



# T.C.

# MARMARA UNIVERSITY

# FACULTY of ENGINEERING

# COMPUTER ENGINEERING DEPARTMENT

## CSE3044 Software Engineering **Homework #3**

## **Project Name :FitEats**

Group Members

Fatih Genç 150119905

Umut Berke Pezük 150119712

Alperen Koruyucu 150119734

Hakan Adaklı 150116069

Mevlüt Eren Topal 150117025

Ceyhun Erdönmez 150120851

Ahmet Bozbay 150119861

Yunus Emre Ocak 150122610

Supervised by

Mustafa Borahan Tümer

# Table of Contents

# 1. Introduction

The demands of modern life can make it difficult to maintain a healthy diet. Long work hours, hectic schedules, and endless to-do lists can leave us with little time or energy to plan and prepare nutritious meals. This is where Fiteats comes in. Our web application makes it easy to receive customized meal recommendations based on their individual preferences. Fiteats can help you achieve your goals by providing you with a range of nutritious and delicious meal options that are easy to prepare and fit seamlessly into your busy lifestyle. With Fiteats, you can take control of your diet and feel great, no matter how busy your day may be.

In addition to providing users with customized meal recommendations, Fiteats is also a software development project that involves a variety of design and development tasks. Throughout the development process, our team will be creating a range of diagrams and models to help guide the project. These may include UML diagrams that outline the system's structure and components, as well as sequence diagrams that depict the flow of actions between different parts of the system. By using these diagrams and models, we can ensure that the software is well-organized, efficient, and easy to maintain. This attention to detail will ultimately result in a better user experience for Fiteats users, who can enjoy seamless and intuitive meal planning without any technical hiccups.

## 1.1 Purpose

At Fiteats, our main purpose is to make it easy for users to follow a diet program that meets their daily calorie needs. By providing personalized meal recommendations that are tailored to each user's individual preferences and goals, we aim to simplify healthy eating and help people achieve their desired outcomes. Whether you're trying to lose weight, build muscle, or simply eat more healthily, Fiteats can provide you with the tools and resources you need to stay on track and reach your goals. Our ultimate purpose is to empower our users to take control of their diet and achieve optimal health and wellbeing.

## 1.2 Scope

The software product to be produced by Fiteats is a web application that provides personalized meal recommendations to users based on their daily calorie needs and individual preferences. The software will allow users to create a profile and receive a list of recommended meals for each day based on their preferences and goals.The software will include features for storing user information and pulling recipes and ingredients from external datasets. The application will determine the maximum and minimum daily calorie intake for the user, based on their specific goals and needs (such as building muscle, losing weight, or gaining weight), and divide this by the number of meals per day to recommend suitable calorie limits for each meal. The software will recommend meals within the recommended calorie limits for each meal, based on the user's preferences and dietary restrictions. The user will also be able to search for recipes based on specific ingredients. These features are designed to simplify the meal planning process, provide users with quick and easy access to healthy meal options, and empower them to take control of their diet and make informed choices about their health.

## 1.3 Definitions, Acronyms, and Abbreviations

User : A person who wants to keep track of his own training program ,diet , body measurements.
RSD: Requirements Specification Document
UCD: Use Case Diagram
UI: User Interface
Python: It is a programming language.
Django: It is a high-level Python web framework that enables rapid development of secure and maintainable websites.
HTML: Hypertext Markup Language is a standard markup language used to create web pages.
Bmi : Body Mass index is a value used on healthy body weight calculations.

## 1.4 References

https://www.kaggle.com/datasets/mehmetuslu/food-recipe-and-dataset-with-nutritional-values
https://www.kaggle.com/datasets/hugodarwood/epirecipes?select=full_format_recipes.json
https://mimoza.marmara.edu.tr/~birol/Courses/CSE3044/lecture_notes.htm
-- Ian Sommerville, Software Engineering, 8th ed. 2007

# 2. General Description

FitEats is a web platform that provides users with healthy recipes and meal plans that are tailored to their individual needs. Users can search for recipes based on criteria such as cuisine type, and the platform offers personalized meal plans. With FitEats, users can easily find and prepare healthy meals to help them achieve their calories goals and lead a various meal style.

## 2.1 Product Perspective

FitEats is a comprehensive web platform that aims to provide users with a convenient and efficient way to achieve their meal goals through healthy eating. It offers a range of healthy recipes and personalized meal plans that cater to individual needs.

## 2.2 Product Functions

- Offers recipes by category and ingredient.
- Search for healthy recipes by cuisine and time.
- Healthy recipes for different needs.
- Recipe categories: meal, cuisine, and diet.
- Find recipes by ingredient with search bar.
- Suggest calorie intake based on goals.

## 2.3 User Characteristics

- People who want to improve their eating habits.
- Individuals who want to manage their weight and calorie intake.
- People seeking to plan and prepare healthy meals.
- Anyone interested in discovering new healthy meal options and recipes.

## 2.4 General Constraints

- The website's design must be responsive and optimized for use on both desktop and mobile devices.
- Compatibility with popular web browsers such as Chrome, Firefox, Safari, and Edge is required.
- The website's performance must be optimized for quick loading times and a seamless user experience.
- The website must implement robust security measures to protect user data and prevent unauthorized access.

- The website must be able to handle a large volume of traffic, particularly during peak periods or special events.
- The website's design must be scalable to allow for future growth and development.
- Integration with third-party APIs, including those for recipe databases or nutrition data must be supported.
- The website must be designed to be easily maintained and updated by the development team.

## 2.5 Assumptions and Dependencies

Assumptions:

- It is assumed that the user will have a device with a compatible web browser to access the website.
- The website will assume that the user provides accurate personal information, such as age, weight, height, and activity level, to calculate their daily calorie needs.
- The website will assume that the calorie information provided for the recipes is accurate and up-to-date.
- It is assumed that the user has basic computer skills and understands how to navigate a website.

Dependencies:

- The website will depend on a reliable and accurate source for the calorie information of the ingredients used in the recipes.
- The website will depend on a database to store recipe information, such as recipe names, ingredients, and instructions.
- The website will depend on a web hosting service that can handle a large number of visitors and ensure fast loading times.
- The website will depend on Python, Django, HTML, Bootstrap, CSS to function properly.
- The website will depend on SQLite and JSON files for database management and storage.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The user will be redirected to the login page by clicking the login button on the navbar. After entering the username and password on the login page, he will be logged into his own account. User can see user information from profile page. Can view the prepared recipes by clicking the recipes button on the navbar.

### 3.1.2 Hardware Interfaces

All devices that can connect to the internet and view internet pages such as smart phones, tablets, smart TVs, computers can be used.

### 3.1.3 Software Interfaces

A modern web browser that can run html, css and javascript technologies should be used.

### 3.1.4 Communications Interfaces

In order to establish a correct connection between the user and the server, protocols such as TCP, IP, HTTP need to be applied during the connection.

## 3.2 Functional Requirements

3.2.1 <Sign up>
3.2.1.1 Introduction / Description
Sign up function signs up the users to the app.
3.2.1.2 Inputs / Display
Users can sign up by tapping the sign up button with their e-mails.
3.2.1.3 Processing
It sends information to the database.
3.2.1.4 Outputs
Successful, if the user registered; otherwise, an error message appeared.
3.2.1.5 Constraints
Emails must be in email format.
3.2.1.6 Error/Data Handling
3.2.2 <Sign in>
3.2.2.1 Introduction / Description
The user information is verified by this function, which then grants them entry to their accounts.

3.2.2.2 Inputs / Display

Users can also sign in by tapping the sign in button by writing their email and password.

3.2.2.3 Processing

This function searches the database with the given parameters. If it finds the given email, check its password to allow the user to enter the app or not.

3.2.2.4 Outputs

Successfully enter the app or error message appeared.

3.2.2.5 Constraints

Users can enter passwords wrong at most five times consecutively.

3.2.2.6 Error/Data Handling

3.2.3 <Calorie tracking>

3.2.3.1 Introduction / Description

The user will be able to track calories from the application.

3.2.3.2 Processing

The user will be able to track calories by entering the foods they consume into the application.

3.2.4 <Body Mass Index (BMI) Calculator>

3.2.4.1 Introduction / Description

This feature will calculate the bmi of a person with the formula considering the person's height and body weight.

3.2.4.2 Inputs / Display

Gender, height, neck and waist measurements.

3.2.4.3 Processing

Bmi = Weight / (Height)^2

3.2.4.4 Outputs

This feature will output a calculated (approximate) bmi (body mass index ) for individuals.

3.2.4.5 Constraints

Height values should be in CM.

Weight values should be in KG.

3.2.5 <Sign out>

3.2.5.1 Introduction / Description

Sign out the account.

3.2.5.2 Inputs / Display

Sign out button.

3.2.5.3 Processing

Sign out the user from the account and return the app’s main interface.

3.2.5.4 Outputs

3.2.5.5 Constraints

3.2.5.6 Error/Data Handling

## 3.3 Non-Functional Requirements

### 3.3.1 Performance

- The system must be able to serve at least 100 users simultaneously.
- The calculation and display of recipes should take no longer than 10 seconds depending on the user's device and internet connection speed.

### 3.3.2 Reliability

- The system must be able to provide service to users 24/7.
- The system must provide the user with accurate recipe selections and calculations.

### 3.3.3 Availability

- The availability of the system in a localhost environment is dependent on the hardware and software resources of the local machine, and the ability of the system to efficiently manage these resources while running multiple processes and handling user requests.

### 3.3.4 Security

- The system must securely store user data.
- The system must protect user privacy.

### 3.3.5 Maintainability

- The code should be clear and understandable.

### 3.3.6 Portability

- The system should be able to run on different operating systems (Windows, Mac, Linux).
- The system should be able to run on different devices (computer, phone, tablet).

## 3.4 Inverse Requirements

- The system should not provide users with incorrect recipes.
- The system should not make incorrect portion calculations for users.

## 3.5 Design Constraints

- The system should use  MySQL database.
- The website should be compatible with modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
- The website's images and content should not contain suitability, illegal, or immoral content.

## 3.6 Logical Database Requirements

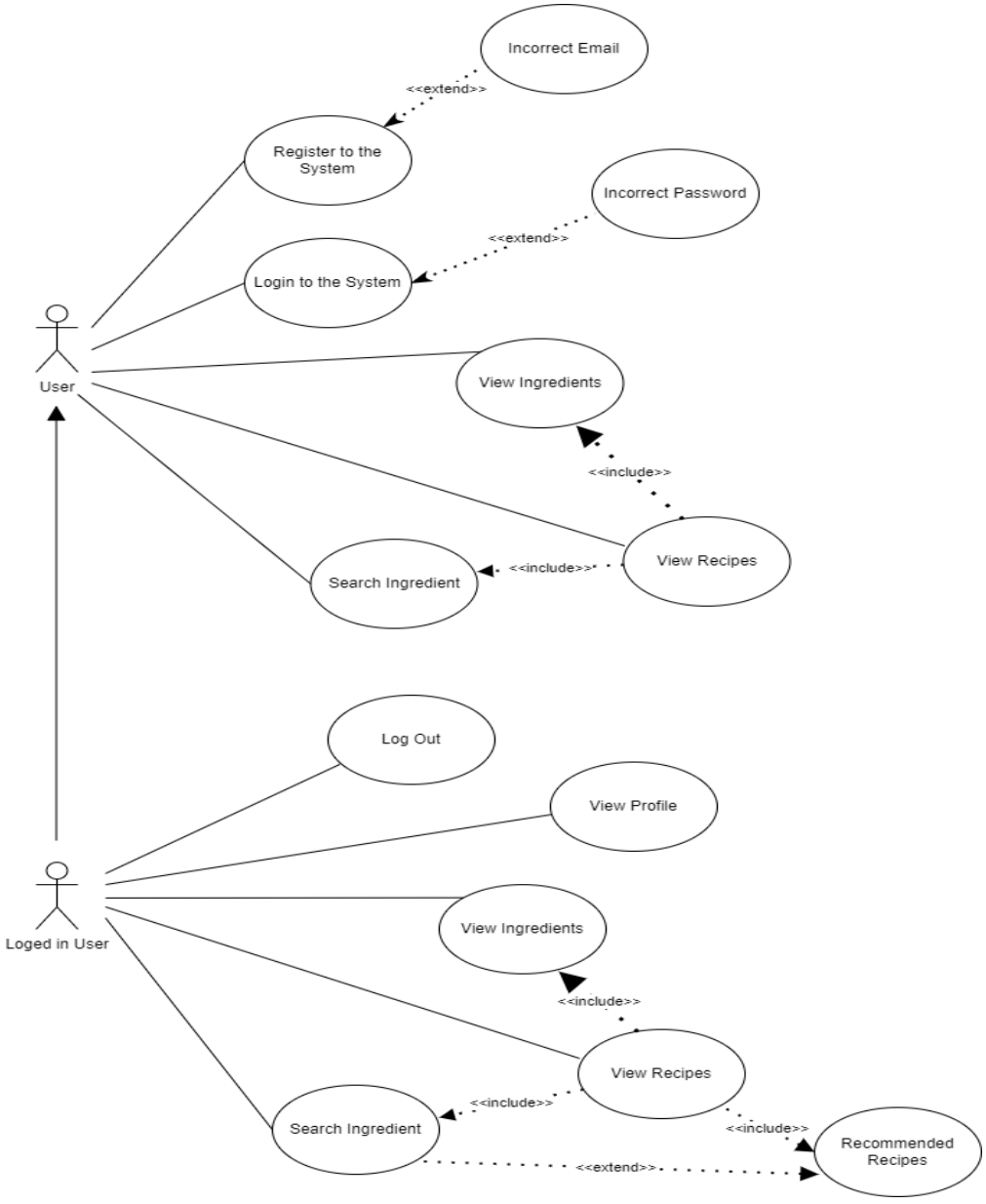
- A database will be used for the website.
- The database should be appropriately protected to ensure the secure storage of data.

## 3.7 Other Requirements

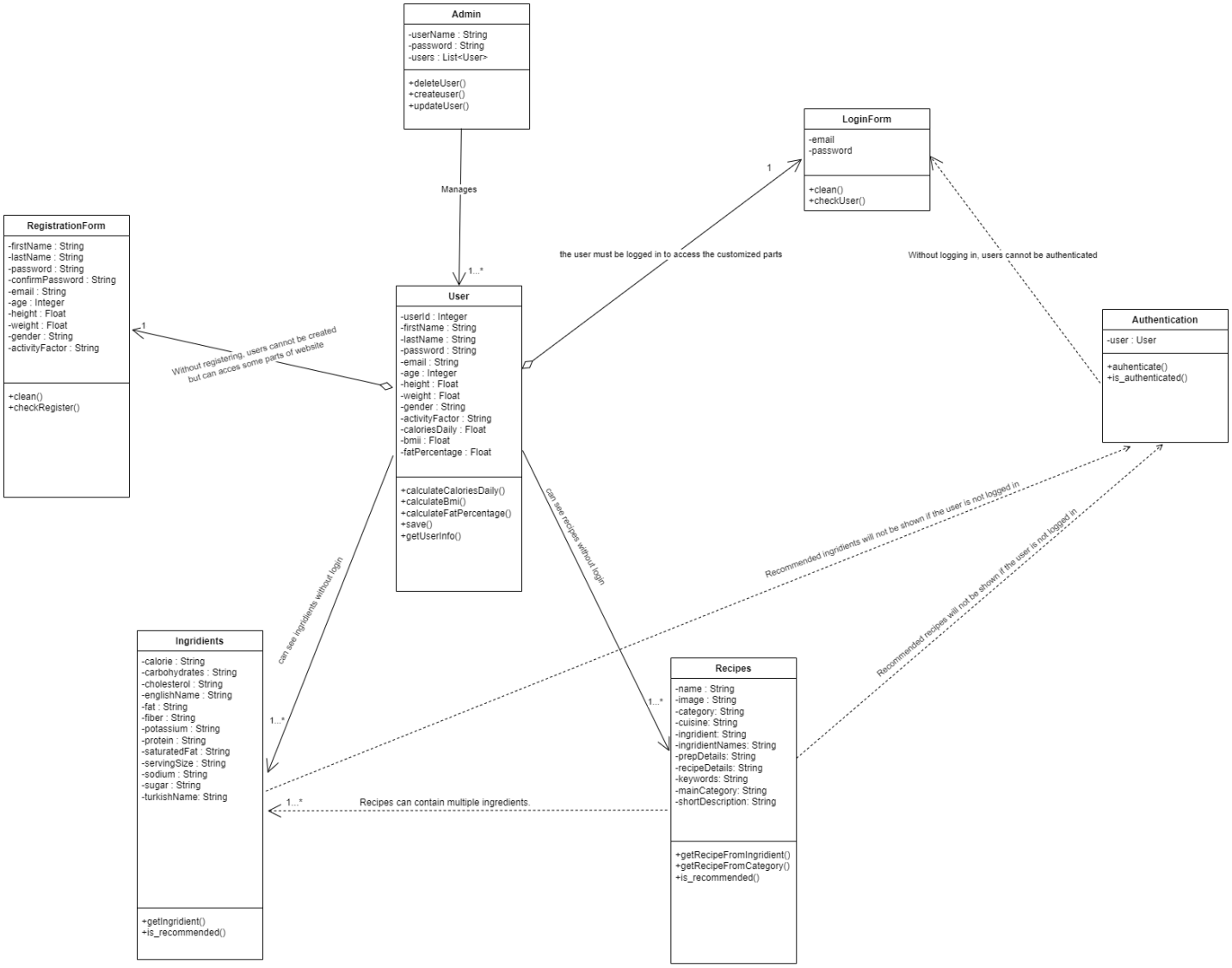
- User accounts should be protected with encryption for security purposes.
- The website should provide users with healthy meal recipes that support a healthy lifestyle.

# 4. UML Diagrams

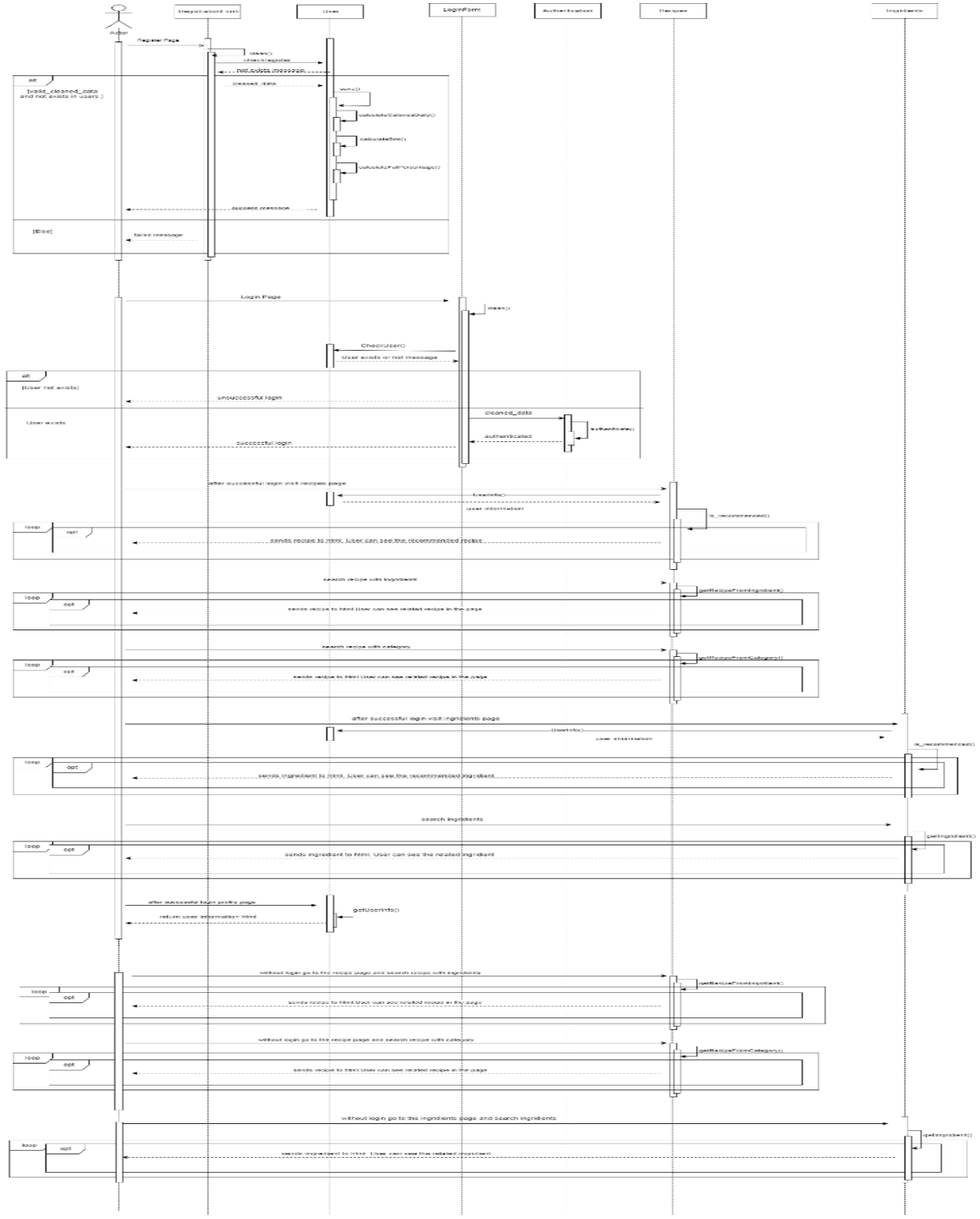
## 4.1 Use Cases



## 4.2 Classes / Objects




## 4.3 Sequence Diagrams


Actor
RegisterPage
User
LoginForm
Authentication
Recipes
Ingredients
clean()
check register
not exists !!!!!!!!!!
cleaned_data
save()
calculateDailyCalorieGoal()
calculateBmi()
calculateFatPercentage()
success message
[Else]
failed message
Login Page
clean()
CheckUser()
User exists or not message
alt
[User not exists]
unsuccessful login
User exists
cleaned_data
authenticate()
authenticated
successful login
after successful login visit recipes page
UserInfo()
user information
is_recommended()
loop
opt
sends recipe to html, User can see the recommended recipe
search recipe with ingridient
getRecipeFromIngridient()
sends recipe to html User can see related recipe in the page
search recipe with category
getRecipeFromCategory()
after successful login visit ingridients page
sends ingredient to html, User can see the recommended ingredient
search ingridients
getIngridient()
sends ingredient to html. User can see the related ingridient
after successful login profile page
return user information html
getUserInfo()
without login go to the recipe page and search recipe with ingridients
without login go to the recipe page and search recipe with category
without login go to the ingridients page and search ingridients


Admin
Admin
User
Login to admin panel
deleteUser()
createUser()
updateUser()

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

## Appendixes

The text describes a diet program that helps users plan and manage their diets based on their age, gender, weight, height, and activity level. The program is designed to promote healthy weight loss by providing a balanced nutrition plan and progress tracking system. Additionally, the program can be personalized to meet individual needs by allowing users to request certain foods be added or removed.Two potential future features include a blog system for users to share their own programs and a rating and liking system to create a list of the best programs monthly and yearly. The overall goal of the project is to promote a healthy lifestyle and motivate users by allowing them to access and exchange ideas with others.To proceed with mail verification, I need to know what specific information or action you want to verify via email.

# 6.Distribution of Tasks

| | |
|---|---|
| Umut Berke Pezük | -1.1 Purpose<br>-1.2 Scope<br>-1.3 Definitions, Acronyms, and Abbreviations<br>-1.4 References<br>-Final control of distribution of tasks<br>-General control of SRS document and edit |
| Fatih Genç | -2.4 General Constraints<br>-2.5 Assumptions and Dependencies<br>**-4.2 Class Diagram**<br>**-4.3 Sequence Diagram**<br>-Overall design of SRS document<br>-General control of SRS document and edit<br>-General control and regulations |
| Alperen Koruyucu | -3.2 Functional Requirements<br>-Overall design of SRS document<br>-General control and regulations<br>-General control of SRS document and edit<br>-Started writing frontend codes |
| Hakan Adaklı | -3.3 Non-Functional Requirements<br>-3.4 Inverse Requirements<br>-3.5 Design Constraints<br>-3.6 Logical Database Requirements<br>-3.7 Other Requirements<br>-General control and regulations<br>-General control of SRS document and edit |

| | |
|---|---|
| Mevlüt Eren Topal | -2 General Description<br>-2.1 Product Perspective<br>-2.2 Product functions<br>-2.3 User Characteristics |
| Ceyhun Erdönmez | -4.1 Use Cases |
| Ahmet Bozbay | -Appendices<br>-Editing part 2 |
| Yunus Emre | -Geç katıldı |