



**Module Code & Module Title**

**CS6P05NI Final Year Project**

**Assessment Weightage & Type**

**25% FYP Interim Report**

**Semester**

**2023-24 Autumn**

**PROJECT TITLE: fitness app**

**Student Name:**

**London Met ID:**

**College ID:**

**Internal Supervisor:**

**External Supervisor:**

**Assignment Due Date:**

**Assignment Submission Date:**

**Word Count: 3898**

*I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

# Sample 2 - BIC.docx

 Islington College, Nepal

---

## Document Details

### Submission ID

trn:oid:::3618:73855439

### Submission Date

Dec 12, 2024, 2:05 PM GMT+5:45

### Download Date

Dec 13, 2024, 8:28 AM GMT+5:45

### File Name

Sample 2 - BIC.docx

### File Size

73.0 KB

124 Pages





10,895 Words

62,503 Characters




# 4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Match Groups

-  **10 Not Cited or Quoted 4%**  
Matches with neither in-text citation nor quotation marks
-  **2 Missing Quotations 0%**  
Matches that are still very similar to source material
-  **0 Missing Citation 0%**  
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 2%  Internet sources
- 1%  Publications
- 1%  Submitted works (Student Papers)

## Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Match Groups

- 105** Not Cited or Quoted 4%  
Matches with neither in-text citation nor quotation marks
- 2** Missing Quotations 0%  
Matches that are still very similar to source material
- 0** Missing Citation 0%  
Matches that have quotation marks, but no in-text citation
- 0** Cited and Quoted 0%  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 2% Internet sources
- 1% Publications
- 1% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

<b>1</b>	Submitted works		
	Swinburne University of Technology on 2023-06-05		0%
<b>2</b>	Submitted works		
	HELP UNIVERSITY on 2013-09-09		0%
<b>3</b>	Submitted works		
	Southern New Hampshire University - Continuing Education on 2023-07-23		0%
<b>4</b>	Submitted works		
	University of Maryland, Global Campus on 2022-11-01		0%
<b>5</b>	Submitted works		
	Asia Pacific University College of Technology and Innovation (UCTI) on 2020-08-19		0%
<b>6</b>	Internet		
	www.nofoodwaste.in		0%
<b>7</b>	Submitted works		
	Manchester Metropolitan University on 2016-04-15		0%
<b>8</b>	Submitted works		
	HELP UNIVERSITY on 2018-05-21		0%
<b>9</b>	Internet		
	www.geeksforgeeks.org		0%
<b>10</b>	Submitted works		
	Adventist University of Central Africa on 2023-08-25		0%

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Project Description .....	1
1.2 Current Scenario .....	2
1.3 Problem domain and the project solution .....	2
1.3.1 Problem domain .....	2
1.3.2 The project solution .....	3
1.4 Aim and Objectives.....	3
1.4.1 Aim .....	3
1.4.2 Objectives .....	4
1.5 Structure of the report .....	5
1.5.1 Background .....	5
1.5.2 Development .....	5
1.5.3 Project progress.....	5
1.5.4 Progress review .....	5
<b>2. Background .....</b>	<b>6</b>
2.1 Project elaboration .....	6
2.2 Similar projects .....	7
2.2.1 Cronometer .....	7
2.2.2 MyFitnessPal.....	8
2.2.3 Nutritionist Pro (Diet Analysis Module).....	9
2.3 Comparisons .....	10
<b>3. Development .....</b>	<b>11</b>
3.1 Considered methodology .....	11
3.1.1 Prototype Model.....	11
3.1.2 Spiral Model.....	13
3.1.3 Waterfall Model .....	15
3.2 Selected methodology .....	17

3.2.1 Scrum methodology .....	17
3.2.2 Reason for choosing scrum .....	18
3.2.2 Phases of Scrum and its implementation .....	19
<b>4. Project Progress .....</b>	<b>21</b>
4.1 Product backlog .....	21
4.2 Development .....	22
4.2.1 Sprint 1 – Signup and Login .....	22
4.3 Gantt Chart.....	29
4.4 Work Break Down Structure .....	30
<b>5. Progress Review .....</b>	<b>31</b>
5.1 Progress Table.....	31
5.2 Further Work.....	32
5.3 Project risks, threats, and contingency plans .....	35
5.4 Project Boundaries and contingency plans .....	36
<b>6. Bibliography .....</b>	<b>37</b>
<b>7. References .....</b>	<b>37</b>
<b>8. Appendix.....</b>	<b>39</b>
<b>8.1 SRS Document.....</b>	<b>39</b>
8.1.1 Introduction.....	40
8.1.2 Overall Description .....	41
8.1.3 External Interface Requirements.....	44
8.1.4 System Features .....	47
8.1.5 Other Non-Functional Requirements .....	55
8.2 Design .....	57
8.2.1 Mind Map.....	57
8.2.2 System Architecture.....	58
8.2.3 Use Case Diagram.....	59
8.2.4 Sequence Diagram .....	60

8.2.5 Collaboration Diagram.....	68
8.2.6 Flow Chart .....	76
8.2.7 Entity Relation Diagram .....	84

## List of Figure

Figure 1: Cronometer (Img source: cronometer) .....	7
Figure 2: MyFitnessPal (Img source: MyFitnessPal) .....	8
Figure 3:Diet Analysis Module (Img source: Dite Analysis Module).....	9
Table 1: Features Comparison .....	10
Figure 4: Prototype Model (Img source: prototype) .....	11
Figure 5: Spiral Model (Img source: spiral) .....	13
Figure 6: Waterfall Model (Img source: waterfall model).....	15
Figure 7: Scrum Methodology (Img source: scrum).....	17
Figure 8:Product Backlog .....	21
Figure 9: Login Wireframe .....	22
Figure 10: Signup Wireframe .....	23
Figure 11: Login UI .....	24
Figure 12: Signup UI .....	24
Figure 13: Signup Model Class.....	25
Figure 14: Signup forms.py 1 .....	26
Figure 15: Signup forms.py 2 .....	26
Figure 16: Signup view .....	27
Figure 17: User Table in PostgreSQL.....	27
Figure 18: User table in admin panel.....	27
Figure 19: Login form.py.....	28
Figure 20: Login view .....	28
Figure 21:Gantt Chart .....	29
Figure 22: Work Break down Structure.....	30
Table 2: Progress Table .....	31
Figure 23: System Architecture SRS .....	41
Figure 24: Login UI SRS .....	44
Figure 25: Signup UI SRS .....	45
Table 3: Signup Functional Requirements.....	48
Table 4: Login Functional Requirements .....	49



Table 5: Exercise Functional Requirements .....	50
Table 6: Food Functional Requirements .....	51
Table 7: Trends Functional Requirements .....	52
Table 8: Share Functional Requirements .....	53
Table 9: Coach Management Functional Requirements .....	54
Table 10: Performance Non-FR.....	55
Table 11: Safety Non-FR .....	55
Table 12: Security Non-FR.....	56
Figure 26: Mind Map .....	57
Figure 27: System Architecture .....	58
Figure 28: Use Case Diagram .....	59
Figure 29: Login Sequence Diagram .....	60
Figure 30: Signup Sequence Diagram .....	61
Figure 31: Food Sequence Diagram .....	62
Figure 32: Exercise Sequence Diagram .....	63
Figure 33: Share Sequence Diagram.....	64
Figure 34: Trends Sequence Diagram.....	65
Figure 35: Client Profile Sequence Diagram .....	66
Figure 36: Client info Sequence Diagram .....	67
Figure 37; Login Collaboration Diagram .....	68
Figure 38: Signup Collaboration Diagram.....	69
Figure 39: Food Collaboration Diagram.....	70
Figure 40: Exercise Collaboration Diagram .....	71
Figure 41: Share Collaboration Diagram .....	72
Figure 42: Trends Collaboration Diagram .....	73
Figure 43: Client Profile Collaboration Diagram .....	74
Figure 44: Client info Collaboration Diagram.....	75
Figure 45: Login Flow Chart .....	76
Figure 46: Signup Flow Chart.....	77
Figure 47: Food Flow Chart.....	78
Figure 48: Exercise Flow Chart .....	79

Figure 49: Share Flow Chart.....	80
Figure 50: Trends Flow chart.....	81
Figure 51: Create Profile Flow Chart.....	82
Figure 52: Client Info Flow Chart .....	83
Figure 53; Entity Relation Diagram.....	84

## List of Tables

Table 1: Features Comparison .....	10
Table 2: Progress Table .....	31
Table 3: Signup Functional Requirements.....	48
Table 4: Login Functional Requirements .....	49
Table 5: Exercise Functional Requirements .....	50
Table 6: Food Functional Requirements.....	51
Table 7: Trends Functional Requirements.....	52
Table 8: Share Functional Requirements.....	53
Table 9: Coach Management Functional Requirements.....	54
Table 10: Performance Non-FR.....	55
Table 11: Safety Non-FR .....	55
Table 12: Security Non-FR.....	56

# **Acknowledgement**

I would like to thank my university London Metropolitan University including this CS6P05NI Final Year Project module in curriculum and providing this coursework which helped me learn about new things related to web development and helps me to learn how to research properly. Lastly, I would like to thank my internal and external supervisor who helped me and guided me in this module and coursework.

# **Abstract**

This report provides the overview of development of *fitness app* web application. Report starts with introduction explain what this project is, why is the project is being done and aims and objectives for doing this project. Comparison with similar project is also shown. Why scrum methodology is used for building this project plus comparison with other methodology is also shown. Project progress and review is also shown. Appendix contains SRS report And Software development diagrams.

## 1. Introduction

In today's world with sedentary lifestyle, fast food and demanding work schedules maintaining regular exercise routines and balanced nutrition have become concern of every individual. Worldwide more than 1 billion people have obesity- 650 million adults, 340 million adolescents and 39 million children (Forbes, 2023). According to World Health Organization (WHO), insufficient physical activity is the 4th leading risk factor for mortality, contributing to approximately 3.2 million deaths globally each year and people who are physically not active have risk of all-cause of mortality compared to people who are engaged in at least 30 minutes of moderate intensity physical activity for most of the days of the week (World Health Organization, n.d.).

### 1.1 Project Description

The project is named *fitness app*. The *fitness app* is a web-based fitness application, with features of workout logging, nutrition tracking, progress analytics and coach management. The web application will allow users to log daily workouts in detail like exercise name, sets, reps, types (strength, cardio, flexibility), targeted muscle groups, equipment used, and workout duration. Users can also create custom exercise. Users can log their daily meals, including food items, portion sizes, and nutritional information. The web application will have a detailed food database with pre-populated nutritional data and the flexibility for users to add custom food items manually. Users can visualize fitness progress through various charts like line graphs, bar charts, and pie charts. User can share data to coach. Coach can create individual tracker profile for their clients which will allow coach to give personalize guidance.

## 1.2 Current Scenario

According to a report from IHRSA (International Health, Racquet & Sports Club Association) the number of people who join gyms each year is estimated to be around 100 million (Sharma, 2023). Fitness application have become very popular in recent years especially after covid-19 lockdown. The global fitness application market size as of 2023 is USD 1.54 billion and is projected to grow at a compound annual growth rate (CAGR) of 17.7% from 2024 to 2030 (Grand View Research , 2022). The fitness applications are popular because it offers wide range of features and functionalities that help users achieve their fitness goals, stay motivated and track their progress. Also, in Nepal fitness industry is growing very fast especially in city areas, the fitness Equipment eCommerce market in Nepal is also predicted to grow, with an estimated growth rate of 8% reaching revenue of USD 4.9 million (ecommercedb, 2022).

## 1.3 Problem domain and the project solution

### 1.3.1 Problem domain

- According to International Health Racquet (IHRSA) 50% of all new gym members quit within the first 6 months (Spraul, 2023). It is tough for people to track workouts, manage nutrition and visualize progress, hindering motivation and goal attainment.
- According to Statista 30% of gym members use a personal trainer (Spraul, 2023). It is difficult for coaches to monitor and manage multiple clients' workout logs, nutrition records, and progress analytics, limiting the delivery of personalized guidance to the client.

### 1.3.2 The project solution

The project will address the problems by developing a web application focused on workout and nutrition tracking, progress analytics and coach management.

- According to the research “*Does Monitoring Goal Progress Promote Goal Attainment? A Meta-Analysis of the Experimental Evidence*” suggest that monitoring goal progress is an important process that helps in attaining a goal and ensure that goals are translated into action (Harkin, 2016). So, the web application will have features where users can log their daily workouts, including exercises, sets, reps, and weights. Users will be able to track their daily meals including food items and portion sizes.
- According to the research “*Which Factors Are Associated with Monitoring Goal Progress?*” suggest that people were more willing to monitor their progress and found it easier to do when they thought about their goal in quantifiable terms (Chang, 2017). So, the visual representation of data trends can be viewed by users in the web application.
- Coaches will have a separate interface for managing and effectively monitoring of clients’ workout logs, nutrition records, and progress analytics for personalized guidance.

## 1.4 Aim and Objectives

### 1.4.1 Aim

The aim is to develop a web application that integrates workout tracking, nutrition tracking, progress visualization and coach management, with an aim of motivating users to attain their fitness goal and help coaches to manage multiple clients effectively.



### **1.4.2 Objectives**

- To research about fitness industry.
- To learn about UI and UX for developing seamless user experience.
- To learn Django framework for backend of the web application.
- To learn about Application Programming Interface (API).
- To create interface for workout tracking and nutrition tracking for collecting data from users.
- To learn and implement PostgreSQL to manage data.
- To learn and implement data visualization in Chart.js.
- To create separate interface for coaches to manage clients.
- To implement robust security protocols to safeguard users' data.
- To document the project.

## **1.5 Structure of the report**

### **1.5.1 Background**

The background portion of the report will describe the project in detail and give clear understanding about the intended user type for the project. This section will also show the analysis of the similar projects in the market. This section will also compare various features offered by the other projects with this project.

### **1.5.2 Development**

The development portion of the report documents the methodology used in the project. This section will also explain about the other considered methodologies and selected methodology for the project.

### **1.5.3 Project progress**

The Project progress section of the report documents an update on the status of the project. It shows the completed tasks, currently in progress and upcoming tasks. This section will have Gantt chart to showcase the project timeline and Work Breakdown Structures to illustrate the project's structural hierarchy.

### **1.5.4 Progress review**

The Progress review section of the report documents what project is doing and what's coming next. The "Progress Table" shows what is achieved, current working task, and what's ahead. The "Further Work" outlines the next important tasks need to do to keep moving forward. This section also shows project risks and boundaries and contingency plans.

## 2. Background

### 2.1 Project elaboration

As stated in introduction, in today's world with sedentary lifestyle, fast food and demanding work schedules maintaining regular exercise routines and balanced nutrition have become concern of every individual. This project *fitness app* aims to help people who wants to be healthy and achieve their fitness goals and for those who lacks motivation to be healthy. It will also help coaches to manage multiple clients simultaneously by monitoring client progress, allowing coach to give personalized guidance to the client without any limitations.

The *fitness app* will have four key features (Exercise) workout logging, (Food) nutrition tracking, (Trends) progress analytics and coach management.

- Users will be able to log their daily workouts, including exercises details like name, types, sets, reps, weights, targeted muscle groups, and equipment used systematically.
- Users will be able to input daily meals, including food items, portion sizes, and nutritional information, offering either pre-populated nutritional data or the option to add custom food items.
- Users will be able to set their workouts goals and nutritional goals. Progress analytics will be a key feature of the web application, where users can view visual representations of their fitness trends using line graphs, bar charts, and pie charts, along with graphical displays showing their progress over time.
- There will be separate interface for coaches which will help coaches to manage multiple clients' data, view workout logs, nutrition records, and progress analytics, ensuring personalized guidance and feedback.

The web application will be responsive for most of the devices. The web application will have robust encryption methods, such as SSL/TLS protocols, to encrypt data during transmission and storage, ensuring the protection of sensitive user data. The web application will have proper input validation to prevent from wrong inputs and SQL injection attacks. The web application will have Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) Protection. The project will have detailed technical documentation providing insights into the software architecture used for future developments.

## 2.2 Similar projects

### 2.2.1 Cronometer

Cronometer is a health and fitness application that offer various features for tracking diet, exercise, and health metrics. It has a database of over 1.2 million verified foods which allows users to track meals and have detailed breakdown of macro-nutrients, micro-nutrients. It also allows users to create custom diet settings and fasting. It allows users to track their workouts, application also integrates with Apple Health and Fitbit, allowing users to track their heart rate, sleep, and exercise. It also gives a nutrition and workout report. The application also allows users to set diet goals, calories goals. Cronometer uses nutritional data from verified sources. Cronometer does not sell user data to third parties, ensuring user that their data is protected and remains private.

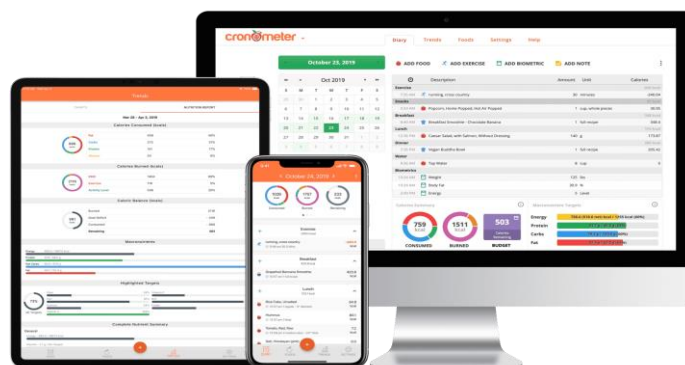


Figure 1: Cronometer (Img source: [cronometer](https://cronometer.com))

Some of the features inspired from Cronometer for my project *fitness app* are:

- Macro-nutrition and Micro-nutrition tracking
- User Interface
- Exercise logging feature
- Trends feature (Charts)

### 2.2.2 MyFitnessPal

MyFitnessPal is health and fitness smartphone application and website that helps users to track daily food intake, it has data of food over 11 million foods. The application allows user to scan the barcode of food packages to quickly log nutritional information. It allows users to set personalized goals for weight management, nutrition, and fitness. It allows users to track exercise and physical activity. It allows users to connect with each other, share progress and get support from a community. It also has a feature to track daily water intake. The application can be integrated with wearable fitness tracker devices to automatically sync data. The application has a collection of recipes and meal ideas. The application shows user progress visually on charts and graphs.

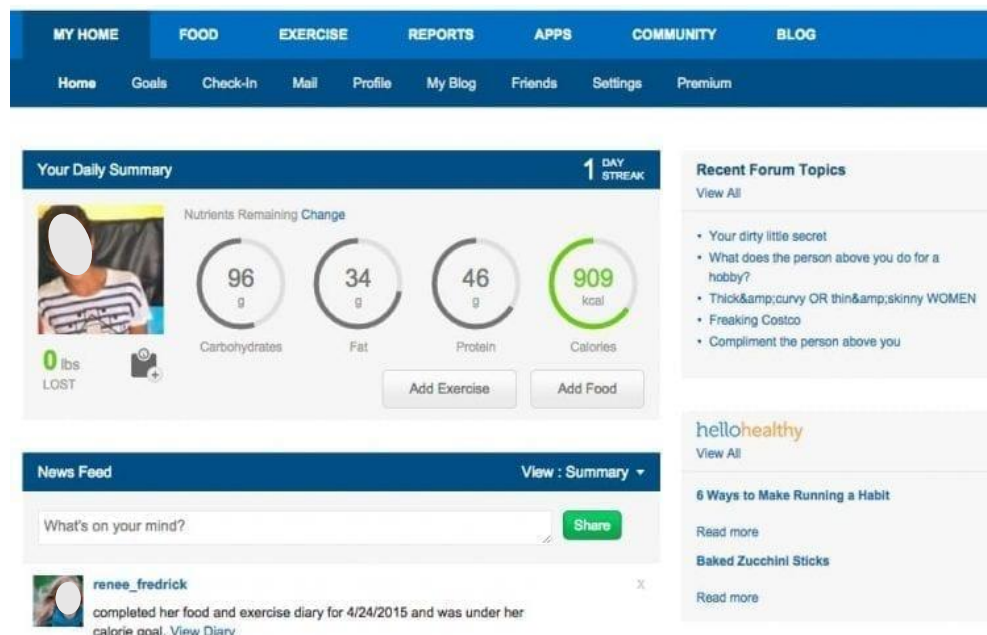


Figure 2: MyFitnessPal (Img source: [MyFitnessPal](#))

Some of the features inspired from MyFitnessPal for my project *fitness app* are:

- Water tracking
- Exercise logging feature
- Share feature
- Reports (Charts)

### 2.2.3 Nutritionist Pro (Diet Analysis Module)

Nutritionist Pro is a software application that offers features such as diet analysis, food labelling, menu creation, and nutrient tracking for the fitness industry. The Diet Analysis Module is a key feature of Nutritionist Pro software, it allows nutritionists, dietitians, coaches, and other food professionals to input client nutrient, exercise, and weight goals to track the information and analyse recipes. It offers automated tools, templates, built-in research, and standard guides, reports, and calculations which helps in client management.

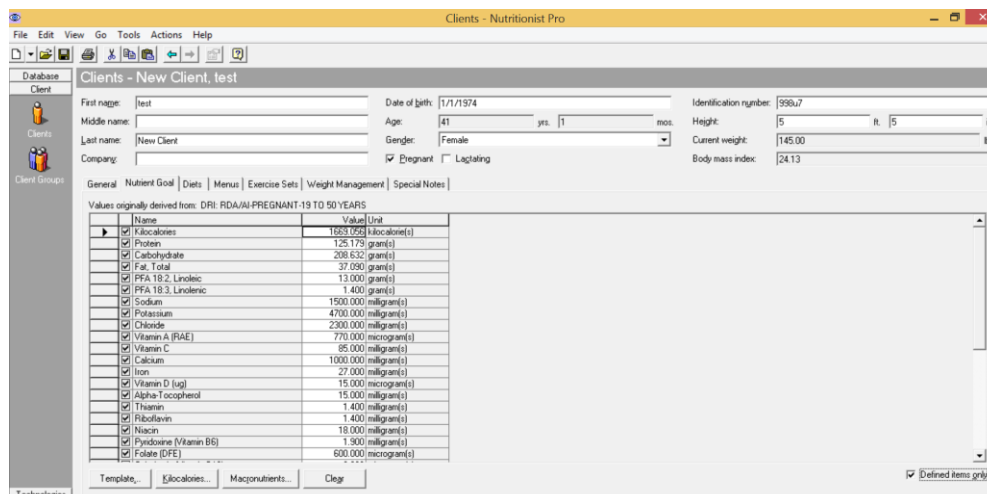


Figure 3:Diet Analysis Module (Img source: [Dite Analysis Module](#))

Some of the features inspired from Diet Analysis Module for my project *fitness app* are:

- Client Management
- Nutrition Tracking

## 2.3 Comparisons

A complete difference between all the above listed applications and my project *fitness app* is listed in a tabular format below:

S. N	Features	fitness app	Cronometer	MyFitnessPal	Nutritionist Pro
1	Exercise Tracking	✓	✓	✓	✓
2	Food Tracking	✓	✓	✓	✓
3	Goal setting	✓	✓	✓	✓
4	Water tracking	✓	✓	✓	✓
5	Custom Exercise	✓	✓	✓	✓
6	Custom Food	✓	✓	✓	✓
7	Barcode Scan (food)	✗	✗	✓	✗
8	Connect with People	✗	✗	✓	✗
9	Client Management	✓	✗	✗	✓
10	Report (Trends)	✓	✓	✓	✓
11	Share Data	✓	✗	✓	✗
12	Recipe Creation	✗	✗	✗	✓
13	Integration with wearables	✗	✓	✓	✗

**Table 1: Features Comparison**

### 3. Development

#### 3.1 Considered methodology

##### 3.1.1 Prototype Model

The prototyping model is a systems development method where a first version of the system is built, tested, and refined based on customer feedback. The system is reworked until an acceptable outcome is developed. This model is useful when client do not have complete idea of the product. Feedback from the client can help developers in development process. Types of prototyping models are Rapid Throwaway Prototyping, Evolutionary Prototyping, Incremental Prototyping, Extreme Prototyping. (geeksforgeeks, 2023)

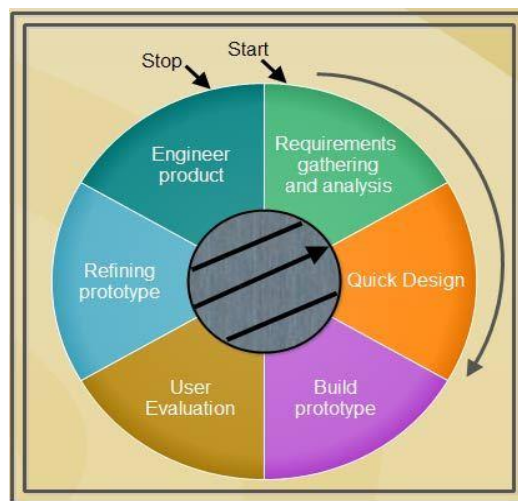


Figure 4: Prototype Model (Img source: [prototype](#))

##### Steps of Prototype model is:

- Step 1: Requirement Gathering and Analysis
- Step 2: Quick Design
- Step 3: Build a Prototype
- Step 4: Initial User Evaluation
- Step 5: Refining Prototype
- Step 6: Implement Product and Maintain



**Advantages of prototyping model are:**

- New requirements can be easily accommodated.
- Early error detection.
- Reduce risk, etc.

**Disadvantages of prototyping model are:**

- Costly and time consuming
- Difficult for developers to accommodate all the changes.
- Poor documentation of the system due to frequent changes, etc.

**Reason for not choosing Prototype model:**

- In prototype model the entire progress depends on the client feedback, the project *fitness app* does not have any client.
- Being single person project, it will be hard to develop frequent prototype.

### 3.1.2 Spiral Model

The Spiral Model is a Software Development Life Cycle (SDLC) model used for risk management which combines iterative development process and waterfall model. It is used for large, expensive, and complicated projects. This model allows to develop a prototype and release and refine the product in each phase of the spiral. (Tech Target, 2019)



Figure 5: Spiral Model (Img source: [spiral](#))

#### Phases of Spiral model is:

- 1. Planning:** Scope of the project is determined and plan for next iteration.
- 2. Risk Analysis:** Risk in the project is identified and evaluated.
- 3. Engineering:** The Software is developed.
- 4. Evaluation:** Check if the software meets customer's requirements.
- 5. Planning:** Next iteration begins with new planning.

**Advantages of the spiral model are:**

- Best development model for risk management because of risk analysis and risk handling at every phase.
- Spiral model helps to produce reliable and quality software due to multiple iterations in software development process.
- Change the requirements in the later phase, etc.

**Disadvantages of the spiral model are:**

- The Spiral Model is complex than other Software Development Life Cycle (SDLC) models.
- The Spiral Model is expensive, it is not suitable for small projects.
- The Spiral Model can be time consuming because it requires multiple evaluation, etc.

**Reason for not choosing spiral model:**

- Being single person project, it is hard to manage time for determining every risk of the project and developing prototype of the system.
- The project might go on infinite spiral.

### 3.1.3 Waterfall Model

The waterfall model is one of the first software development model, it is very important because all other software development life cycle (SDLC) models are based on waterfall model. In waterfall model the phases of the SDLC never overlap each other. It is useful where all the project requirements are known. (geeksforgeeks, 2023)

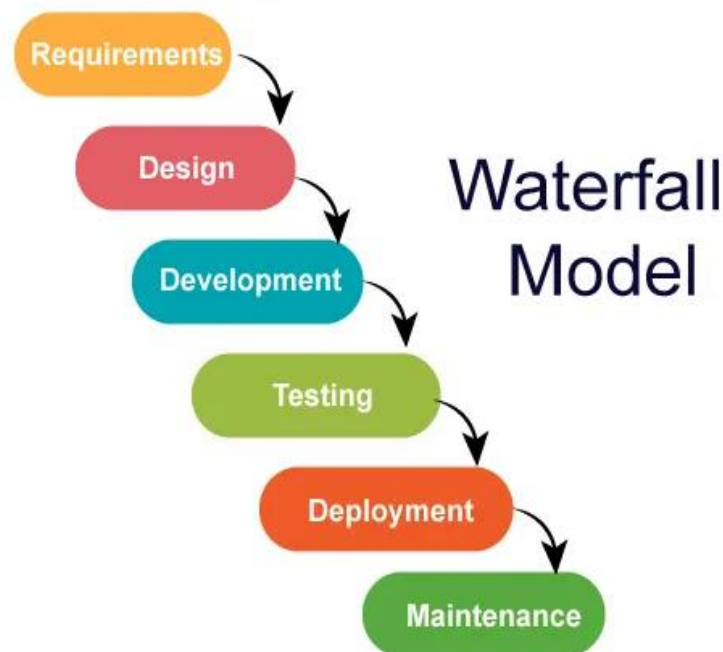


Figure 6: Waterfall Model (Img source: [waterfall model](#))

#### Phase of waterfall model is:

- 1. Requirement gathering:** Requirement is gathered from stakeholders, and it is analysed to understand the scope and objective of project.
- 2. Design:** The design document is created that contains software architecture, user interface, and system components.
- 3. Implementation:** The coding of the software is done based on design.

4. **Testing:** The whole software is tested.
5. **Deployment:** Software is deployed to production environment.
6. **Maintenance:** This phase involves fixing issues that comes after deploying.

**Advantages of waterfall model are:**

- Easy to understand.
- All processes are well documented.
- Waterfall model has clear and well-understood milestones, etc.

**Disadvantages of waterfall model are:**

- It is difficult to change any request made by customer after requirement phase.
- No feedback paths.
- It is not suitable for complex projects, etc.

**Reason for not choosing waterfall model:**

- Project requirements might change in the future.
- Testing is only done after complete development of the system which may result in many bugs.
- Should have complete picture of the whole system because in this model future changes are not accepted.

## 3.2 Selected methodology

### 3.2.1 Scrum methodology

Scrum is an agile development framework that simplify software development process through iterative and incremental processes, it offers adaptability, agility, and efficiency to ensure a continuous flow of value to clients throughout project development(S, 2023). Scrum helps to deliver new software capability every 2-4 week. The project will be broken into smaller parts called sprints; each sprint will have specific goals.

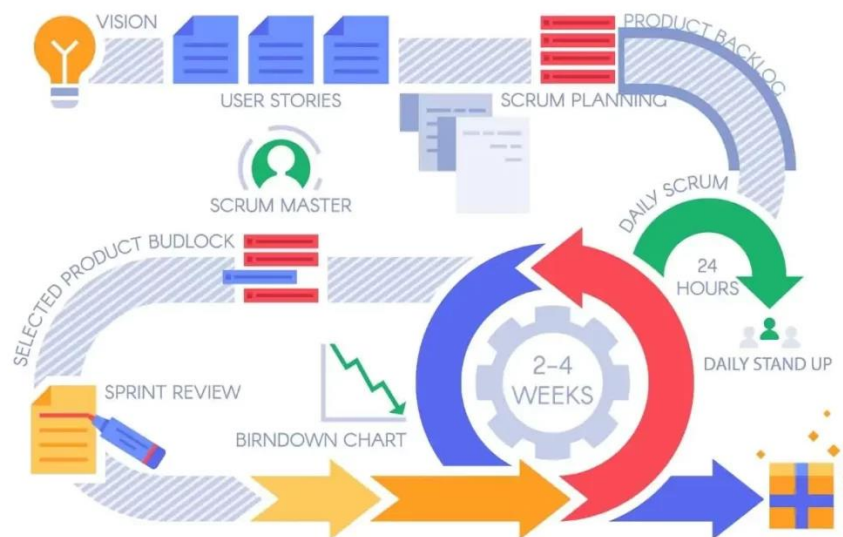


Figure 7: Scrum Methodology (Img source: [scrum](#))

#### Advantages of Scrum are:

- Help teams complete project quickly and efficiently.
- Large projects are divided into easily manageable sprints.
- The individual efforts of each team members are visible.
- Enables changes in the project easily, etc.

**Disadvantages of Scrum are:**

- Experienced team members required.
- Daily meeting might be frustration for team members.
- If team members leave the project in middle, it can cause negative impact on the project, etc.

**3.2.2 Reason for choosing scrum**

**General reason:** Scrum designed is to work with team, but for this project I am planning to do my project in scrum on analysis basis because scrum is most popular agile methodology, and it is widely used in recent years. I will be both product owner and development team.

**Reason for choosing SCRUM methodology for the project:**

- Changes in the project might occur after planning the project in scrum changes are accepted.
- Solo sprint meeting encourages self-reflection, which will increase efficiency.
- Iterative nature of scrum helps in identifying problem and receive feedback early, which lead in continuous improvement.
- Scrum's sprint will help in managing time and making sure the completion of project on time.
- Scrum's regular review and meeting helps in identifying risks early in the development process.
- The scrum will help in project documentation with product backlog and scrum artifacts.

### 3.2.2 Phases of Scrum and its implementation

**Initiation:** This is the first phase of the Scrum in which vision of the project is created. The phase prioritized on gathering information from client and creating documentation using it. Role to team members is also assigned in this phase (Indeed, 2022).

Task required:

- Create Project Vision
- Identify Scrum Master and Stakeholder(s)
- Create Prioritized Product Backlog
- Conduct Release Planning

**Plan and Estimate:** Plan for sprint is created, after completion of each sprint, it can be combined to project backlog. Relevant items from the backlog are selected and moved into sprint backlog. Planning and estimating follow an iterative cycle after breaking down the sprint. (Indeed, 2022)

Task required:

- Create User Stories
- Approve, Estimate, and Commit User Stories
- Create Tasks
- Estimate Tasks
- Create Sprint Backlog

**Implementation:** In this phase, team execute the planned sprint, backlog is maintained by removing completed items and new are assigned. Meeting is held to share project updates and discuss work plans. This process can be repeated several times until the project concludes. (Indeed, 2022)

Task required:



- Writing code and developing UI
- Performing Daily Scrum meeting
- Creating prototype using wireframe developed.
- Testing on the code

**Reviewing:** In this phase, project review is conducted gathering feedback at the end of each sprint. In the meeting, the overall execution of the Sprint is discussed and brainstorm ideas to improve the work completed. (Indeed, 2022)

Task required:

- Make changes in product backlog.
- Getting feedback from the client and supervisor

**Releasing:** In this phase final products to the stakeholders is delivered and bringing product to market.

Task required:

- Deploying the final product
- Testing in the live server

## 4. Project Progress

As mentioned above this project will be done by following scrum framework. The project was started by brainstorming about the features for the *fitness app* and discussing with supervisors and finalizing the project. System design (System architecture, Use Case diagram, Sequence and collaboration diagram, Flowchart, ERD) was completed before the development which is available in appendix ([System Design](#)).

### 4.1 Product backlog

The product backlog is the list of important tasks for development of the project.

PRODUCT BACKLOG :: fitness app					
User Story ID	User Story	Priority	Sprint	Task owner	Status
U.S 1	As a User and Coach I should be able to Create my account so, I could gain access to the application.	Must	1	Hrishik Sangroula	Ongoing
U.S 2	As a User and Coach I should be able to Login into my account so, I could use the application	Must	1	Hrishik Sangroula	Ongoing
U.S 3	As a User and Coach I should be able to recover my account when I forget my password.	Should	1	Hrishik Sangroula	Ongoing
U.S 4	As a User and Coach I should be able have access to dashbord so I can use features of the application.	Must	2	Hrishik Sangroula	Incomplete
U.S 5	As a User and Coach I should be able change my general information and logout.	Must	2	Hrishik Sangroula	Incomplete
U.S 6	As a User I should be able to create custom meals, so I can track my food habits.	Must	3	Hrishik Sangroula	Incomplete
U.S 7	As a User I should be able log my different meals, so I can track my food habits.	Must	4	Hrishik Sangroula	Incomplete
U.S 8	As a User I should be able log my different exercise, so I can track my exercies.	Must	5	Hrishik Sangroula	Incomplete
U.S 9	As a Coach I should be able create profile for my client, so I could track thei exercise and food habits.	Must	6	Hrishik Sangroula	Incomplete
U.S 10	As a User and Coach I should be able see data in different charts, so I could understand about my food and exercise habits and trends.	Must	7	Hrishik Sangroula	Incomplete
U.S 11	A a User I should be able shsre my data with coach , so coach could guide me.	Must	8	Hrishik Sangroula	Incomplete
U.S 12	As a Coach I should be able give feedback to shared data.	Must	8	Hrishik Sangroula	Incomplete
U.S 13	As a User and Coach I should be able create notes	Could	9	Hrishik Sangroula	Incomplete

Figure 8:Product Backlog

## 4.2 Development

As project follow scrum, all development is broken down in sprint. Sprint is a time where specific task must be completed and made ready for review.

### 4.2.1 Sprint 1 – Signup and Login

For the development Signup and Login were chosen as first task from product backlog. It was chosen as the first task because Sign up and Login are the entry point the system which will allows user and coach to use the application.

1. **Wireframe:** Wireframe of both Signup and Login were designed.

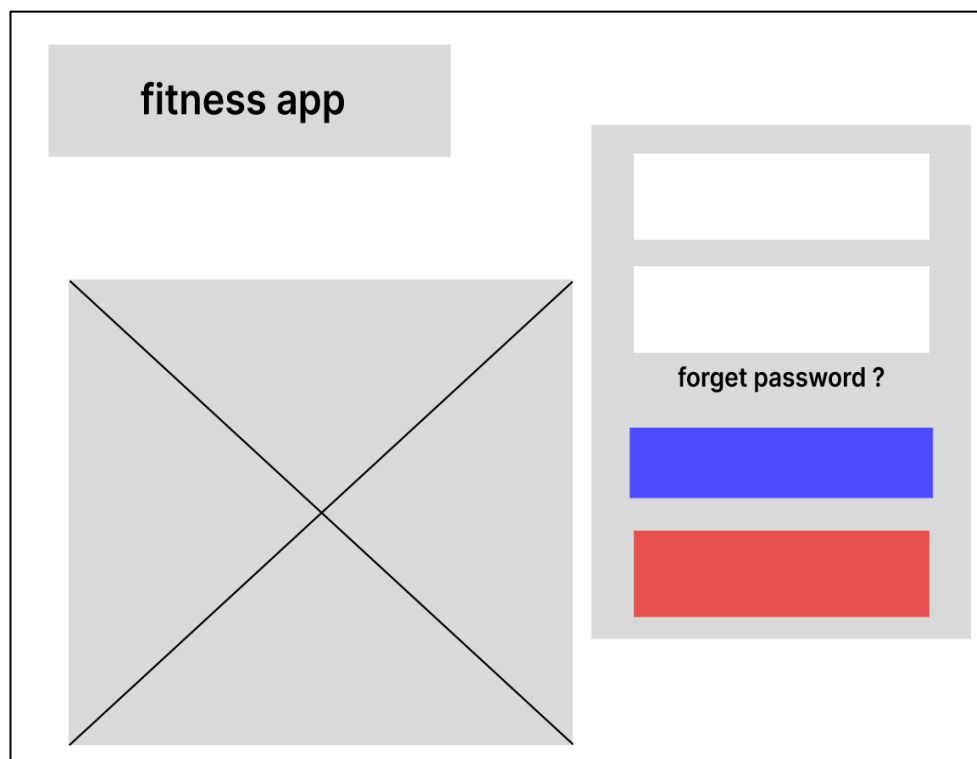


Figure 9: Login Wireframe

**fitness app**

**Create Your Account Free**

Full Name

Email

Passsword

Confirm password

Are yoy a coach?

☐

Yes

☐

No

**Profile Details**

Sex

☐

Male

☐

Female

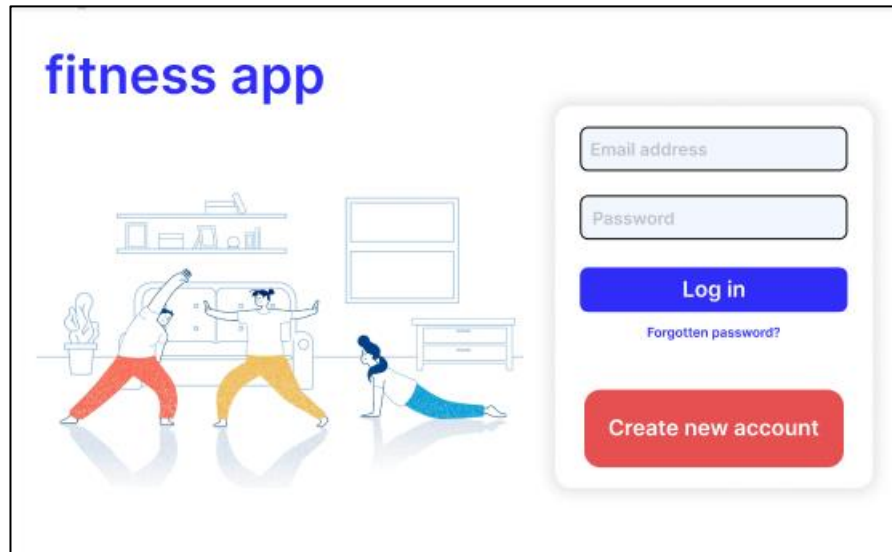
Height

Weight

Birthday

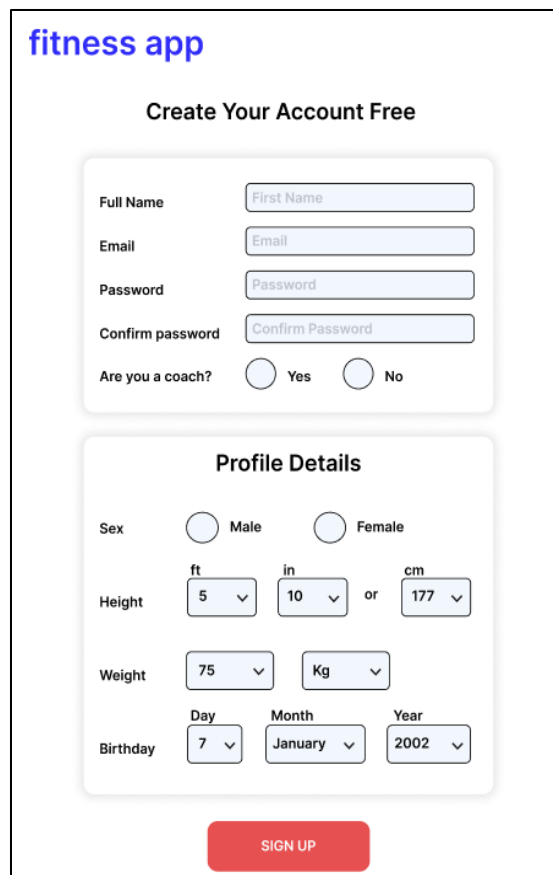
*Figure 10: Signup Wireframe*

2. **UI design:** UI was designed in *Figma* after wireframe was designed.



The login UI for the fitness app features a clean, modern design. On the left, there is an illustration of three people in a living room setting: a man in red pants doing a yoga pose, a man in yellow pants doing a yoga pose, and a woman in blue pants doing a yoga pose. Above them, the text "fitness app" is displayed in a bold, blue font. To the right of the illustration is a white rounded rectangle containing the login form. The form includes two input fields: "Email address" and "Password". Below these fields is a blue "Log in" button. Underneath the button is a link that says "Forgotten password?". At the bottom of the form is a red "Create new account" button.

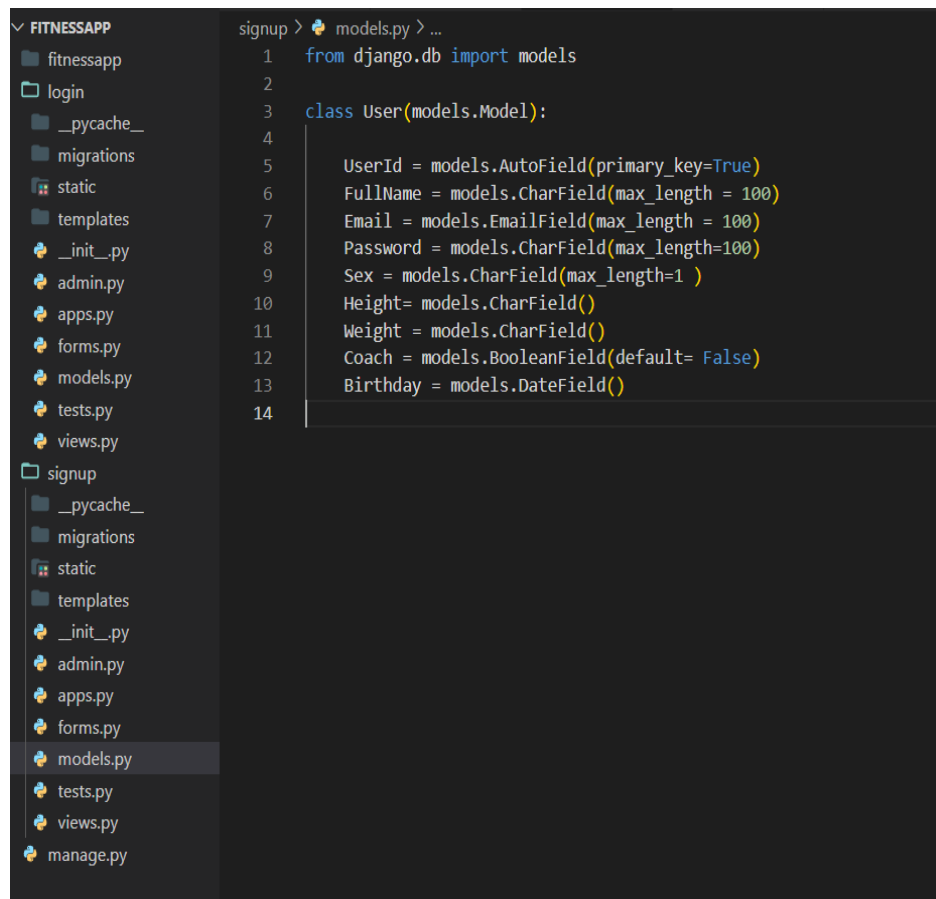
Figure 11: Login UI



The signup UI for the fitness app is designed to be user-friendly and informative. It features a white rounded rectangle with a blue "fitness app" header. Below the header is the text "Create Your Account Free". The form is divided into two main sections. The first section, "Create Your Account Free", contains four input fields: "Full Name" (with a sub-field for "First Name"), "Email", "Password", and "Confirm password". Below these fields is a radio button question "Are you a coach?" with "Yes" and "No" options. The second section, "Profile Details", contains several input fields: "Sex" (with "Male" and "Female" radio buttons), "Height" (with "ft", "in", and "cm" units and dropdown menus), "Weight" (with a dropdown menu and "Kg" unit), and "Birthday" (with "Day", "Month", and "Year" dropdown menus). At the bottom of the form is a red "SIGN UP" button.

Figure 12: Signup UI

- 3. Backend Development of Signup:** First backend code of Signup was written. Model class was written to store required information on database. Form fields and validation were done after model class. View function was written after that. Django Admin panel was also created.



```
signup > models.py > ...
1  from django.db import models
2
3  class User(models.Model):
4
5      UserId = models.AutoField(primary_key=True)
6      FullName = models.CharField(max_length = 100)
7      Email = models.EmailField(max_length = 100)
8      Password = models.CharField(max_length=100)
9      Sex = models.CharField(max_length=1 )
10     Height= models.CharField()
11     Weight = models.CharField()
12     Coach = models.BooleanField(default= False)
13     Birthday = models.DateField()
14
```

Figure 13: Signup Model Class

```

from django import forms
from .models import User

class UserRegistration(forms.ModelForm):

    confirm_password = forms.CharField(widget=forms.PasswordInput())
    inch = forms.IntegerField(required=False)
    centimeter = forms.FloatField(required=False)
    height = forms.IntegerField(required=False, widget=forms.NumberInput())
    weight_unit = forms.CharField(label='', widget=forms.Select(
        choices=[('Kg', 'Kg'), ('lbs', 'lbs')]))
    birth_date = forms.DateField(
        label='Birthday', required=True, widget=forms.SelectDateWidget(years=range(1900, 2101)))

    class Meta:
        model = User
        fields = ['FullName', 'Email', 'Password', 'confirm_password',
                  'Coach', 'Sex', 'height', 'inch', 'centimeter', 'Weight', 'weight_unit', 'birth_date']
        widgets = {
            'Password': forms.PasswordInput(),
            'Sex': forms.RadioSelect(choices=[
                ('M', 'Male'),
                ('F', 'Female'),
            ]),
            'Coach': forms.RadioSelect(choices=[(True, 'Yes'), (False, 'No')]),
        }

```

Figure 14: Signup forms.py 1

```

def clean(self):
    cleaned_data = super().clean()
    valpass = cleaned_data.get('Password')
    valrpass = cleaned_data.get('confirm_password')
    valheight = cleaned_data.get('height')

    if valpass and len(valpass) < 5:
        raise forms.ValidationError(
            "Password less than 5 character.")
    if valpass != valrpass:
        raise forms.ValidationError(
            "Password did not match.")

    if valheight and valheight > 9:
        raise forms.ValidationError(
            "Height in ft must be less than 10.")

    return cleaned_data

```

Figure 15: Signup forms.py 2

```

signup > views.py > sign_up
1  from django.shortcuts import render
2  from .forms import UserRegistration
3  from .models import User
4
5
6  def sign_up(request):
7      if request.method == 'POST':
8          fm = UserRegistration(request.POST)
9          if fm.is_valid():
10             full_name = fm.cleaned_data['FullName']
11             email = fm.cleaned_data['Email']
12             password = fm.cleaned_data['Password']
13             coach = fm.cleaned_data['Coach']
14             sex = fm.cleaned_data['Sex']
15             height = fm.cleaned_data['height']
16             inch = fm.cleaned_data['inch']
17             centi = fm.cleaned_data['centimeter']
18             weight = fm.cleaned_data['Weight']
19             w_unit = fm.cleaned_data['weight_unit']
20             dob = fm.cleaned_data['birth_date']
21             weight_w_unit = weight+" "+w_unit
22             total_height = str(height) + str(inch) + " " + str(centi)
23             signup = User(FullName = full_name, Email = email, Password = password,
24                           Coach = coach, Sex = sex,
25                           Height = total_height, Weight = weight_w_unit, Birthday = dob )
26             signup.save()
27             fm = UserRegistration()
28         else:
29             fm = UserRegistration()
30         return render(request, 'signup/signup.html', {'form': fm,})
31

```

Figure 16: Signup view

#### 4. Table in PostgreSQL and admin panel:

Data Output Messages Notifications

	Userid [PK] integer	FullName character varying (100)	Email character varying (100)	Password character varying (100)	Sex character varying (1)	Height character varying	Weight character varying	Coach boolean	Birthday date
1	1	hrishik	hrishiksangroul@gmail.com	@hrishik17	M	59 175.0	75 Kg	false	2002-01-01
2	2	Taushif Reza	Taushifreza@gmail.com	Godisgreat	M	510 178.0	50 Kg	false	1998-05-07

Figure 17: User Table in PostgreSQL

Action:  Go 0 of 2 selected

<input type="checkbox"/>	USERID	FULLNAME	EMAIL	PASSWORD	SEX	HEIGHT	WEIGHT	BIRTHDAY	COACH
<input type="checkbox"/>	2	Taushif Reza	Taushifreza@gmail.com	Godisgreat	M	510 178.0	50 Kg	May 7, 1998	✖
<input type="checkbox"/>	1	hrishik	hrishiksangroul@gmail.com	@hrishik17	M	59 175.0	75 Kg	Jan. 1, 2002	✖

2 users

Figure 18: User table in admin panel



5. **Backend Development of Login:** Form was created for email and password. View function was written after that.

```
login > forms.py > ...
1  from django import forms
2
3  class UserLoginForm(forms.Form):
4      email = forms.EmailField()
5      password = forms.CharField(widget=forms.PasswordInput())
6
7
```

Figure 19: Login form.py

```
login > views.py > user_dashboard
1  from django.shortcuts import render, HttpResponseRedirect
2  from .forms import UserLoginForm
3  from signup.models import User
4  from django.contrib import messages
5
6  def user_login(request):
7      if request.method == "POST":
8          fm = UserLoginForm( data = request.POST)
9          if fm.is_valid():
10             email = fm.cleaned_data['email']
11             password = fm.cleaned_data['password']
12             users= User.objects.all()
13             list_emails= []
14             list_pass = []
15             for user in users:
16                 list_emails.append(user.Email)
17                 list_pass.append(user.Password)
18             if email in list_emails and password in list_pass:
19                 return HttpResponseRedirect('/dashboard/')
20             else:
21                 fm = UserLoginForm()
22         else:
23             fm = UserLoginForm()
24
25         return render(request, 'login/login.html', {'form': fm, })
26
27
28 def user_dashboard (request):
29     return render(request, 'login/dashboard.html')
```

Figure 20: Login view

### 4.3 Gantt Chart

Changes were made to the previous Gantt chart to follow scrum methodology strictly. Previous version did not follow scrum, it didn't have any sprint.

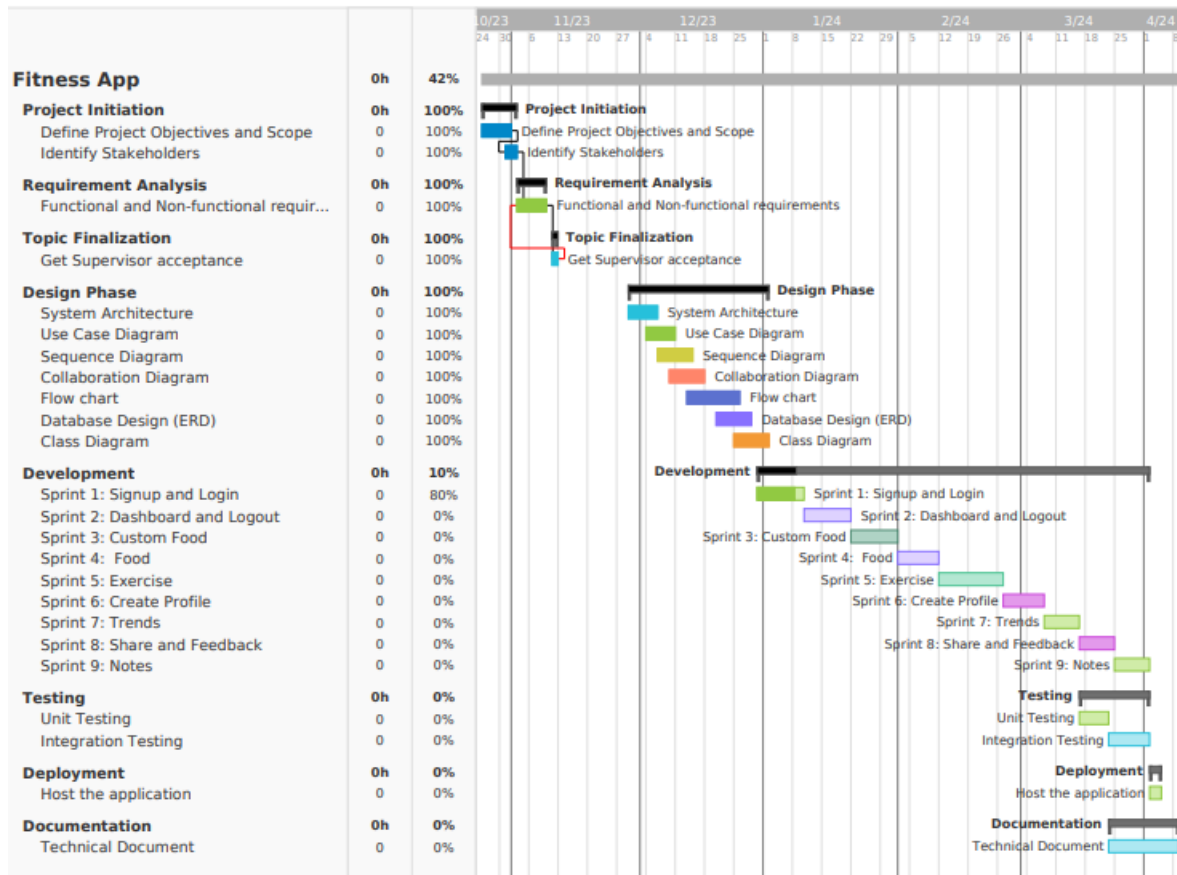


Figure 21:Gantt Chart

## 4.4 Work Break Down Structure

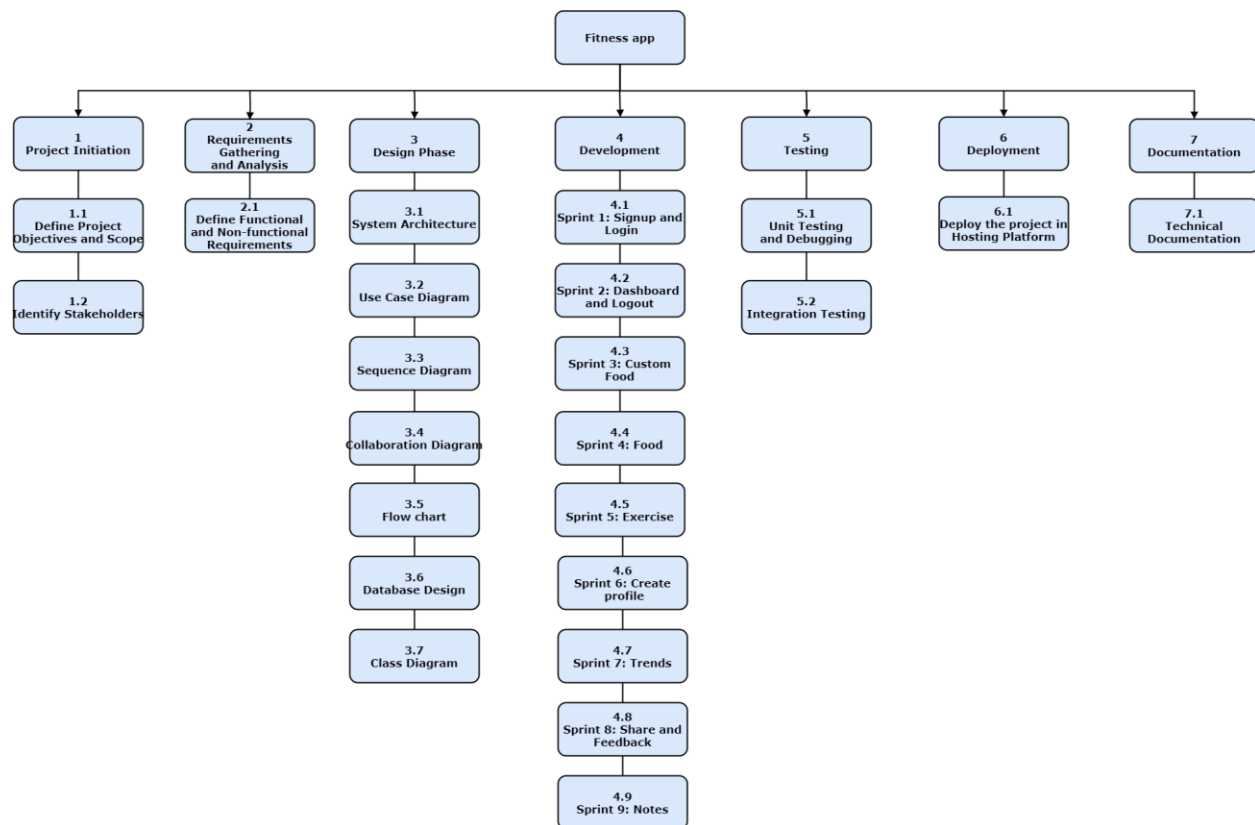


Figure 22: Work Break down Structure

Changes were made to Work Breakdown Structure (WBS) after following scrum methodology strictly (Old WBS Appendix). The Work Breakdown Structure (WBS) for the development of a fitness web application involves a systematic breakdown of the project into manageable components. The project starts by defining project's objectives and scope and identifying stakeholders for the project. Then, research and requirement analysis are done focusing on features of the web application. Following this, in the design phase various diagrams like Flow chart, System architecture, UseCase diagram, Sequence and collaboration diagram, ERD, and Class Diagram and were done. Finally, Backend and Frontend with UI/UX development will be done in 8 sprints after design phase. After development is completed, testing will be done. After testing documentation of the project will be created. Finally, the web application will be deployed. The project will be completed in 4-5 months with most the time involving in Design, Development and Testing.

## 5. Progress Review

### 5.1 Progress Table

The progress table helps to determine the actual progress of the project.

S. N	Tasks	Status	Progress
1.	Topic Selection	Completed	100%
2.	Feasibility Study	Completed	100%
3.	Research on similar project	Completed	100%
4.	Project approval from supervisors	Completed	100%
5.	Completing Proposal	Completed	100%
6.	Develop System Architecture Diagram	Completed	100%
7.	Develop Use Case Diagram	Completed	100%
8.	Develop Sequence Diagram	Completed	100%
9.	Develop Collaboration Diagram	Completed	100%
10.	Develop Flow Chart	Completed	100%
11.	Develop Initial Entity Relation Diagram	Completed	100%
12.	Develop Class Diagram	Completed	100%
13.	Develop SRS Document	Completed	100%
14.	Sprint 1: Signup and Login	Almost Completed	80%
15.	Sprint 2: Dashboard and Logout	Not Completed	0%
16.	Sprint 3: Custom Food	Not Completed	0%
17.	Sprint 4: Food	Not Completed	0%
18.	Sprint 5: Exercise	Not Completed	0%
19.	Sprint 6: Create Client	Not Completed	0%
20.	Sprint 7: Trends	Not Completed	0%
21.	Sprint 8: Share and Feedback	Not Completed	0%
23.	Testing the project	Not Completed	0%
24.	Deploy project in hosting platform	Not Completed	0%
25.	Do documentation	Not Completed	0%
26.	Submit the project	Not Completed	0%

Table 2: Progress Table

## 5.2 Further Work

Further work includes the remain tasks to be done following the time estimated in the Gantt Chart.

### **I. Sprint 1: Signup and Login**

80 % of the work of this sprint is already complete. Now, frontend of the Login and Signup page is to be made. Backend of forgotten password is to be made. Function based view will be turned into Class based view.

### **II. Sprint 2: Dashboard and Logout**

Defining the scope and objectives of Dashboard and Logout. Outlining tasks, estimate time for implementation for each task. In Dashboard general information about food and exercise will be shown and button will be created for user to logout. Evaluating the implementation of Dashboard and Logout features and performing testing.

### **III. Sprint 3: Custom Food**

Defining the scope and objectives of Custom Food features. Outlining tasks, estimate time for implementation for each task. Form will be made for this feature where users and coaches can fill the data like amount, calories, nutritional information about food. Evaluating the implementation of Custom food features and performing testing.

### **IV. Sprint 4: Food**

Defining the scope and objectives of Food features. Outlining tasks, estimate time for implementation for each task. Third party food API will be used for prepopulating the food database, form will be created where users and coach can put food data to log it. Evaluating the implementation of Custom food features and performing testing.

**V. Sprint 5: Exercise**

Defining the scope and objectives of Exercise features. Outlining tasks, estimate time for implementation for each task. Form will be created where users and coach can put exercise data like name, types, sets, reps, weights, targeted muscle groups, and equipment used systematically. Evaluating the implementation of exercise features and performing testing.

**VI. Sprint 6: Create Client**

Defining the scope and objectives of Create Client features. Outlining tasks, estimate time for implementation for each task. Form will be created where coach can put information of client and coach can put exercise and food data of client. Evaluating the implementation of Create Client features and performing testing.

**VII. Sprint 7: Trends**

Defining the scope and objectives of trends features. Outlining tasks, estimate time for implementation for each task. Chart.js will be used to create charts with user's and client's data. Evaluating the implementation of Trends features and performing testing.

**VIII. Sprint 8: Share and Feedback**

Defining the scope and objectives of Share and Feedback features. Outlining tasks, estimate time for implementation for each task. Form will be created where user can enter coach email and send their data. Coach will be able to receive data and give feedback through form. Evaluating the implementation of Share and Feedback features and performing testing.

**IX. Testing the project**

Unit testing and integration testing will be done, Test cases will be used to test the system whether the system has bug or not. If any error or bug is found, then it will be solved immediately.

**X. Deploy project in hosting platform**

The project will either be deployed in AWS or Microsoft azure.

**XI. Do documentation**

Final Year Project final report will be written and technical report will be written.

### 5.3 Project risks, threats, and contingency plans

**I. Threat:** Malicious attacks targeting users' data.

**Risk:** Unauthorized access or leak of user's data.

**Contingency plans:** Using encryption protocols and robust authentication measures.

**II. Threats:** Unexpected incidents like system failures or errors.

**Risks:** Unexpected incidents leading to data loss or data corruption.

**Contingency plans:** Implementing regular backups and adapting best practices  
in data storage and management to minimize the risk of data loss.

**III. Threats:** Technical challenges and complexities while developing multiple functionalities.

**Risks:** Not completing project on time.

**Contingency plans:** Breaking down the project in multiple modules, conducting regular  
progress evaluations and learning and solving technical challenges and  
complexities.

**IV. Threats:** UI not optimized for various devices, browsers, or screen sizes.

**Risks:** Inconsistent user experiences, functionality issues, and decrease accessibility.

**Contingency plans:** Using responsive design principles and testing UI across multiple  
devices and browsers.

**V. Threats:** Inadequate technical research on suitable tools, or technologies.

**Risks:** Choosing unsuitable or outdated technologies.

**Contingency plans:** Getting help from supervisors and comparing available technologies  
and tools.



## 5.4 Project Boundaries and contingency plans

### I. Project Timeline

**Boundary:** Limited time to the project.

**Contingency plans:** Making sure completing every sprint on time and keeping track of progress.

### II. Technology stack

**Boundary:** Tools and technology used in the project.

**Contingency plans:** Making sure alternative is identified, which can be used if chosen technology gives unexpected challenges.

### III. Resource Constraints

**Boundary:** Limited budget or human resources.

**Contingency plans:** Making sure all the project features can be done singly and making sure, project can be achieved under the budget.

### IV. Quality Standards

**Boundary:** Fast, safe and error free product.

**Contingency plans:** Using industry standards protocols. Testing throughout the development process.

## 6. Bibliography

Chang, B. P. W. T. L. B. Y. & S. C. B., 2017. Which Factors Are Associated with Monitoring Goal Progress?. *Frontiers in psychology*, 8(434), p. 10.

Harkin, B. W. T. L. C. B. P. P. A. C. M. K. I. B. Y. & S. P., 2016. Does monitoring goal progress promote goal attainment? A meta-analysis of the experimental evidence.. *Psychological bulletin*, 142(2), p. 198 –229.

## 7. References

ecommercedb, 2022. *Fitness Equipment eCommerce Market in Nepal - Data & Trends / ECDB.com*. [Online]

Available at: <https://ecommercedb.com/markets/np/fitness-equipment>  
[Accessed 30 December 2023].

Forbes, 2023. *Obesity Statistics In 2023 – Forbes Health*. [Online]

Available at: <https://www.forbes.com/health/body/obesity-statistics/#:~:text=Worldwide%2C%20more%20than%201%20billion,million%20children%2C%20according%20to%20WHO.>  
[Accessed 22 11 2023].

geeksforgeeks, 2023. *Software Engineering / Prototyping Model - GeeksforGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/>  
[Accessed 31 December 2023].

geeksforgeeks, 2023. *Waterfall Model - Software Engineering - GeeksforGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/waterfall-model/>  
[Accessed 31 December 2023].

Grand View Research , 2022. *Fitness App Market Size, Share & Trends Analysis Report By Type (Exercise & Weight Loss, Diet & Nutrition, Activity Tracking), By Platform (Android, iOS), By Device, By Region, And Segment Forecasts, 2022 - 2030*, s.l.: Grand View Research.

Indeed, 2022. *5 Scrum Phases for Project Management (Plus Benefits) / Indeed.com*. [Online]

Available at: <https://www.indeed.com/career-advice/career-development/scrum-phases>  
[Accessed 31 December 2023].

S, B., 2023. *What Is Scrum Methodology? & Scrum Project Management*. [Online]

Available at: <https://www.nimblework.com/agile/scrum-methodology/>  
[Accessed 31 December 2023].

Sharma, N., 2023. *Fitness Industry Statistics 2023 : Breaking Down The Number*. [Online] Available at: <https://www.nimbleappgenie.com/blogs/fitness-industry-statistics/#::~text=The%20number%20of%20people%20who,increase%20by%200.6%25%20in%202023.>

[Accessed 30 December 2023].

Spraul, T., 2023. *Gym Membership Statistics in 2023 (Big List) | Exercise.com*. [Online] Available at: <https://www.exercise.com/grow/gym-membership-statistics/>

[Accessed 22 11 2023].

Tech Target, 2019. *What is Spiral Model and How is it Used?*. [Online] Available at: <https://www.techtarget.com/searchsoftwarequality/definition/spiral-model>

[Accessed 31 Dercember 2023].

World Health Organization, n.d. *Physical inactivity*. [Online] Available at: <https://www.who.int/data/gho/indicator-metadata-registry/imr-details/3416>

[Accessed 21 11 2023].

## **8. Appendix**

### **8.1 SRS Document**

---

# **Software Requirements Specification**

**for**

# **fitness app**

**Version 1.0 approved**

**Prepared by Student Name**

**Islington College**

**2024-01-02**

### 8.1.1 Introduction

#### Purpose of Document

The SRS document shows all the requirements of the *fitness app* web application. This document helps to list all the requirements needed in an application with its features. It also includes all the requirements like Hardware and Software needed to develop the application.

#### Document Convention

The fitness app Web Application Software Requirements Specification (SRS) follows the IEEE standards for documenting software requirements.

#### Intended Audience and Reading Suggestions

- Programmers who are interested in working on the project by further developing it or fix existing bugs.

#### Product Scope

The main objective of the application is to develop web application that integrates workout tracking, nutrition tracking, progress visualization and coach management, with an aim of motivating users to attain their fitness goal and help coaches to manage multiple clients effectively. The project will be completed in 4 months. The web application will have features:

- Workout logging
- Food tracking
- Visualize data in different charts.
- Client management for coach

## References

IEEE Template for System Requirement Specification Documents

Link: <https://goo.gl/nsUFwy>

## 8.1.2 Overall Description

### Project Perspective

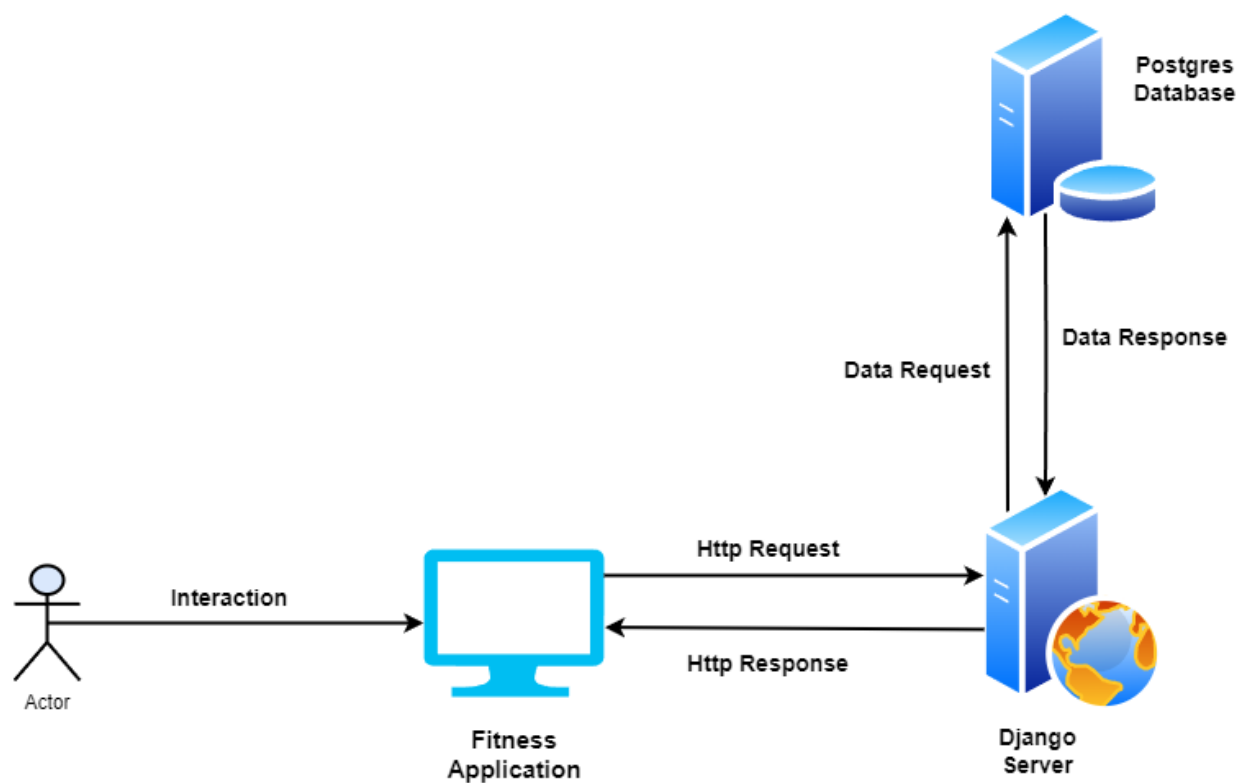


Figure 23: System Architecture SRS

The *fitness app* web application is developed to motivate people to achieve their fitness goal, it is a user-friendly application to track fitness activities, manage nutrition, visualize their progress and for coaches it will help to manage clients efficiently. The final product will be reliable, user-friendly, and effective web application.

**Product Function**

- **Login:** Allows user to enter email address and password to access their account.  
Reset forgotten password.
- **Signup:** Allows to enter personal details like name, email, password and personal details to create account to use the web application.  
  
Allows coach to tick if they are coach. Coach don't have to enter personal details.
- **Dashboard:** Shows general information.
- **Exercise:** Allows to put data to log workouts.
- **Food:** Allows to put data to log Meals.
- **Trends:** Shows visual representation of data.
- **General setting:** Shows user information about account.
- **Share:** Allows to share information with coach.

*Detail Product Functions of Exercise, Food, Trends, General setting and Share will be included in final FYP report SRS.*

**User Classes and characteristics****Users:**

- Log daily workouts systematically.
- Monitor and analyse progress.
- Track daily meals.
- Share information with coach.

**Coach:**

- Create client profile to track client's workouts and nutrition.
- Give feedback to shared client's information.
- Efficiently manage and organize client information.

**Operating environment**

- The web application will run on all modern browser like Chrome, Mozilla, Safari, etc.
- The web application will be responsive for most the devices with different screen dimensions.
- The web application will be secured over internet using TLS/SSL protocol.

**Design and Implementation constraints**

- The web application will be developed using SCRUM methodology.
- The web application will have Model View Template (MVT) architecture.
- The web application will be developed using Django for backend.
- The data visualization related to workouts and meals will be shown with Chart.js.



## User Documentation

The user guide will be provided to users within the web application which, will guide users how to use the web application and its features.

## Assumptions and Dependencies

- Users and coaches will provide correct data related to workouts and meals.
- Edamam food API will be used to provides nutritional information for various food items.

### 8.1.3 External Interface Requirements

#### User Interface

- Login Page

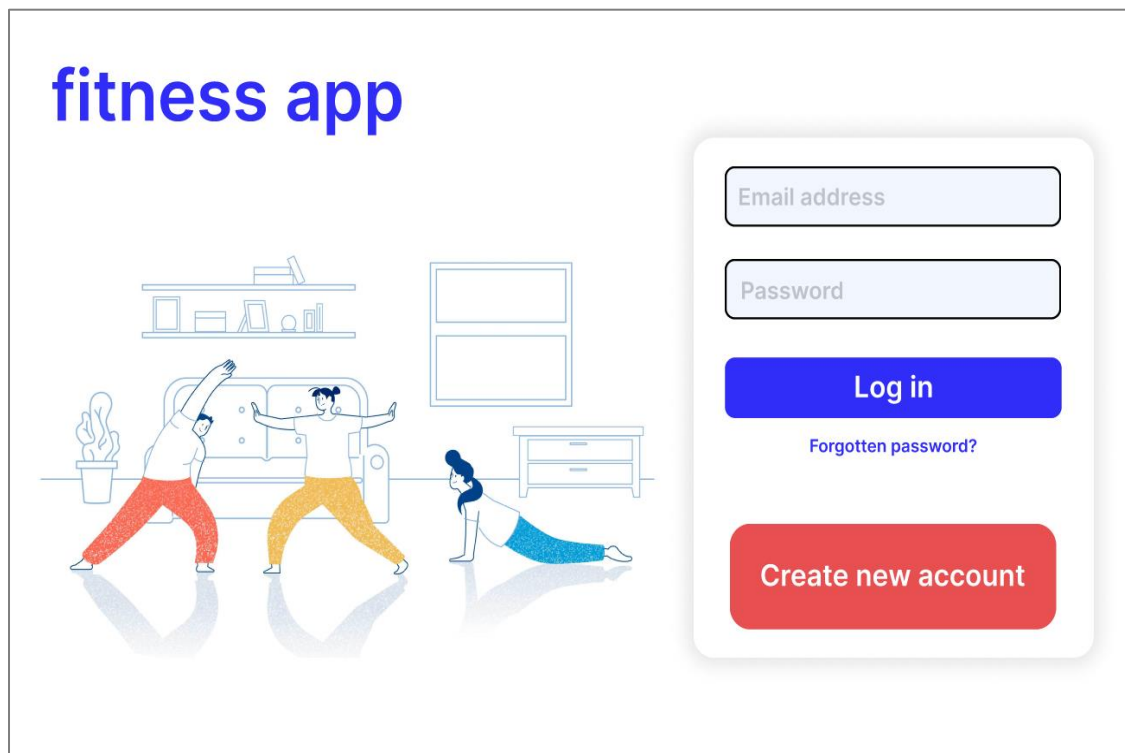


Figure 24: Login UI SRS

- Sign Up Page

## fitness app

### Create Your Account Free

First Name

Email

Password

Confirm password

Are you a coach?

☐ Yes ☐ No

### Profile Details

Sex

☐ Male ☐ Female

Height

ft

5

▼

in

10

▼

or

cm

177

▼

Weight

75

▼

Kg

▼

Birthday

Day

7

▼

Month

January

▼

Year

2002

▼

SIGN UP

Figure 25: Signup UI SRS

*Other UI pages will be included in final FYP report.*

**Hardware Interfaces**

The minimum hardware requirement of fitness app web application is a device such as desktops, laptops, tablets, and smartphones.

**Software Interfaces**

The minimum software requirement of fitness app web application is a web browser which supports JavaScript like Chrome, Safari, Edge, etc.

**Communications Interfaces**

The fitness app web application requires an internet connection to use its features. Email is also required to access its features.

#### 8.1.4 System Features

- **Exercise:** Users will be able to log their daily workouts, including exercises details like name, types, sets, reps, weights, targeted muscle groups, and equipment used systematically.
- **Food:** Users will be able to input daily meals, including food items, portion sizes, and nutritional information, offering either pre-populated nutritional data or the option to add custom food items.
- **Trends:** Users will be able to set their workouts goals and nutritional goals. Progress analytics will be a key feature of the web application, where users can view visual representations of their fitness trends using line graphs, bar charts, and pie charts, along with graphical displays showing their progress over time.
- **Coach management:** There will be separate interface for coaches which will help coaches to manage multiple clients' data, view workout logs, nutrition records, and progress analytics, ensuring personalized guidance and feedback.

## Functional Requirements

### Sign-Up

Two types of users; user and coach can create their account.

Req.ID	Requirement Information		Priority	Complexity
FR.01	Users can create the account by filling the form.		Most	Medium
	System Requirement			
	SR.01	Users will get Login page when they first access the application.		
	SR.02	Users click the signup button on the login page.		
	SR.03	Users fill the form.		
	SR.04	If users tick yes on, are you coach? they don't have to fill personal details form. If they tick no, they have fill personal details form.		
	SR.05	Users will submit the form.		
	SR.06	The system verifies all the fields and throw error if data cannot be validated.		
	SR.07	If data are correct system will add user in database.		
	SR.08	The system shows the message of whether user account creation was successful or not.		

Table 3: Signup Functional Requirements

## Login

Two types of users; user and coach can access their account.

Req.ID	Requirement Information		Priority	Complexity
FR.02	All users can access Login page.		Most	Medium
	System Requirement			
	SR.01	Users will get Login page when they first access the application.		
	SR.02	Users must provide their information; Email address and Password.		
	SR.03	Users will click on log in button.		
	SR.04	The system verifies all the fields and throw error if data cannot be validated.		
	SR.05	The system checks whether submitted Email Address and password matched with data in database.		
	SR.06	The system throws error if login credentials were not correct.		
	SR.07	The system redirect user to respective dashboard if user credentials were correct.		

**Table 4: Login Functional Requirements**

**Exercise**

Two types of users; user and coach can access Exercise feature. User can log their daily workouts with different exercises. Coach can make log of different exercise of their clients. Both users can update exercise data.

Req.ID	Requirement Information		Priority	Complexity
FR.03	All users can access Exercise feature.		Most	Hard
	System Requirement			
	SR.01	Users will access exercise feature from menu.		
	SR.02	Users will either choose from inbuilt list of exercises or create their custom exercise.		
	SR.03	Users will fill the exercise form and submit the form.		
	SR.04	The system adds the data in database.		

**Table 5: Exercise Functional Requirements**

**Food**

Two types of users; user and coach can access Food feature. User can log their daily meals with different foods. Coach can make log of different meals of their clients. Both users can update food data.

Req.ID	Requirement Information		Priority	Complexity
FR.04	All users can access Food feature.		Most	Hard
	System Requirement			
	SR.01	Users will access Food feature from menu.		
	SR.02	Users will either choose from inbuilt list of foods or create their custom food.		
	SR.03	Users will fill the food form and submit the form.		
	SR.04	The system adds the data in database.		

**Table 6: Food Functional Requirements**



## Trends

Two types of users; user and coach can Trends features. User can view visual representations of their fitness trends using different charts. Coach can view client's data in visual representation.

Req.ID	Requirement Information		Priority	Complexity
FR.05	All users can access Trends feature.		Most	Hard
	System Requirement			
	SR.01	Users will access Trends feature from menu.		
	SR.02	Users will select type of chart to see visual representation of different exercise and nutrition data.		
	SR.03	The system will access user’s data from database.		
	SR.04	The system generate chart from accessed data and represent it.		

**Table 7: Trends Functional Requirements**

**Share**

User can access Share feature. Coach doesn't have access to this feature. User can share their data to their coach.

Req.ID	Requirement Information		Priority	Complexity
FR.05	User can access Share feature.		Most	Medium
	System Requirement			
	SR.01	User will access share feature from menu.		
	SR.02	Users will enter the email address of coach to share their data.		
	SR.03	The system will verify the email address.		
	SR.04	The system will show error message if email address is not in the database.		
	SR.05	The system will allow user to click on share button if email address exist.		
	SR.06	The system will show shared successfully message.		

**Table 8: Share Functional Requirements**

### Coach Management

Coach have access to this feature. It allows coaches to create multiple client's profile and manage multiple clients' data.

Req.ID	Requirement Information		Priority	Complexity
FR.06	Coach have access to Coach Management feature.		Most	Hard
	System Requirement			
	SR.01	Coach can create client profile and add food and exercise data.		
	SR.02	The system will add data in database.		
	SR.03	The system will show all the shared information.		
	SR.04	Coach can view shared information and give feedback to user.		
	SR.05	The system will store feedback in database.		

**Table 9: Coach Management Functional Requirements**

### 8.1.5 Other Non-Functional Requirements

#### Performance Requirements

Req.ID	Performance Requirement s	Priority
PR.01	The web application should response to user within 2 seconds for responsive user experience.	Should
PR.02	The web application should load workout logs, nutritional information, and trends within 2 seconds.	Should
PR.03	The web application should perform read write operation from database within 2 seconds.	Should

Table 10: Performance Non-FR

#### Safety Requirements

Req.ID	Safety Requirements	Priority
SR.01	User data, including personal details and fitness information, must be stored securely.	Should
SR.02	Regular backups of user data.	Could

Table 11: Safety Non-FR

**Security Requirements**

Req.ID	Security Requirements	Priority
SR.01	User authentication should be implemented securely.	Should
SR.02	The application must use SSL to secure communication between the user's browser and the web server.	Should
SR.03	Any secured content could be communicated in encrypted format	Could

**Table 12: Security Non-FR****Software Quality Attributes**

- The web application code should be well-documented and maintained for future updates.
- The user Interface should be responsive to all the devices.
- The web application should be reliable, minimizing errors and downtimes.

## 8.2 Design

[Go to top](#): Project Progress

### 8.2.1 Mind Map



Figure 26: Mind Map

## 8.2.2 System Architecture

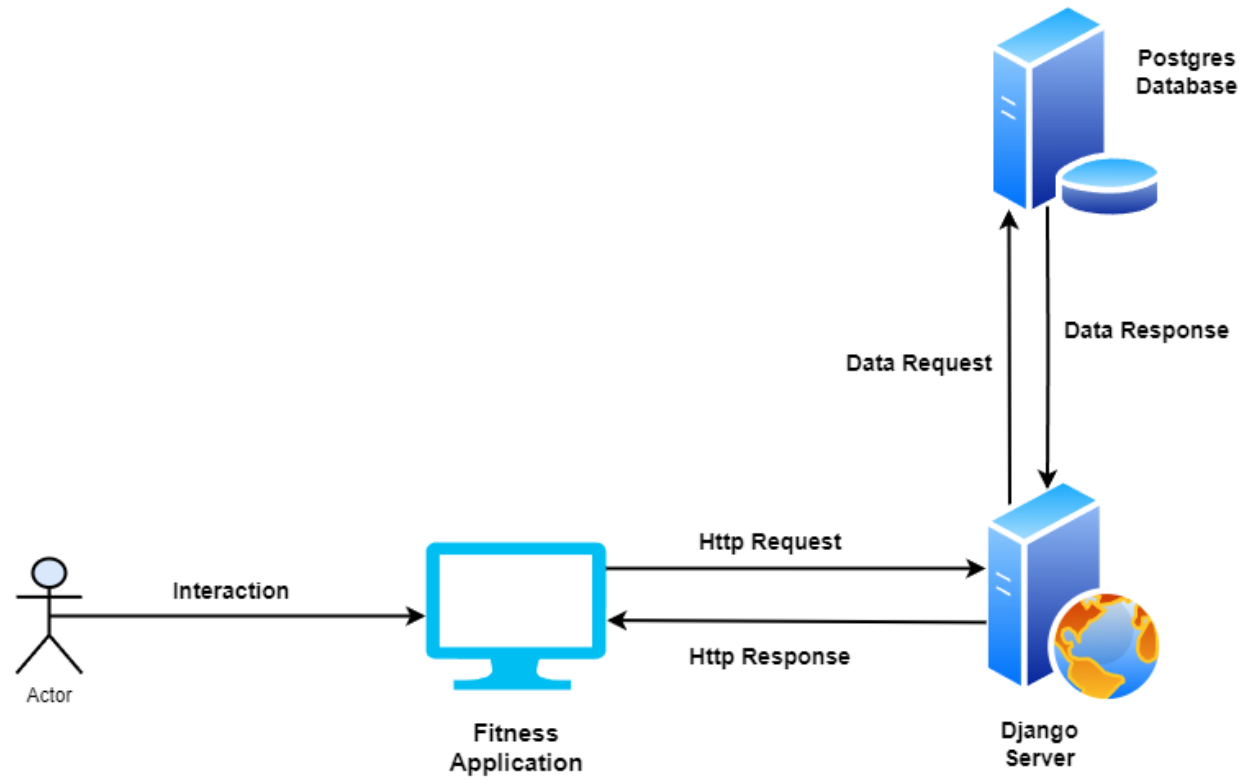


Figure 27: System Architecture

### 8.2.3 Use Case Diagram

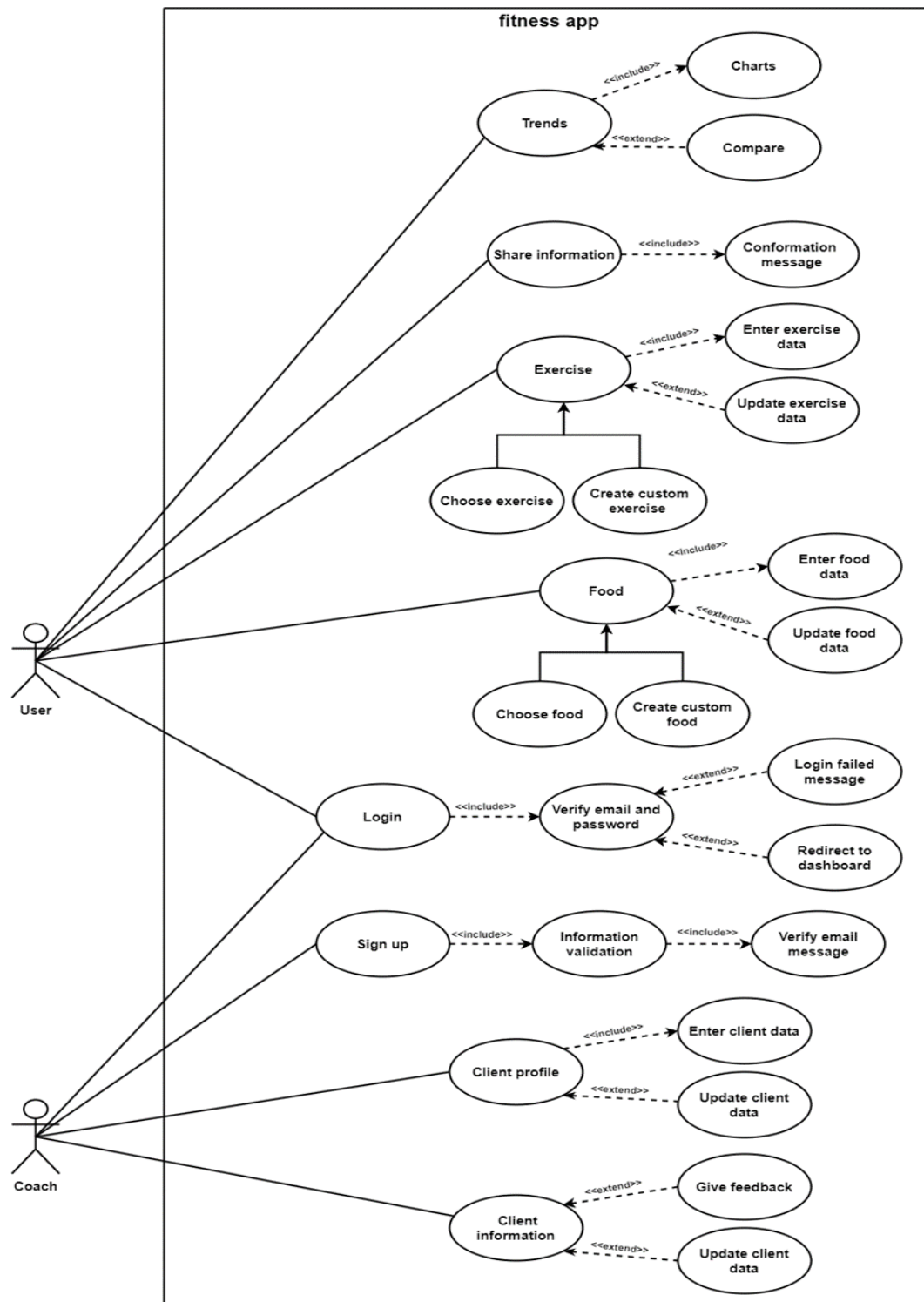


Figure 28: Use Case Diagram



## 8.2.4 Sequence Diagram

### 1. Login Sequence Diagram

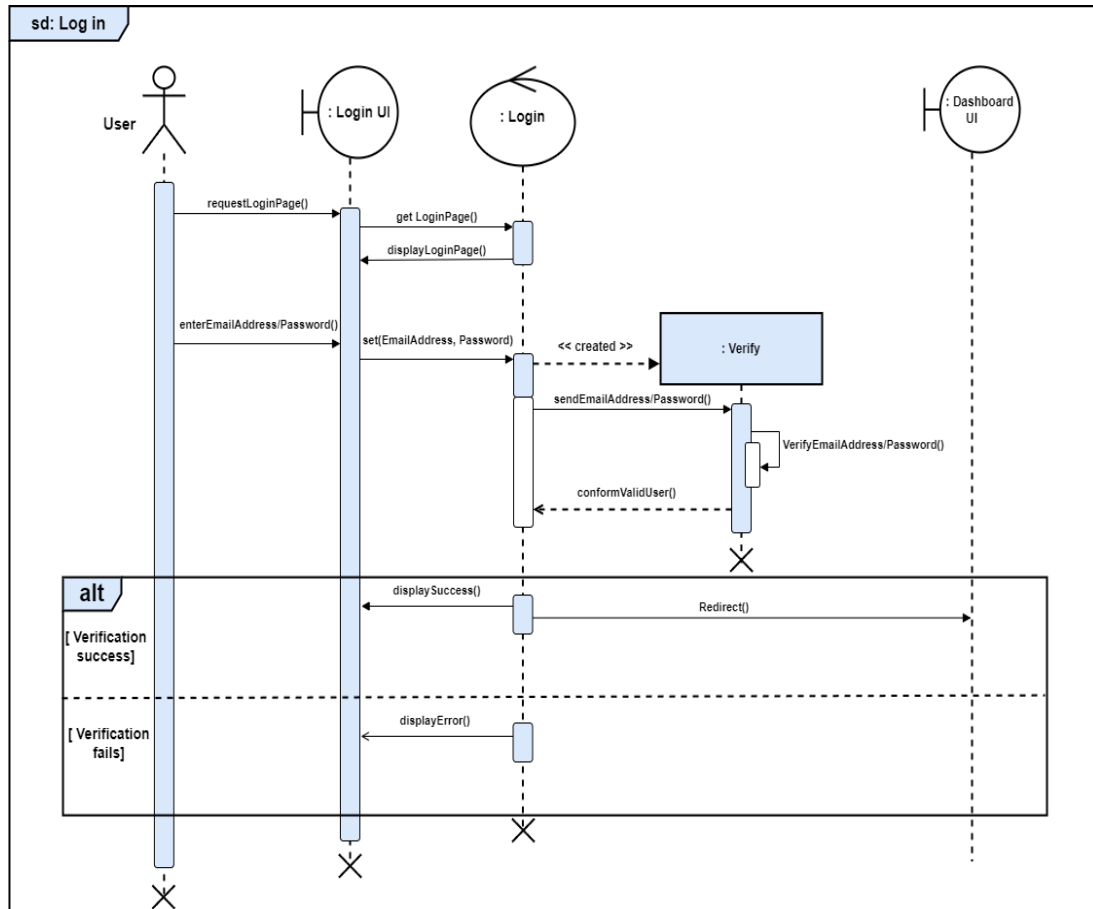


Figure 29:Login Sequence Diagram

## 2. Signup Sequence Diagram

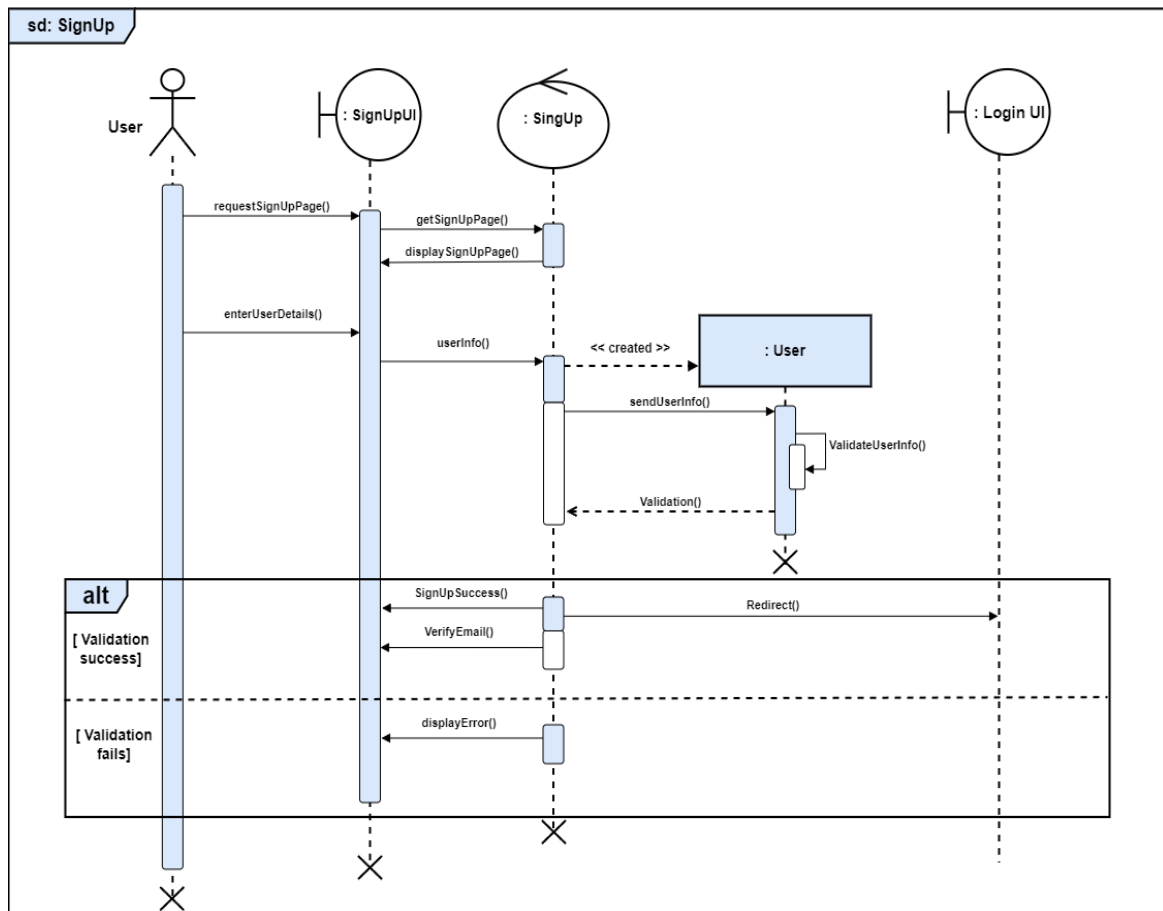


Figure 30: Signup Sequence Diagram

### 3. Food Sequence Diagram

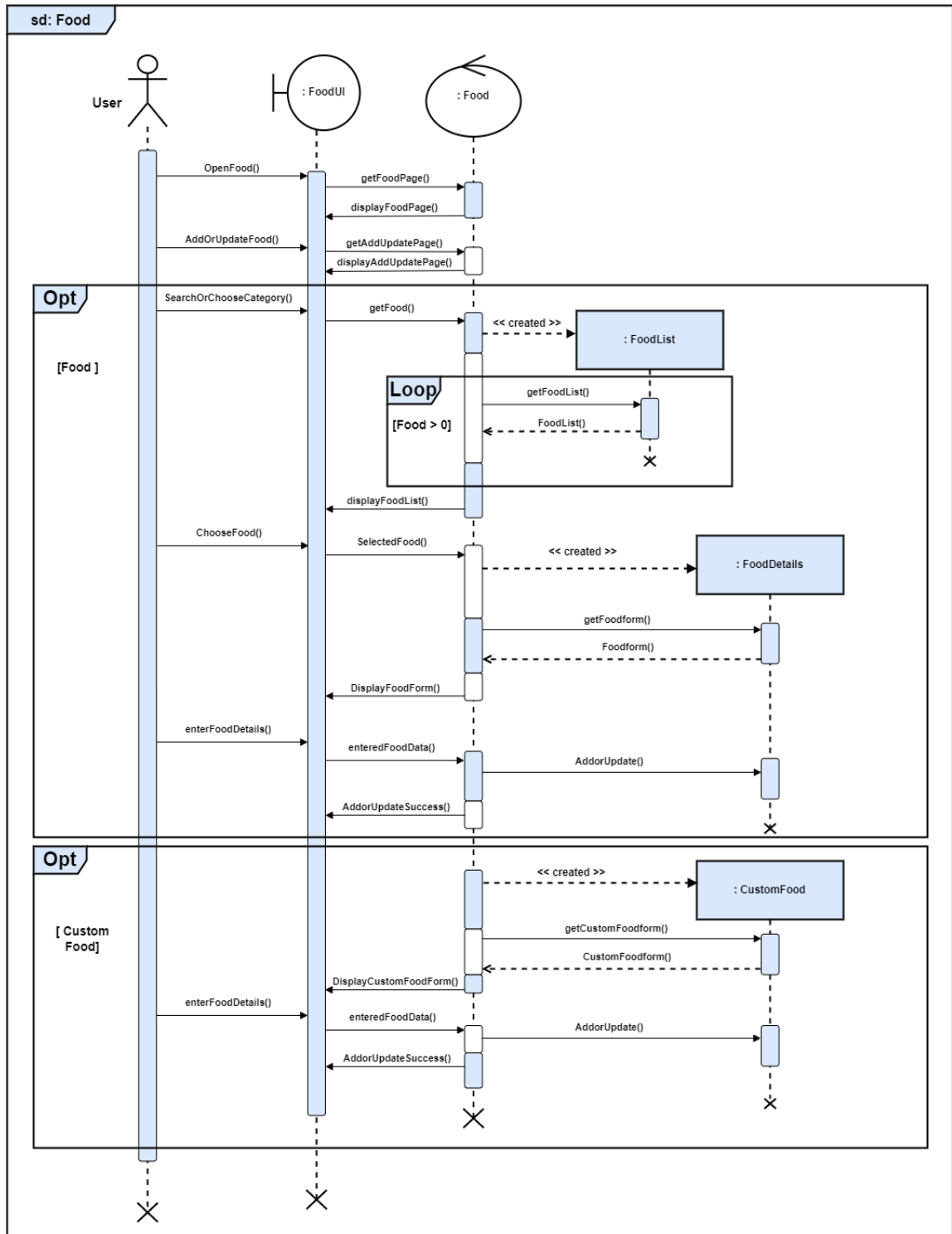


Figure 31: Food Sequence Diagram

#### 4. Exercise Sequence Diagram

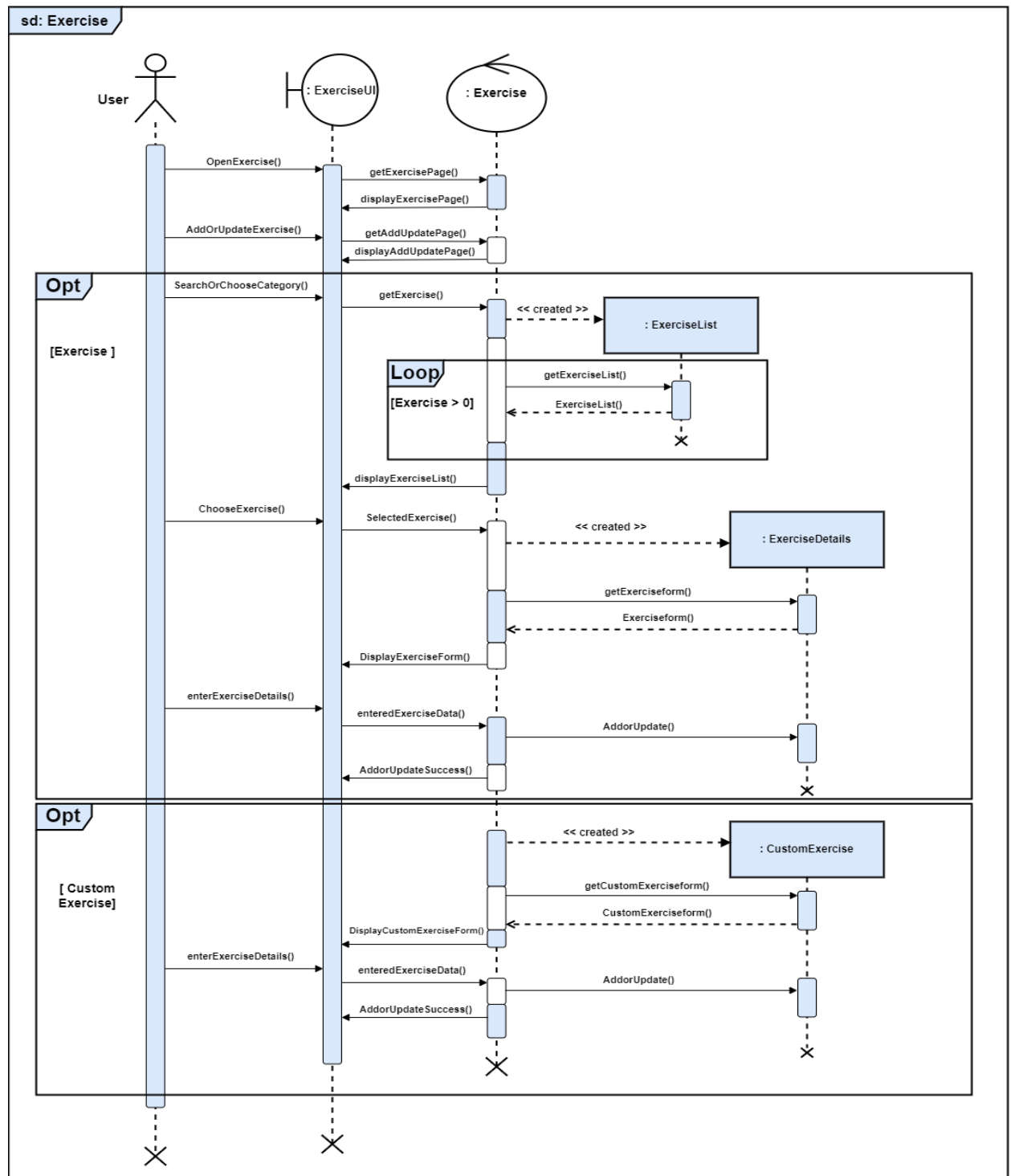


Figure 32: Exercise Sequence Diagram

## 5. Share Sequence Diagram

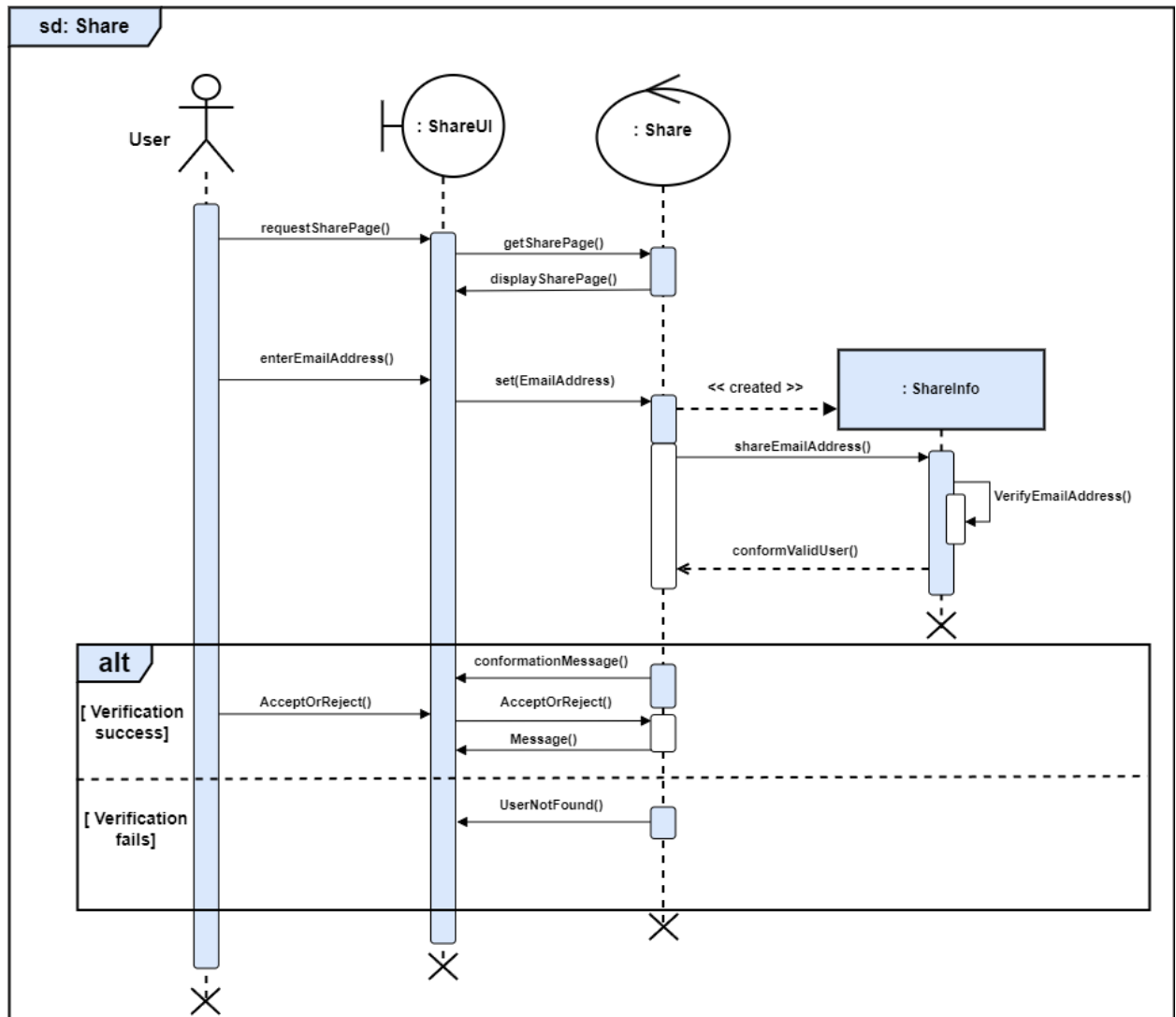


Figure 33: Share Sequence Diagram

## 6. Trends Sequence Diagram

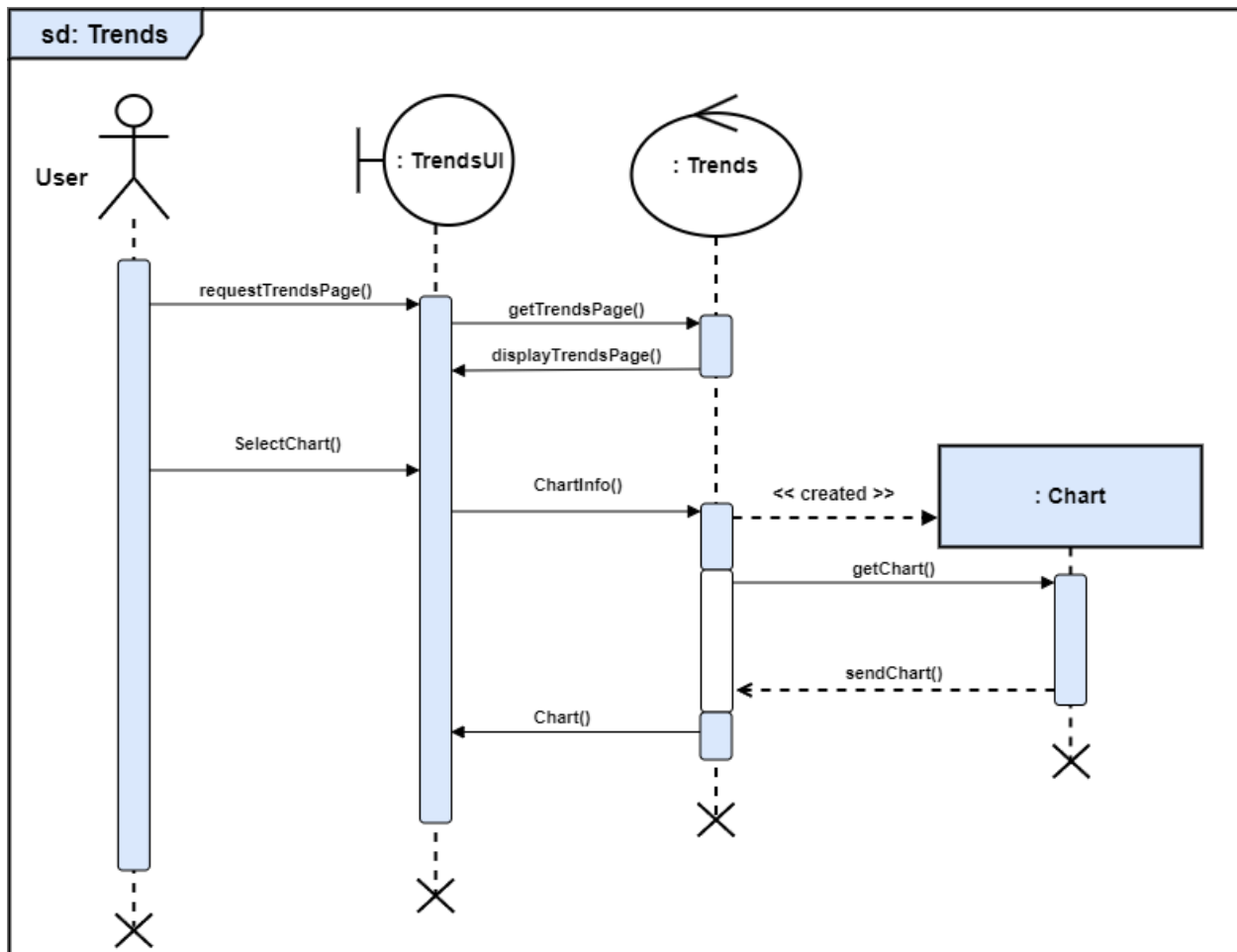


Figure 34: Trends Sequence Diagram

## 7. ClientProfile Sequence Diagram

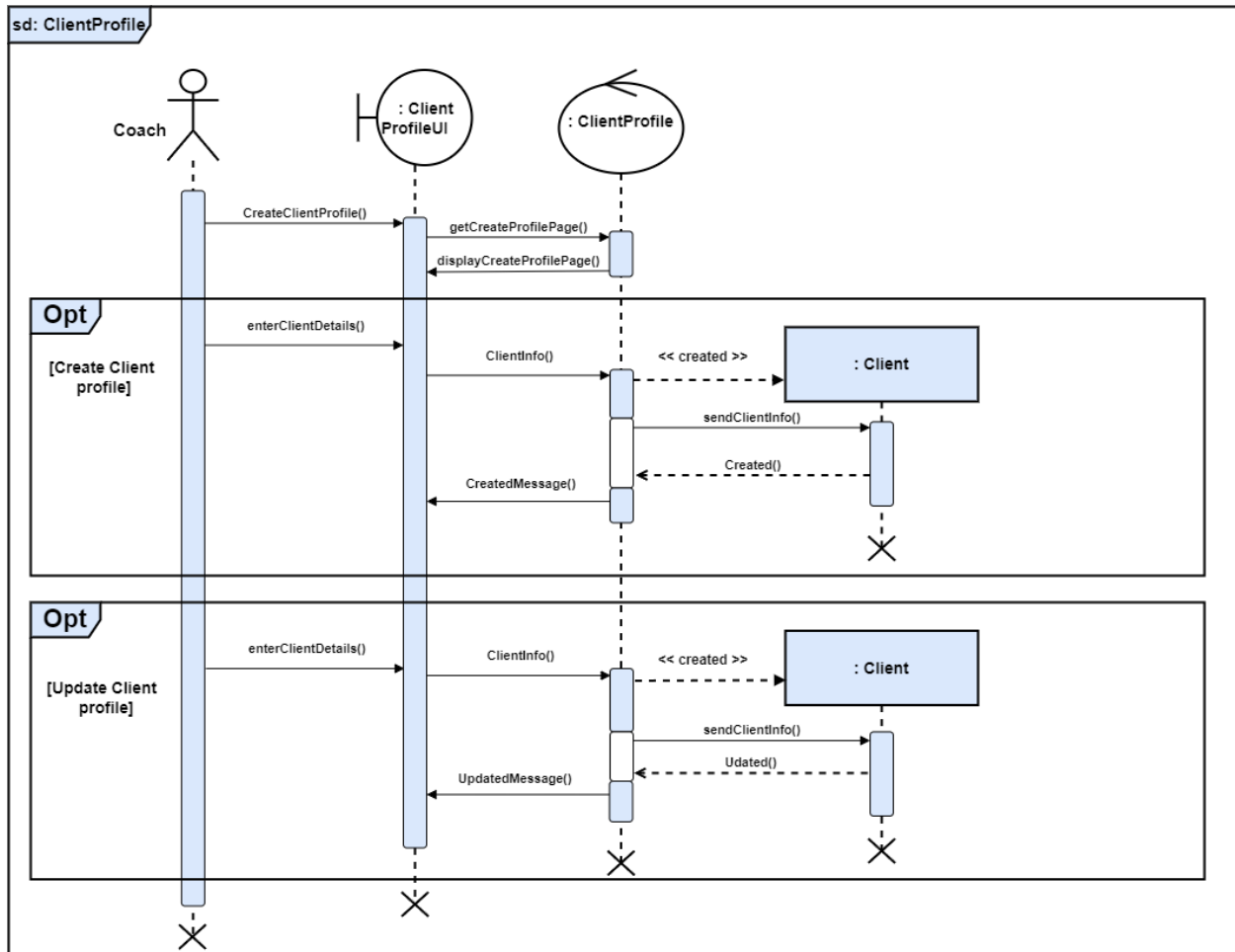


Figure 35: Client Profile Sequence Diagram

## 8. Clientinfo Sequence Diagram

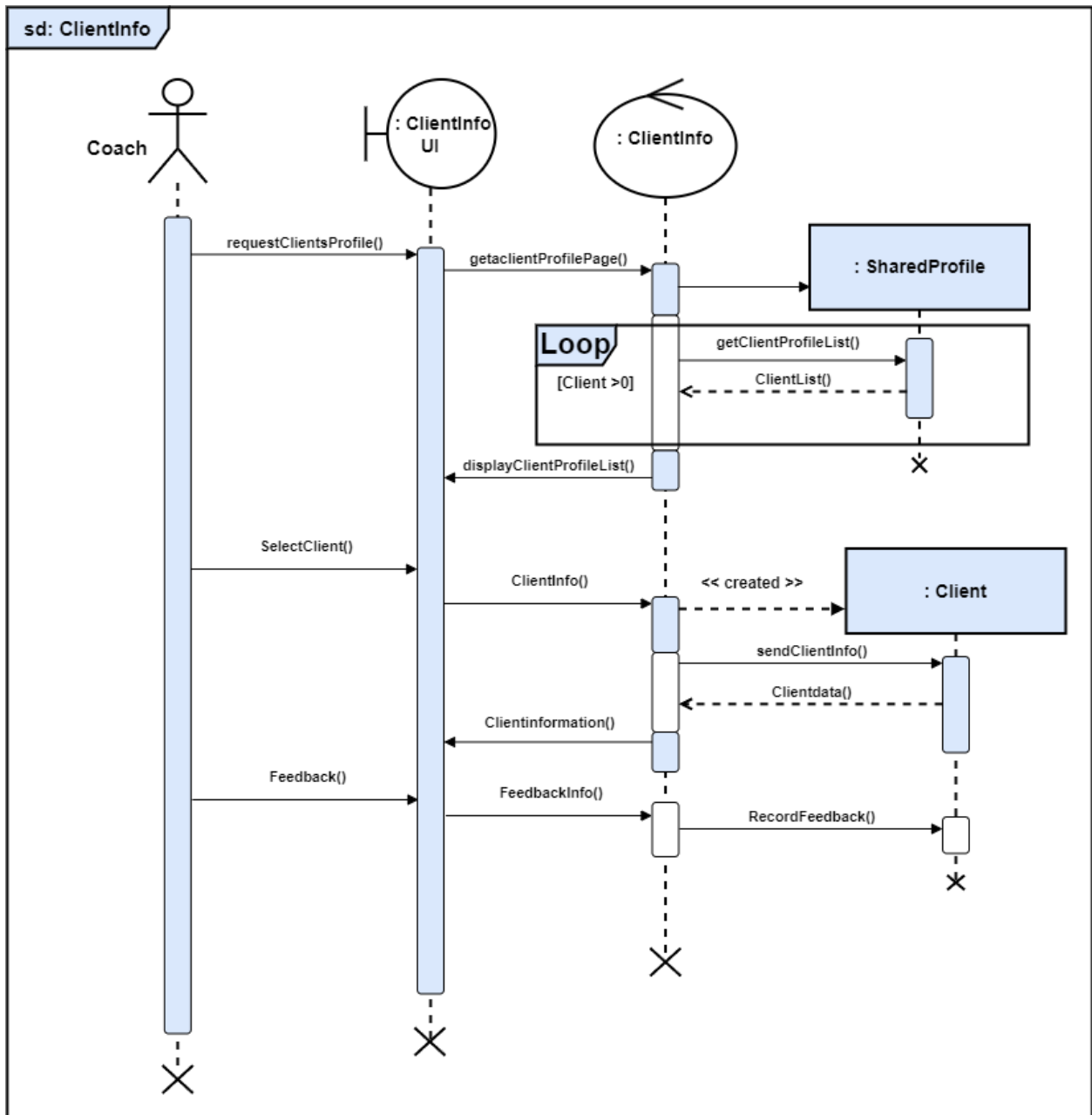


Figure 36: Client info Sequence Diagram



## 8.2.5 Collaboration Diagram

### 1. Login Collaboration Diagram

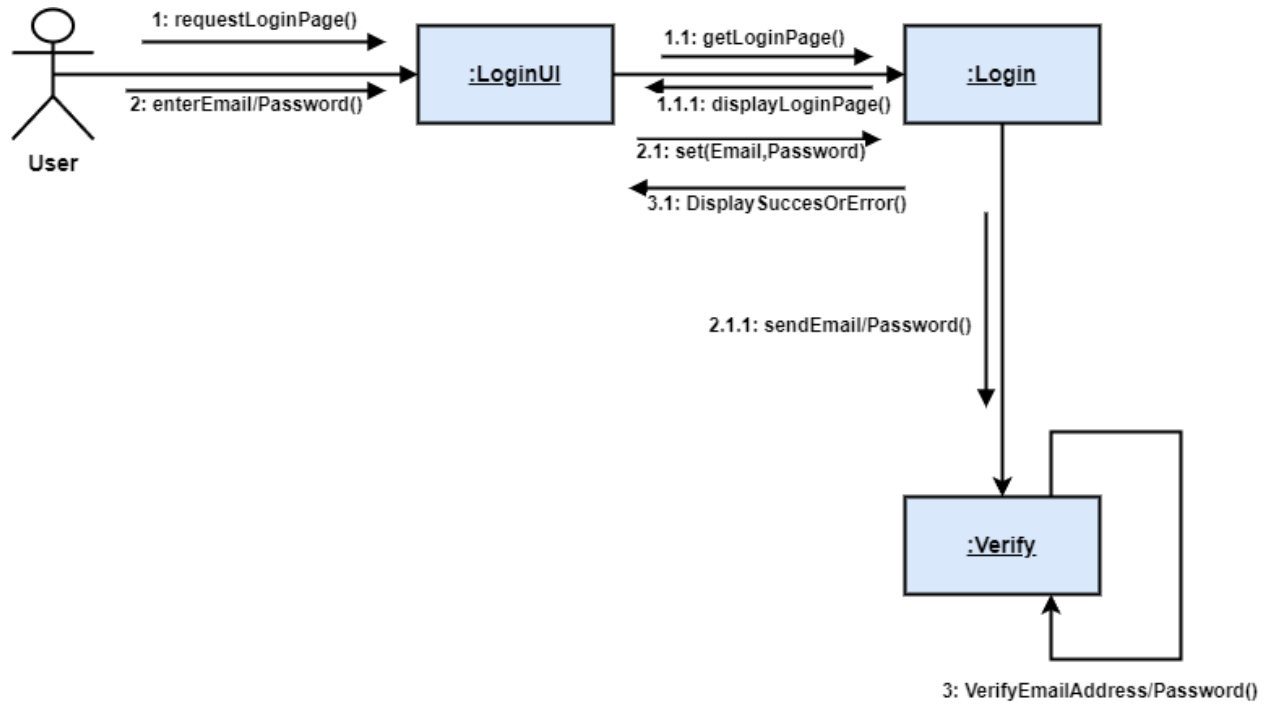


Figure 37; Login Collaboration Diagram

## 2. Signup Collaboration Diagram

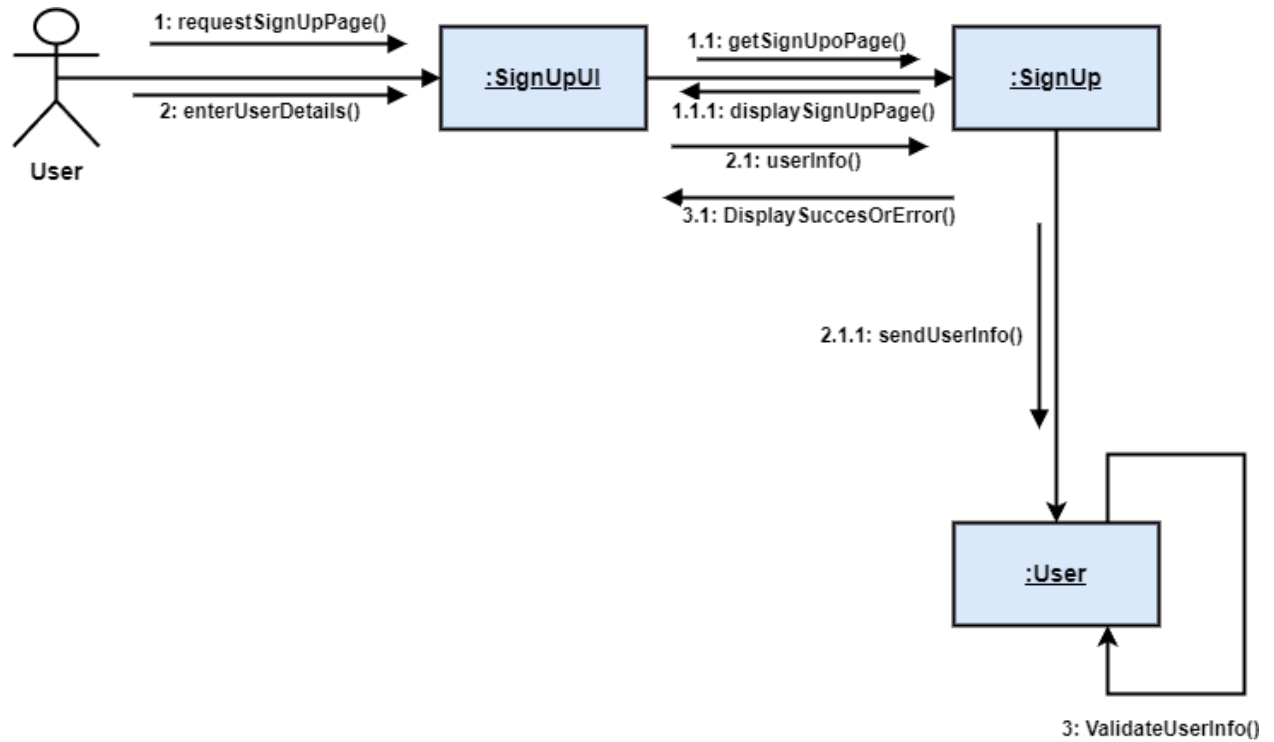


Figure 38: Signup Collaboration Diagram

### 3. Food Collaboration Diagram

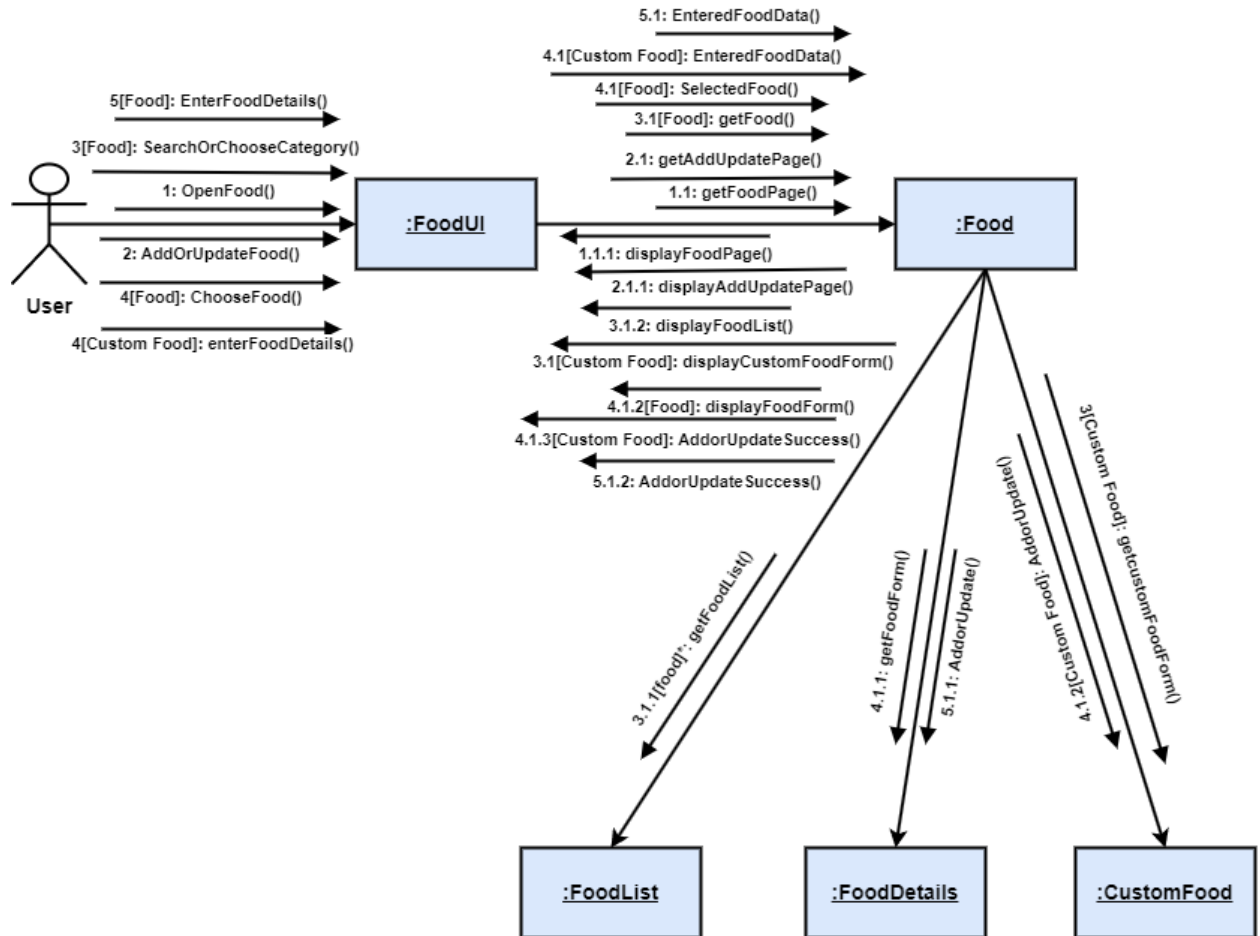


Figure 39: Food Collaboration Diagram

#### 4. Exercise Collaboration Diagram

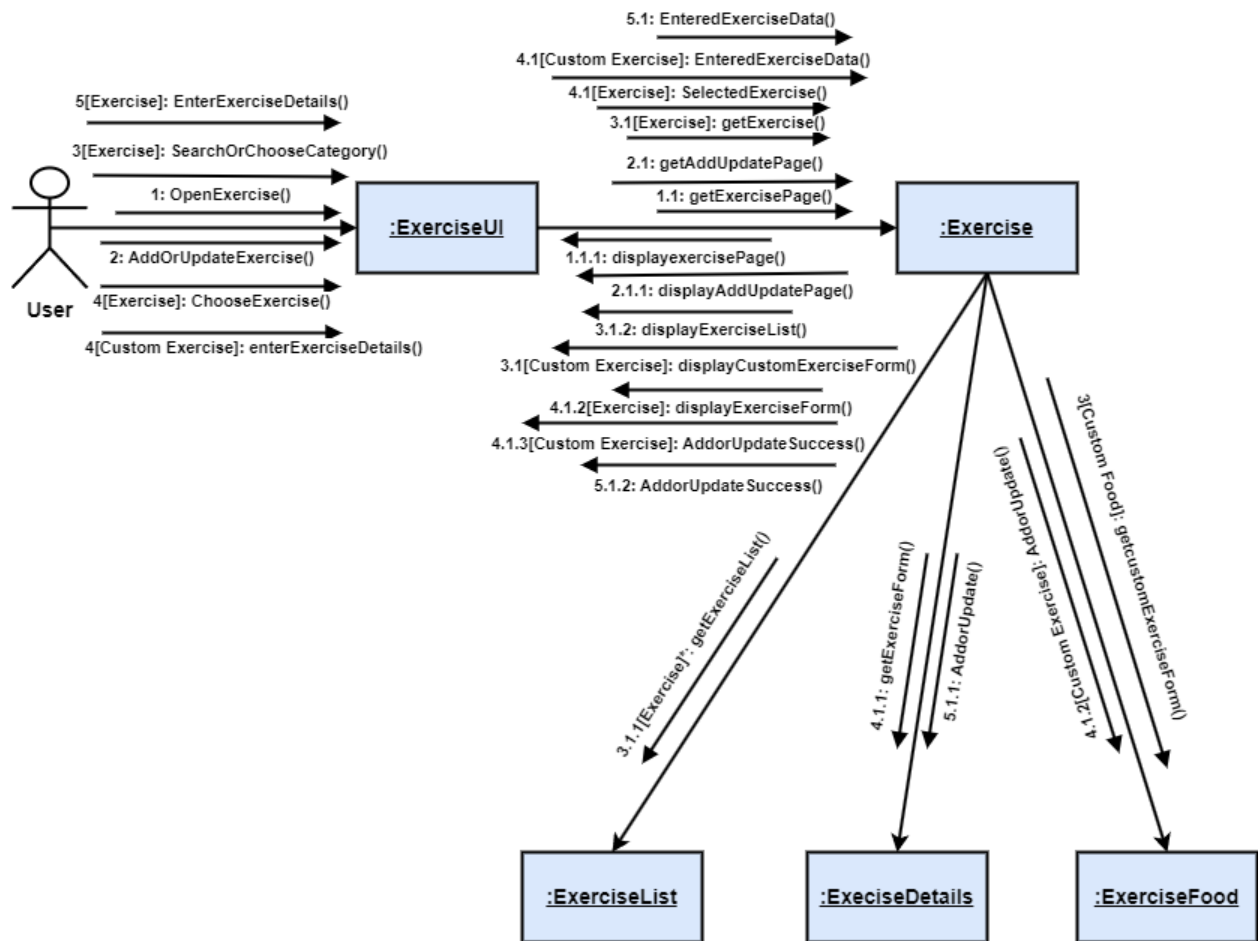


Figure 40: Exercise Collaboration Diagram

## 5. Share Collaboration Diagram

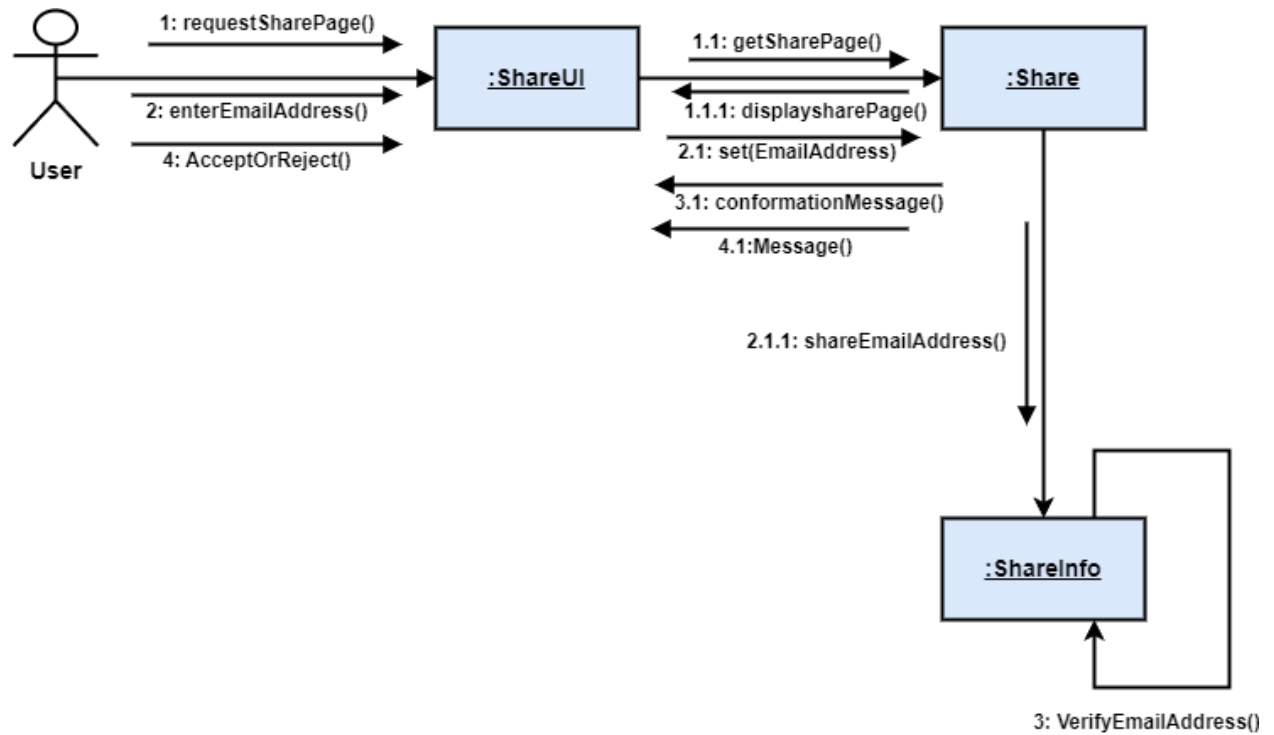


Figure 41: Share Collaboration Diagram

## 6. Trends Collaboration Diagram

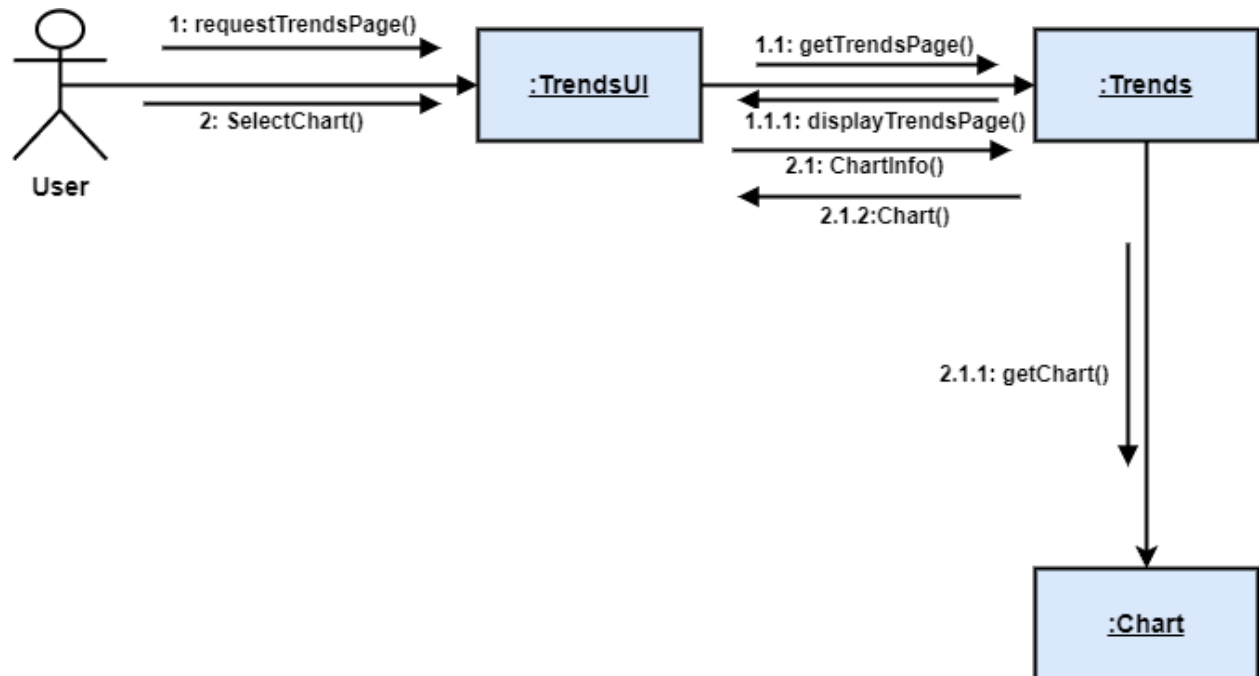


Figure 42: Trends Collaboration Diagram

## 7. Client Profile Collaboration Diagram

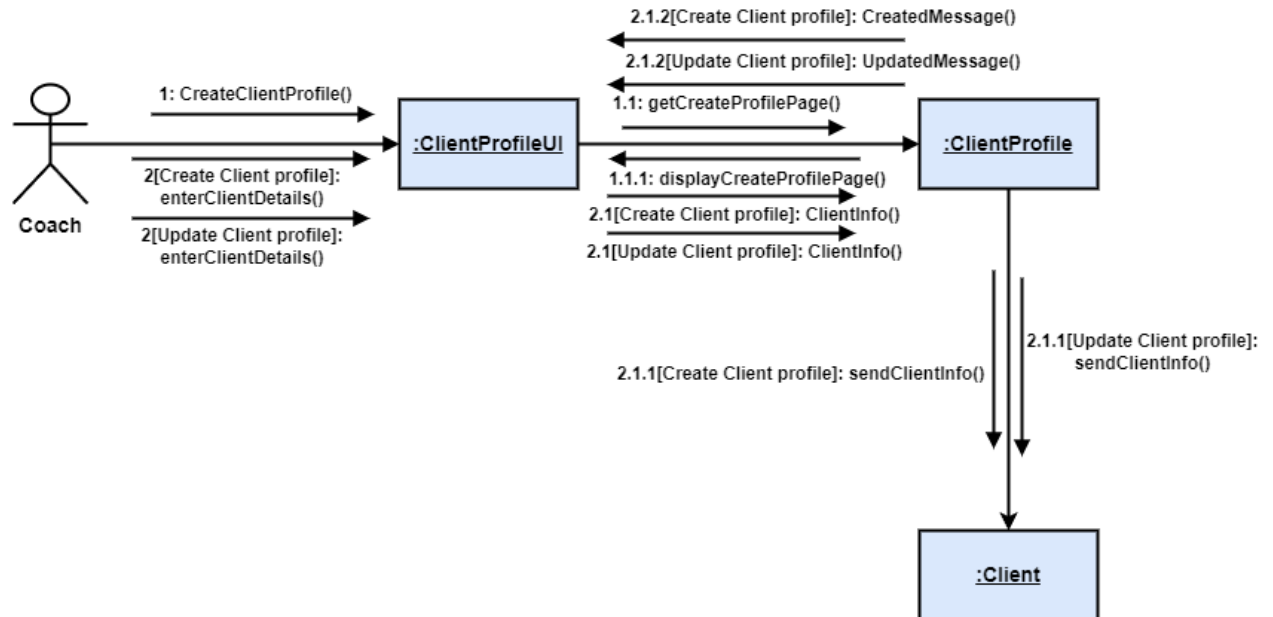


Figure 43: Client Profile Collaboration Diagram

## 8. Clientinfo Collaboration Diagram

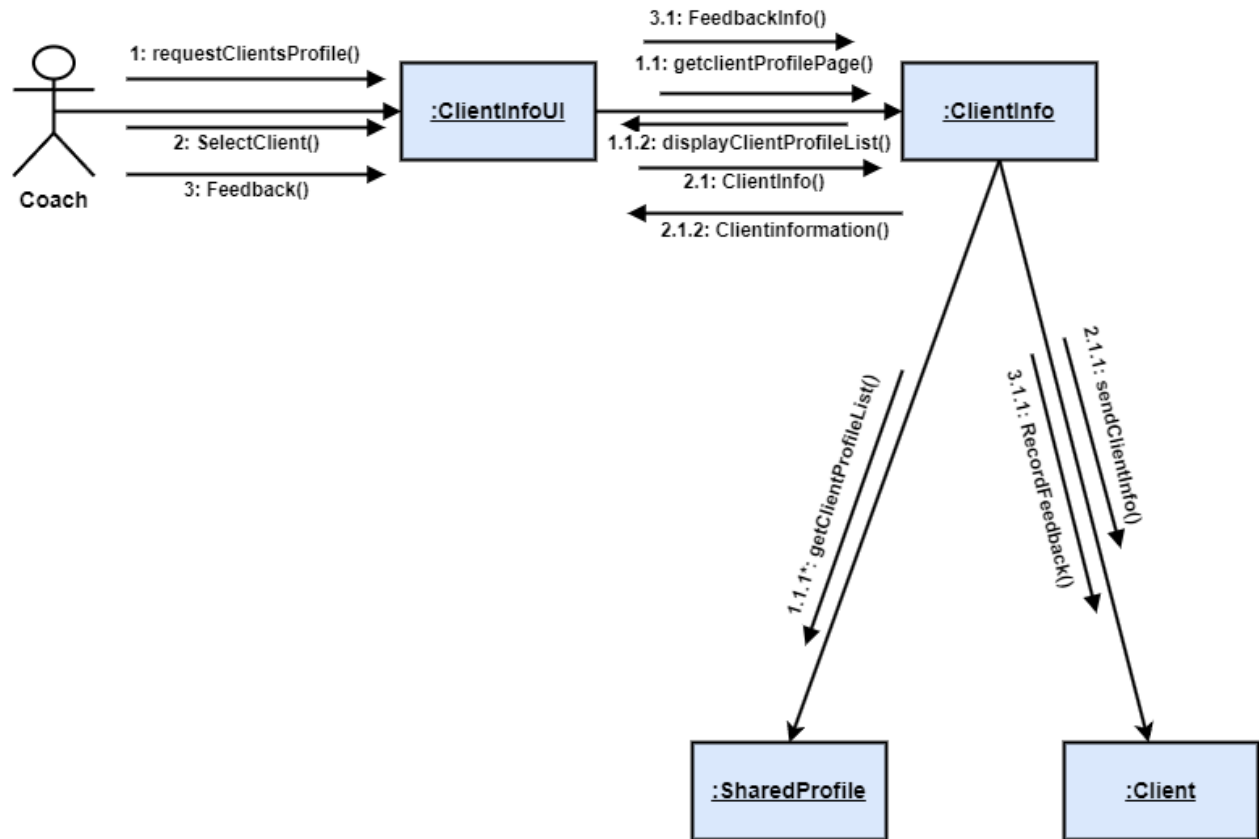


Figure 44: Client info Collaboration Diagram



## 8.2.6 Flow Chart

### 1. Login Flow Chart

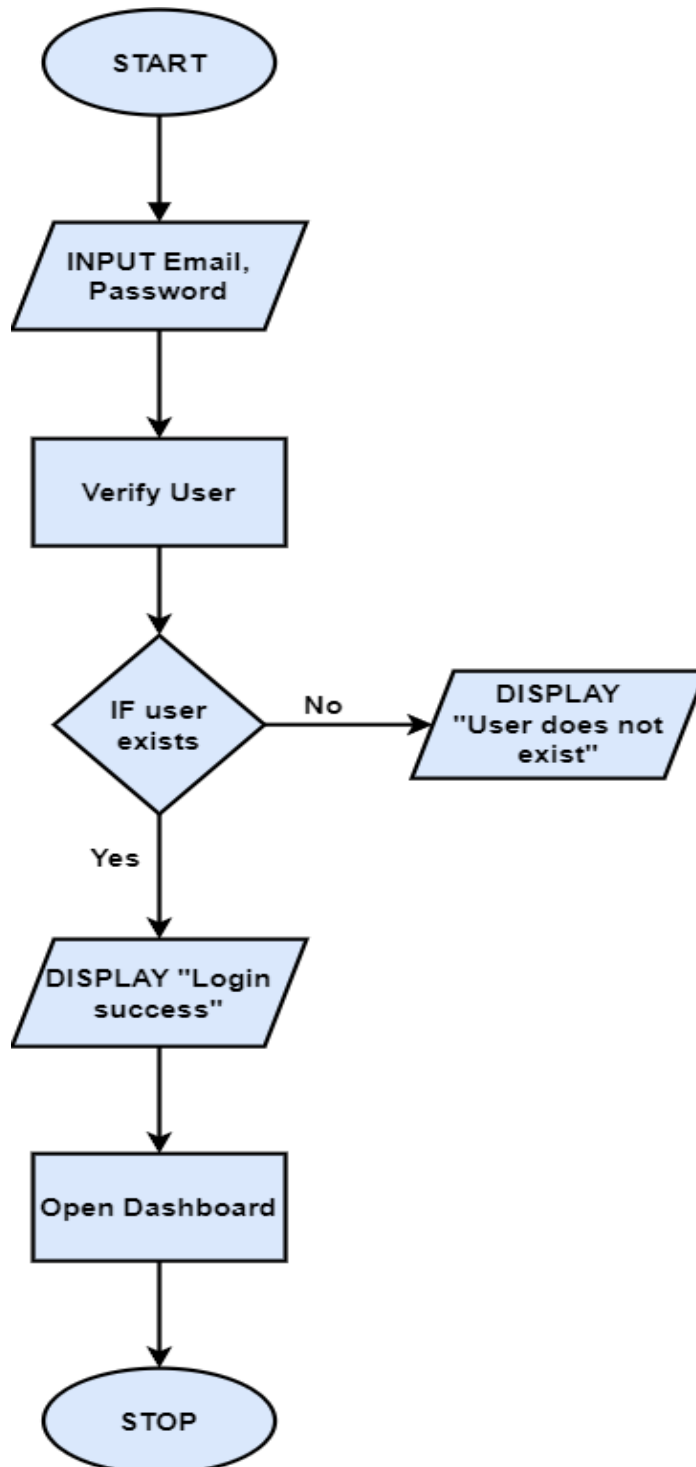


Figure 45: Login Flow Chart

## 2. Signup Flow Chart

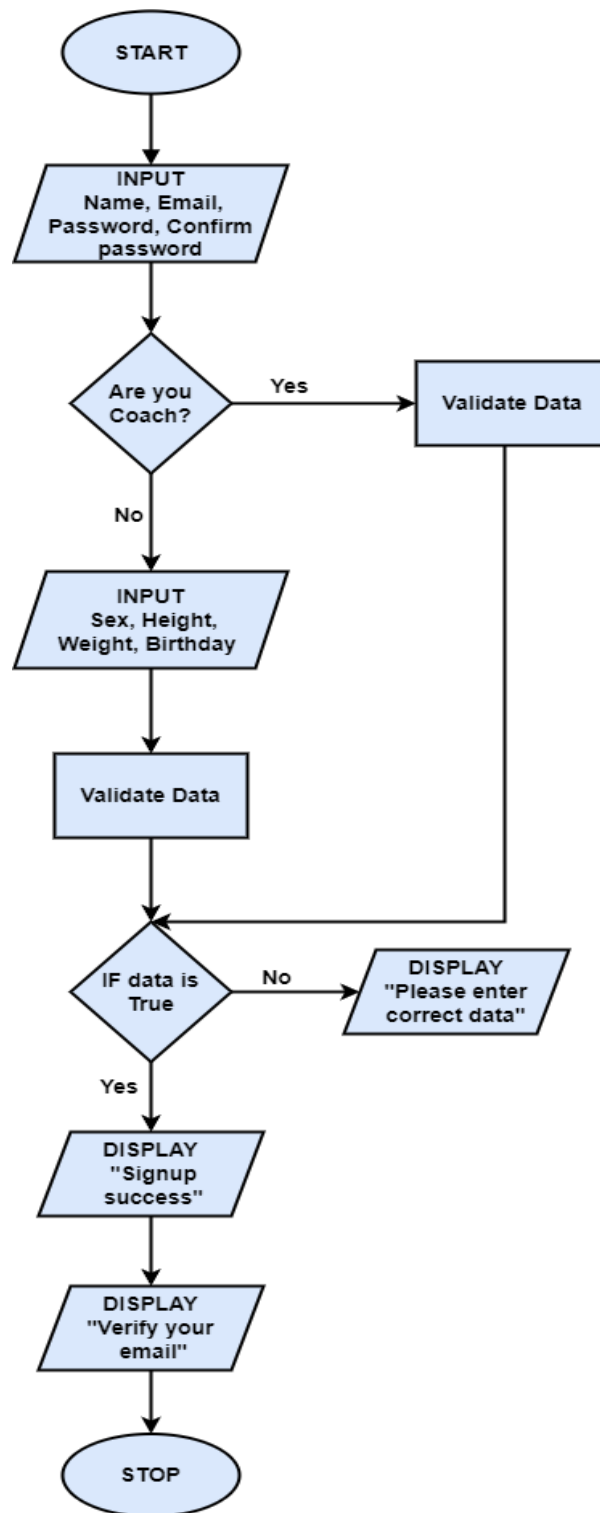


Figure 46:Signup Flow Chart

### 3. Food Flow Chart

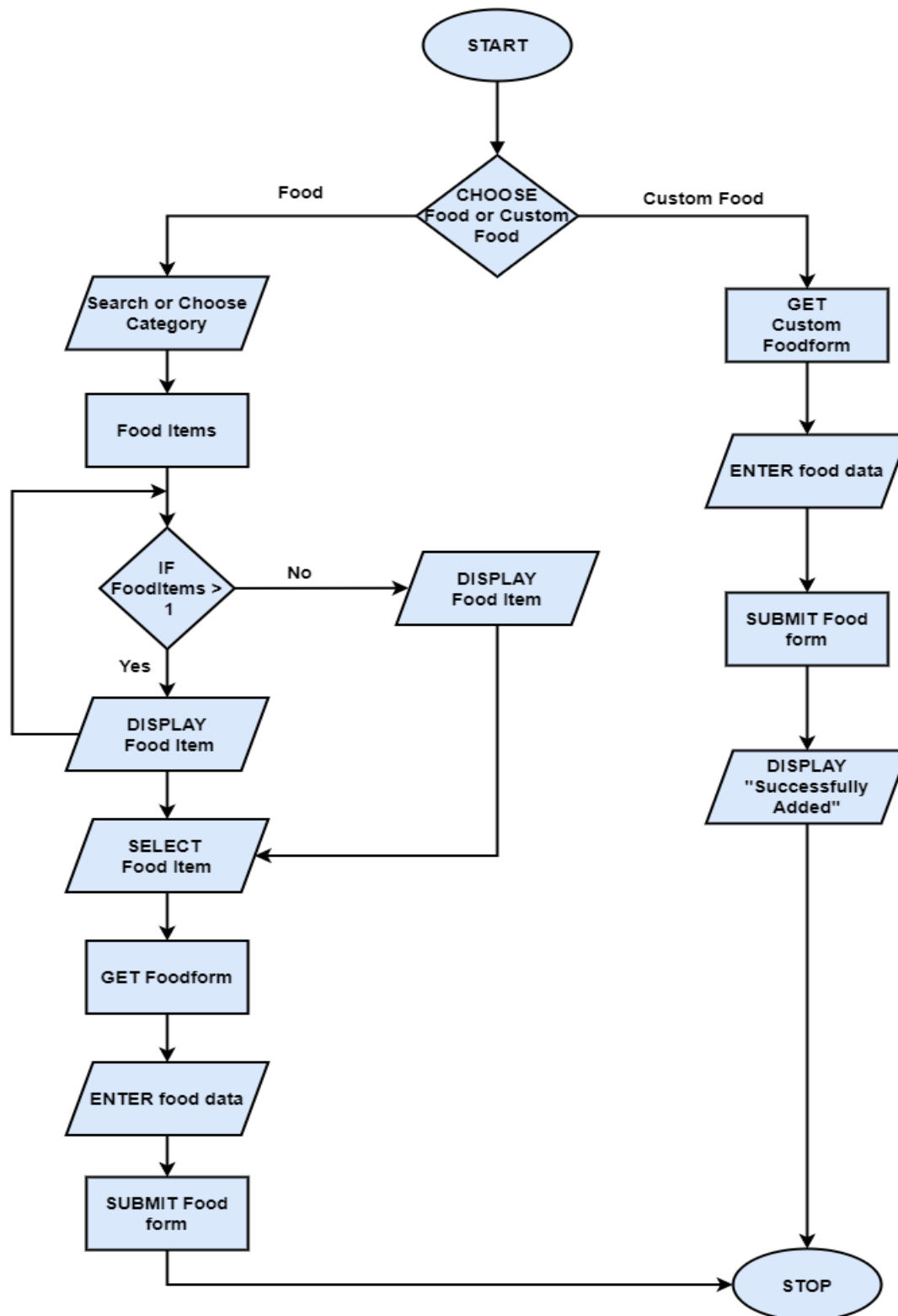


Figure 47:Food Flow Chart

#### 4. Exercise Flow Chart

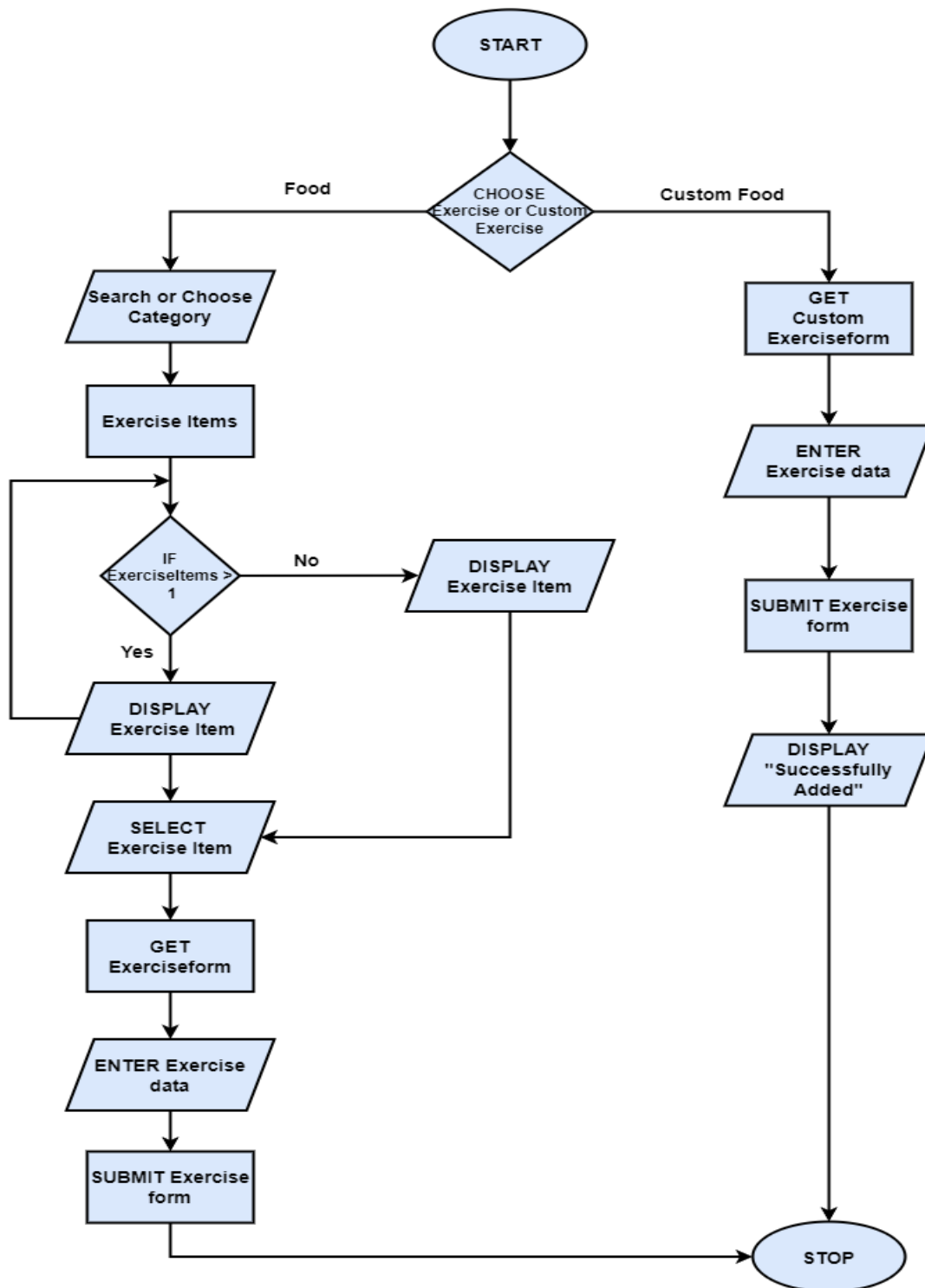


Figure 48: Exercise Flow Chart

## 5. Share Flow Chart

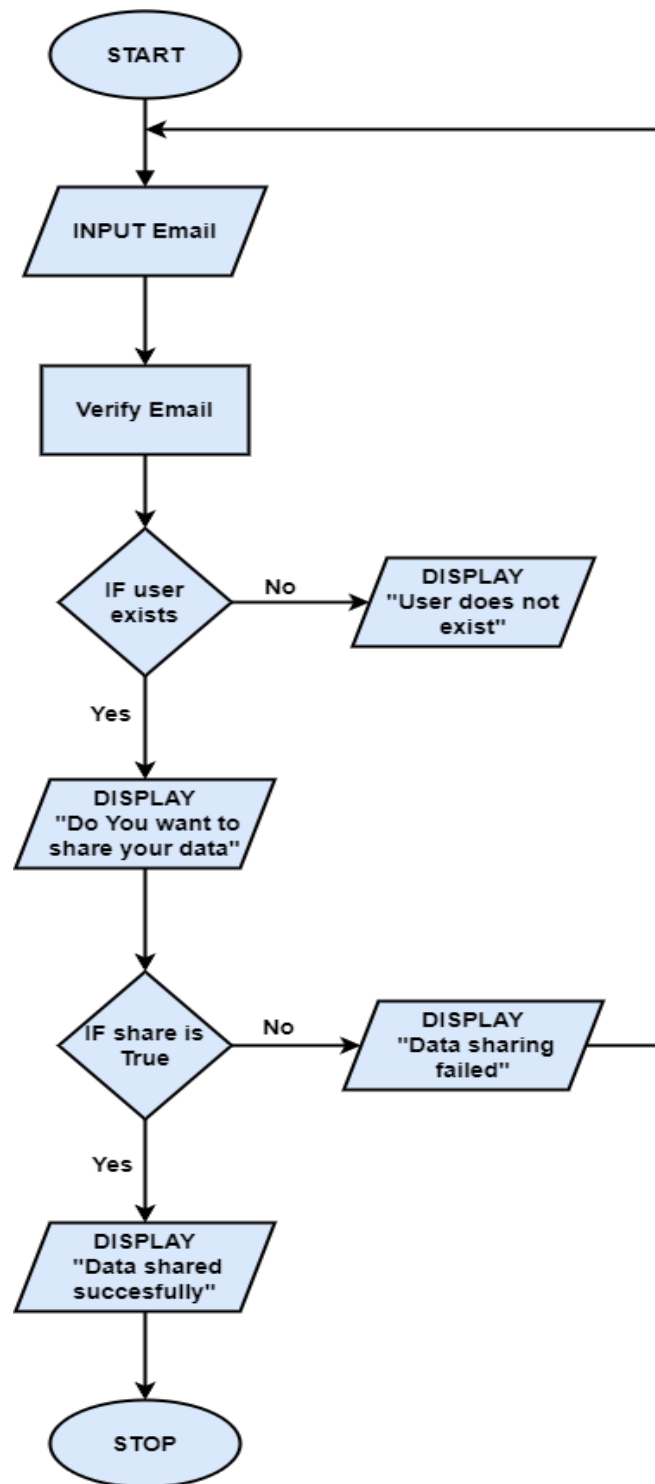


Figure 49: Share Flow Chart

## 6. Trends Flow Chart

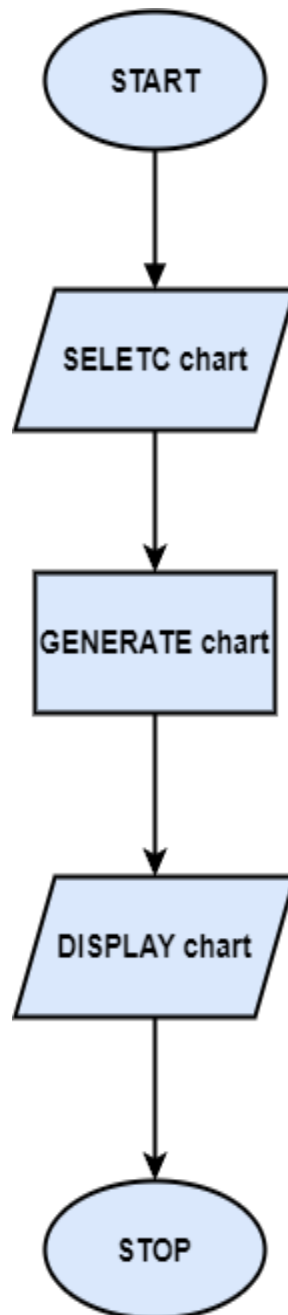


Figure 50: Trends Flow chart

## 7. Create Profile Flow Chart

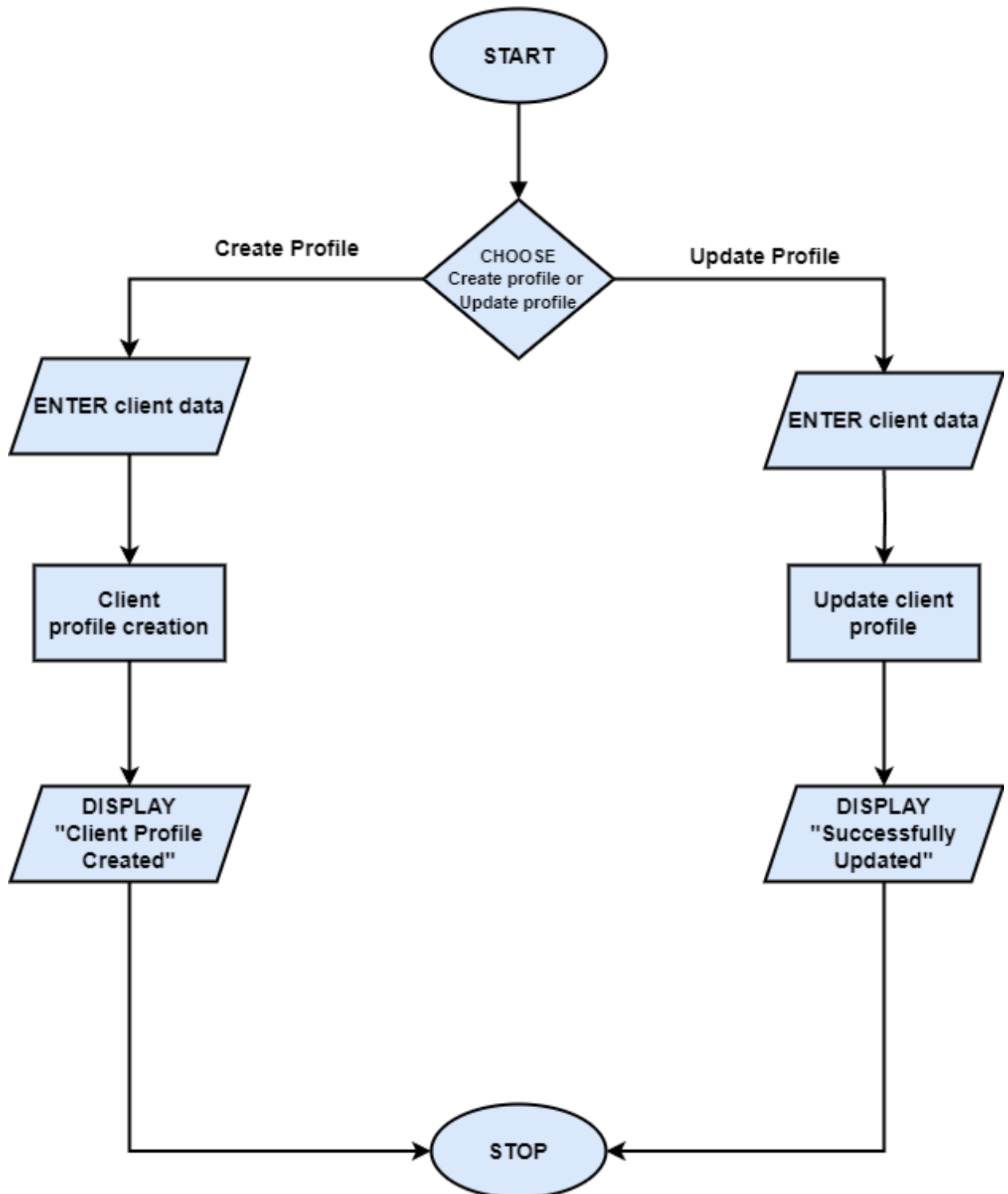


Figure 51: Create Profile Flow Chart

## 8. Client Info Flow Chart

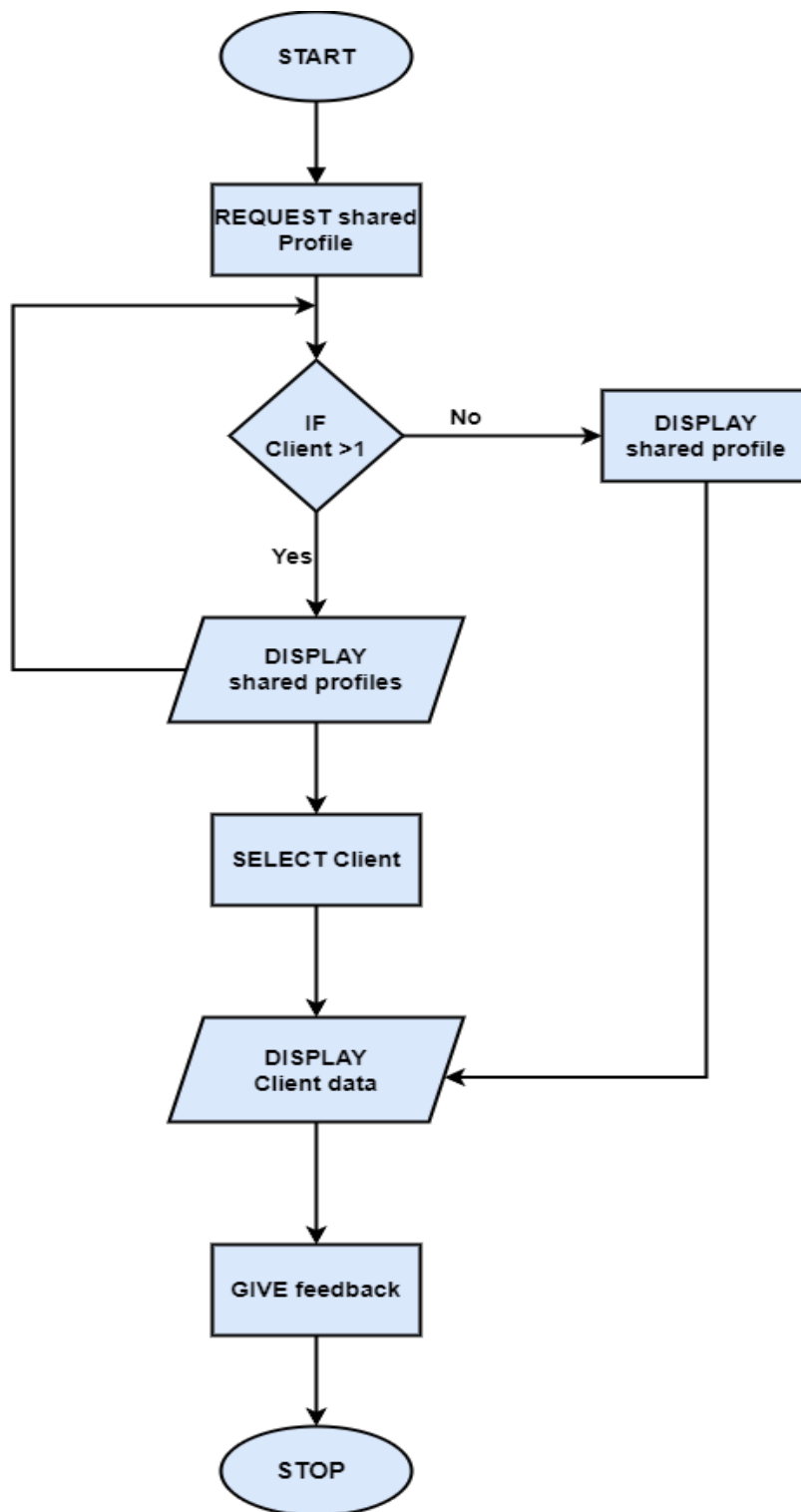


Figure 52: Client Info Flow Chart



### 8.2.7 Entity Relation Diagram

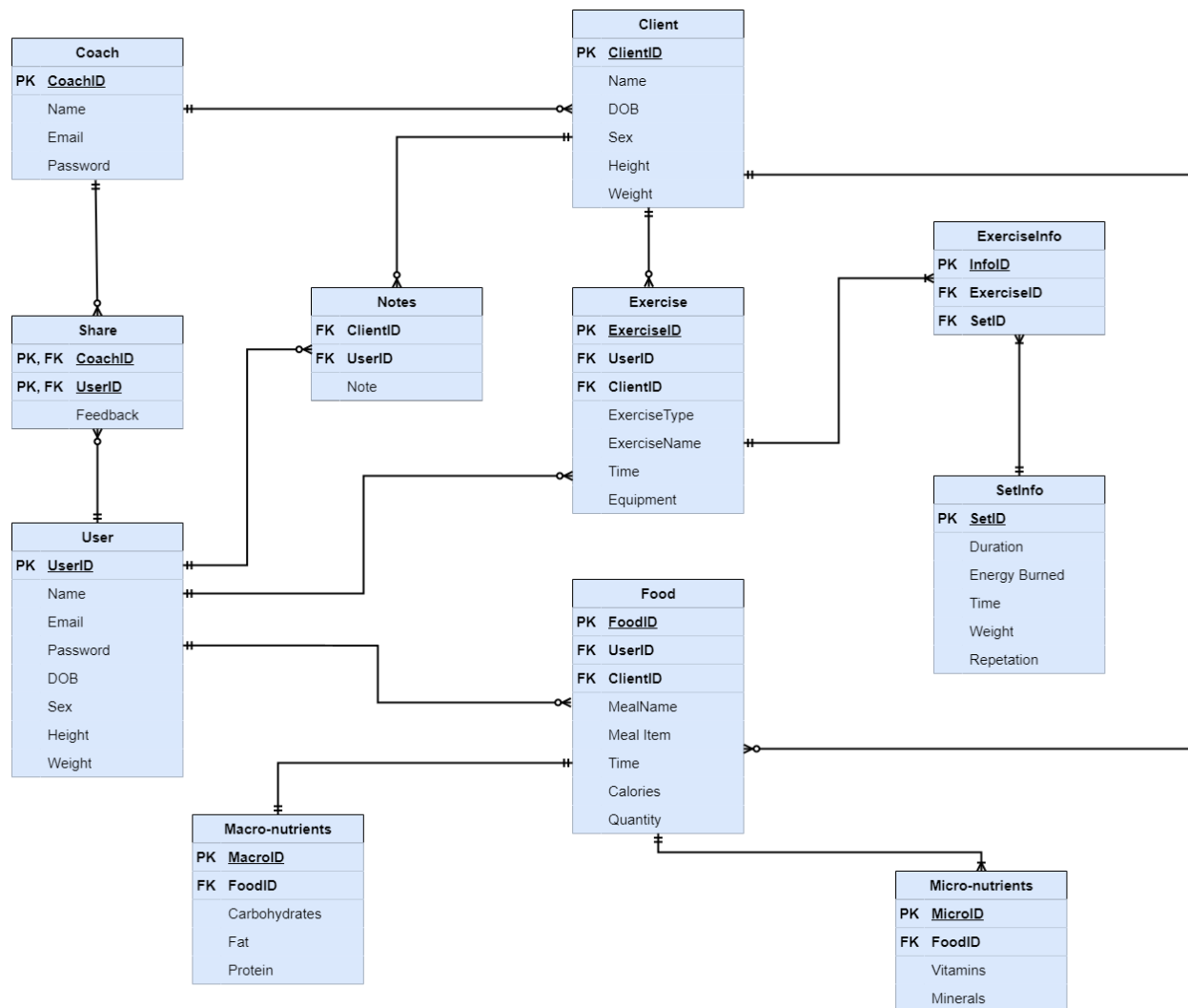


Figure 53; Entity Relation Diagram

