http://cengproject.cankaya.edu.tr/wp-content/uploads/sites/10/2017/12/SDD-ieee-1016-2009
.pdf.

23. Home - Pencil Project. Accessed on: https://pencil.evolus.vn.

24. Flowchart Maker & Online Diagram Software. Accessed on: https://app.diagrams.net.

25. Lucid visual collaboration suite: Log in. Accessed on:
https://lucid.app/documents#/documents?folder_id=home.