

VIETNAMESE-GERMAN UNIVERSITY



MULTI-DISCIPLINARY PROJECT REPORT

TRACK ME RUN

Major: Computer Science

Instructor	: TRUONG TUAN ANH, PhD
Student 1	: Đào Gia Hưng (10423048)
Student 2	: Lê Mai Bảo Linh (10423072)
Student 3	: Lê Quang Minh Khoa (10423057)
Student 4	: Nguyễn Đức Khang (10423054)
Student 5	: Võ Khánh Trần (10423177)

---000---

HO CHI MINH CITY, 07/2024

ABSTRACT

Contents

CHAPTER 1 INTRODUCTION	6
I. Context	6
II. System description	6
CHAPTER 2 SYSTEM ANALYSIS	7
I. System stakeholders	7
II. System requirements	7
1. Functional requirements	7
2. Non-functional requirements	7
III. Use case Diagram	7
IV. Use case Description	8
V. Activity Diagram	13
VI. Class Diagram	20
VII. Sequence Diagram	22
CHAPTER 3 SYSTEM DESIGN	27
I. Architectural design	27
1. Introduction	27
2. Architecture Overview	27
3. System Components	27
4. Architecture Diagram	28
5. Detailed Component Descriptions	28
6. Data Flow	28
7. Scalability Considerations	28
8. Performance Considerations	28
9. Deployment Architecture	28
10. Technology Stack	28
11. Conclusion	29
II. Database design	29
III. UI design	30
CHAPTER 4 SYSTEM IMPLEMENTATION	37
CHAPTER 5 SYSTEM TESTING	37
CHAPTER 6 SYSTEM DEPLOYMENT	38
CHAPTER 7 CONCLUSIONS	39
REFERENCES	39

LIST OF TABLES

Table 1: Use case Description - Log in	9
Table 2: Use case Description - Log out	9
Table 3: Use case Description - Create Account	10
Table 4: Use case Description - View Profile	10
Table 5: Use case Description - Start Workout Session	11
Table 6: Use case Description - Change Password	11
Table 7: Use case Description - Update Profile	12
Table 8: Use case Description - View Statistics	13
Table 9: UI Design - HTML pages	31

LIST OF FIGURES

Figure 1: Use case Diagram	8
Figure 2: Activity Diagram - Log in	13
Figure 3: Activity Diagram - Log out	14
Figure 4: Activity Diagram - Create Account	14
Figure 5: Activity Diagram - View Profile	15
Figure 6: Activity Diagram - Start Workout Session	16
Figure 7: Activity Diagram - Change Password	17
Figure 8: Activity Diagram - Update Profile	18
Figure 9: Activity Diagram - View Statistics	19
Figure 10: Class Diagram	21
Figure 11: Sequence Diagram - Sign in	23
Figure 12: Sequence Diagram - Log out	25
Figure 13: Sequence Diagram - Reset Password	26
Figure 14: Architecture Diagram for Track Me Run	28
Figure 15: Database Structure Diagram	29
Figure 16: Login Interface	31
Figure 17: Sign up	32
Figure 18: Fill Profile	32
Figure 19: User Profile	33
Figure 20: Update Profile	33
Figure 21: Home Page	34
Figure 22: Change Password	34
Figure 23: Set Goal	35
Figure 24: Start Session	35
Figure 26: Completed Goal	36

CHAPTER 1 INTRODUCTION

I. Context

Due to higher demands in personal fitness training nowadays, many fitness enthusiasts and service providers are constantly looking for ways to enhance physical well-being and make exercising products more accessible. In recent years, a particular type of workout has emerged as one of the most preferable means: running. In fact, according to World Economic Forum, running is the second most popular form of weekly exercise, chosen by 19 percent of 21,000 surveyed adults across 29 markets. Evidently, more adults are paying attention to running and jogging as an effective health maintenance method.

In order to improve running workout quality and motivate runners to keep on their practice, there is a need for runners to interact with their workout statistics in a meaningful way. One practical and easy to implement tracking data is calorie consumption. The number of calories burnt is the primary point of focus for many running participants. By tracking it, runners are able to proactively build a reasonable diet to ensure a balance between calorie intake and consumption, thereby maintaining their physical fitness.

There are numerous approaches when it comes to collecting calorie consumption data among runners. For seasoned and dedicated runners, they usually opt for a wearable fitness tracker, such as an Apple Watch or Google Fitbit. These gadgets offer a significantly high level of precision and are well-trusted by many. However, since there is a considerable barrier of entry to these products for newbies, they typically utilize the tracking sensors available on their smartphones. Therefore, developing accessible and cost-saving mobile software to meet this need is a solution in great demand. For this project, the team has decided to utilize the sensor capabilities of a smartphone and IoT platform support of Arduino Cloud to develop a mockup program to track users' positions during a running session and calculate calorie consumption from that data.

II. System description

The aim of our website is to manage and keep track of the information of the users' running sessions. Such information as calories burnt, total distance, total time, the maximum and average speed is tracked, stored in our database, and accessible by users. This system can help users with running training and their physical health building journey. After each of the users' sessions, they can check the statistics of their performance, which allows for necessary changes in running methods or in future plans.

CHAPTER 2 SYSTEM ANALYSIS

I. System stakeholders

- Standard users: register an account and get access to the data of such account (the data collected from the GPS device that the GPS holder holds).
- GPS Holders: whose training sessions are tracked and the statistics of this person are stored in the system (this person can also be the one whose role is the standard user).
- Admin: responsible for managing the system and database and maintaining and upgrading the website.

II. System requirements

1. Functional requirements

The system provides the following functional requirements:

- As an admin, he/she can access to the database and user profiles
- As a registered user, he/she can create personal accounts to have their own data saved and view the calories burnt, max speed, average speed, total distance, and total time he/she has run in a particular session between the start and the end of the programme.

2. Non-functional requirements

The system provides the following non-functional requirements:

- The user data is secured by being encrypted by the system with every movement executed.
- English is the main language used in the web application.
- User-friendly, users should be able to view the information within 2 clicks.
- The response time is less than 2 seconds.
- The system ensures 24/7 operation. Downtime shall not exceed 2 hours per year.

III. Use case Diagram

The use case diagram of the system is depicted in Figure 1.

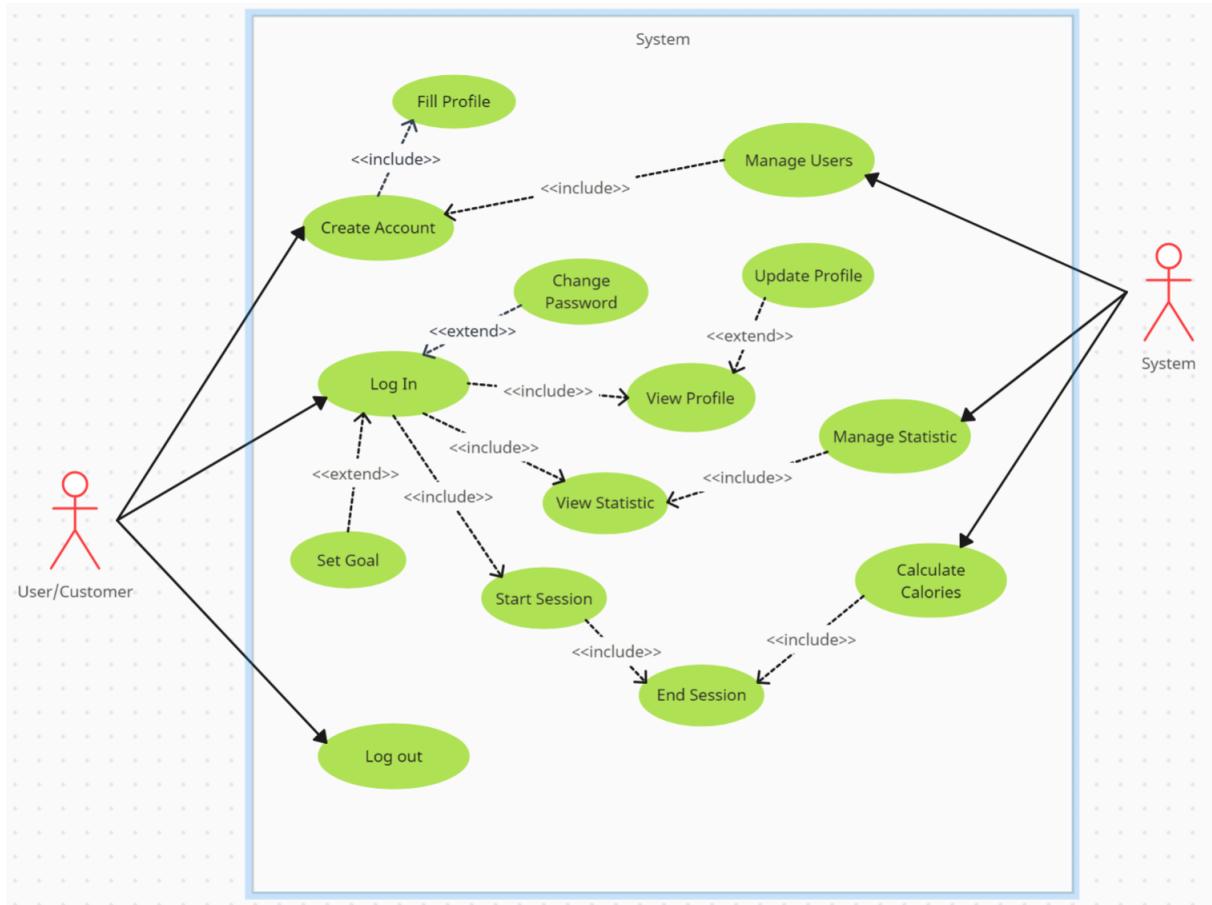


Figure 1: Use case Diagram

IV. Use case Description

❖ Use case 1. Log in

Use case: Log in	
Purpose	To access user information and begin a new session on the web app
Description	User provides a username and password to log in
Actor(s)	User
Pre-Condition(s)	User is on the log-in page, providing a username and password
Post-Condition(s)	Username and password combination matches one in a predetermined database, the web app logs user in
Basic flows	User
	1. Enter username, password, click Log In button
	System
	2. Retrieve Log In data and validate it using an external database

		3. If matching, allow user to log into their account
Alternative Flows	3.1. Web app displays a message about incorrect login credentials, requesting user to reinput login information.	

Table 1: Use case Description - Log in

❖ Use case 2. Log out

Use case: Log out		
Purpose	To end the login session on the web app	
Description	User log out	
Actor(s)	User and web app	
Pre-Condition(s)	User is in the login session	
Post-Condition(s)	User sends an exit request by pressing a log out button on the interface.	
	User	System
Basic flows	1. Press the log out button on the interface.	2. Send the exist request to the Server. 3. Log out user account.
Alternative Flows	3.1. Not applicable	

Table 2: Use case Description - Log out

❖ Use case 3. Create Account

Use case: Create Account		
Purpose	To create a new account for a new user	
Description	User goes to the Sign Up page, fills out username and password information and other telemetry data for account creation	
Actors	User and system	
Pre-Conditions	Account has not been created before	
Post-Conditions	New user account is created and a new account entry is added to database	
	User	System

Basic flows

	1. Provide a new username and password 2. Click the Sign Up button	3. Compare username with existing usernames in the database 4. If not matching, accept it as a new account's credential and add it to the database
Alternative Flows	System does not add a new username to the database Displays a message "Username already exists"	

Table 3: Use case Description - Create Account

❖ Use case 4. View Profile

Use case: View Profile		
Purpose	To see the profile of the user and change profile's information	
Description	Show and update the information of the user	
Actor(s)	User	
Pre-Conditions	User is in the login session.	
Post-Conditions	User press the Home button. Or if the User change his/her profile, he/she enters new data and update the profile.	
Basic flows	User	System
	1. Check profile 2. Update profile's information (optional)	3. Send the input back to the system 4. Update the database based on the user's input 5 System always show the up-to-date profile to the user.
Alternative Flows	Not applicable	

Table 4: Use case Description - View Profile

❖ Use case 5. Start Workout Session

Use case: Start Workout Session	
Purpose	To begin a new workout session
Description	User start a workout session
Actor(s)	User

Pre-Condition	User is logged in, wants to start a workout session	
Post-Condition	None	
	User	System
Basic flows	1. Start a new session 2. Set workout goal 3. Start workout 4. Finish workout	5. Save user session 6. Calculate user statistic
Alternative Flows	1. Open Arduino App 2. Start workout 3. Finish workout	

Table 5: Use case Description - Start Workout Session

❖ Use case 6. Change Password

Use case: Change Password		
Purpose	Change the password of the account	
Description	Change the password of the account	
Actor(s)	User	
Pre-Condition	User is already in the login session.	
Post-Condition	User must enter the old password correctly, and then enter the new password and confirm.	
	User	System
Basic flows	1. Enter old password 2. Enter new password and confirm 3. Reenter if necessary (wrong old password or new password and the confirm password is not the same)	4. Check the old password and send the new password back to the main file. 5. Update the database based on the user's input.
Alternative Flows	The page shows warning to the user when he/she inputs the wrong password.	

Table 6: Use case Description - Change Password

❖ Use case 7. Update Profile

Use case: Update Profile

Purpose	Check for personal basic information of that account and update if needed	
Description	Show personal information and update when user changes it	
Actor(s)	User	
Pre-Condition	Login	
Post-Condition	None	
Basic flows	User	System
	1. View personal information 2. Update if needed	3. Update information is sent to database 4. System updates user information and show the new data
Alternative flows	Not applicable	

Table 7: Use case Description - Update Profile

❖ Use case 8. View Statistics

Use case: View Statistics		
Purpose	Check for past session of exercising	
Description	User can see past session information: calories burnt, distance run, highest speed, average speed, time spent,...	
Actor(s)	User	
Pre-Condition	User is in the login session	
Post-Condition		
	User	System

Basic flows

	1. Check for past session's data	2. Retrieve data from database and show user past session information
Alternative Flows	Not applicable	

Table 8: Use case Description - View Statistics

V. Activity Diagram

❖ Activity 1. Log in

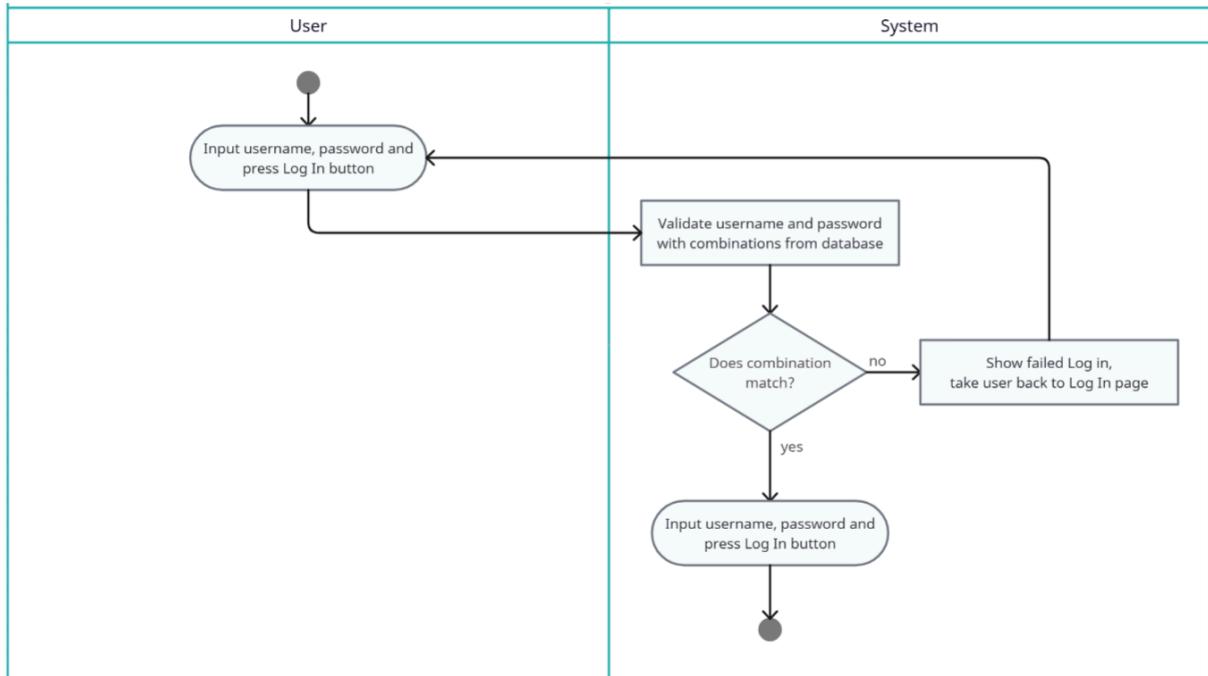


Figure 2: Activity Diagram - Log in

Description: Short description for this activity diagram...

❖ Activity 2. Log out

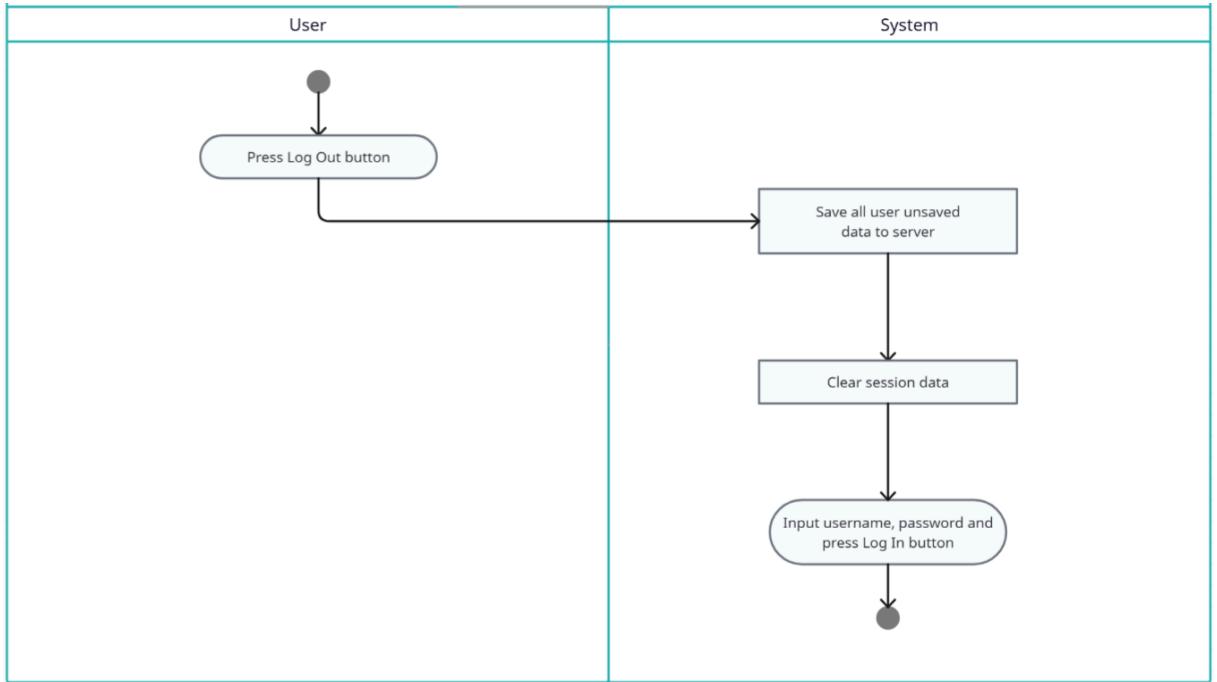


Figure 3: Activity Diagram - Log out

Description: Short description for this activity diagram...

❖ Activity 3. Create Account

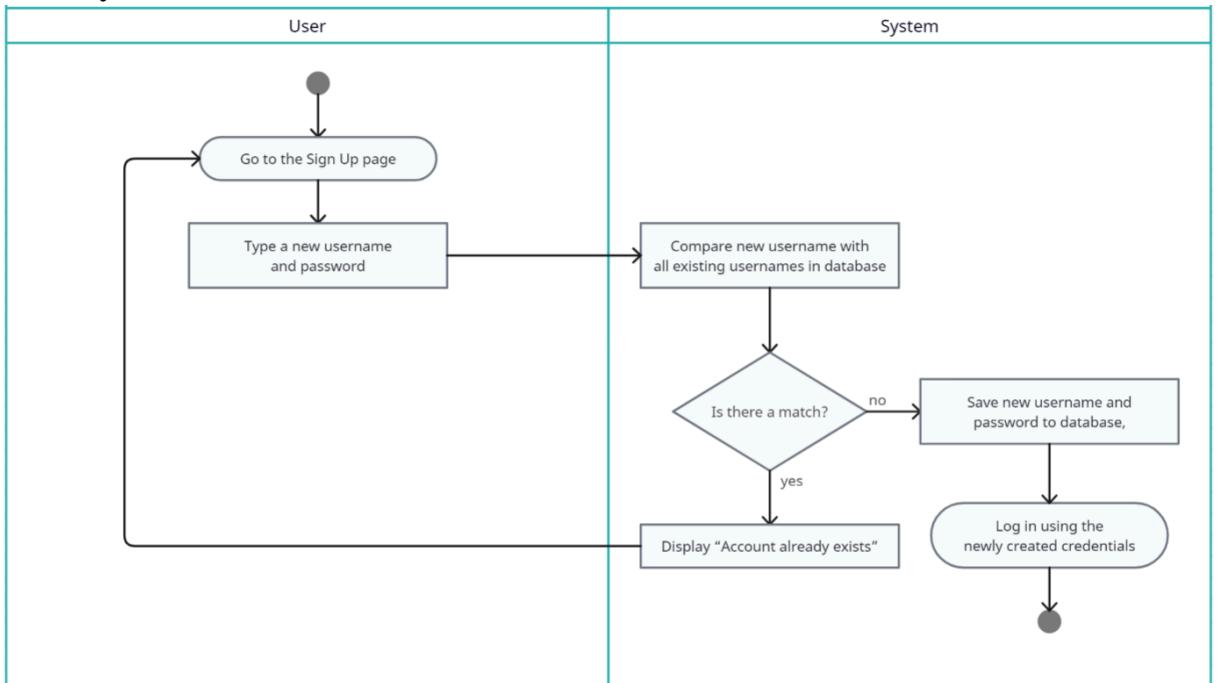


Figure 4: Activity Diagram - Create Account

Description: Short description for this activity diagram...

❖ Activity 4. View Profile

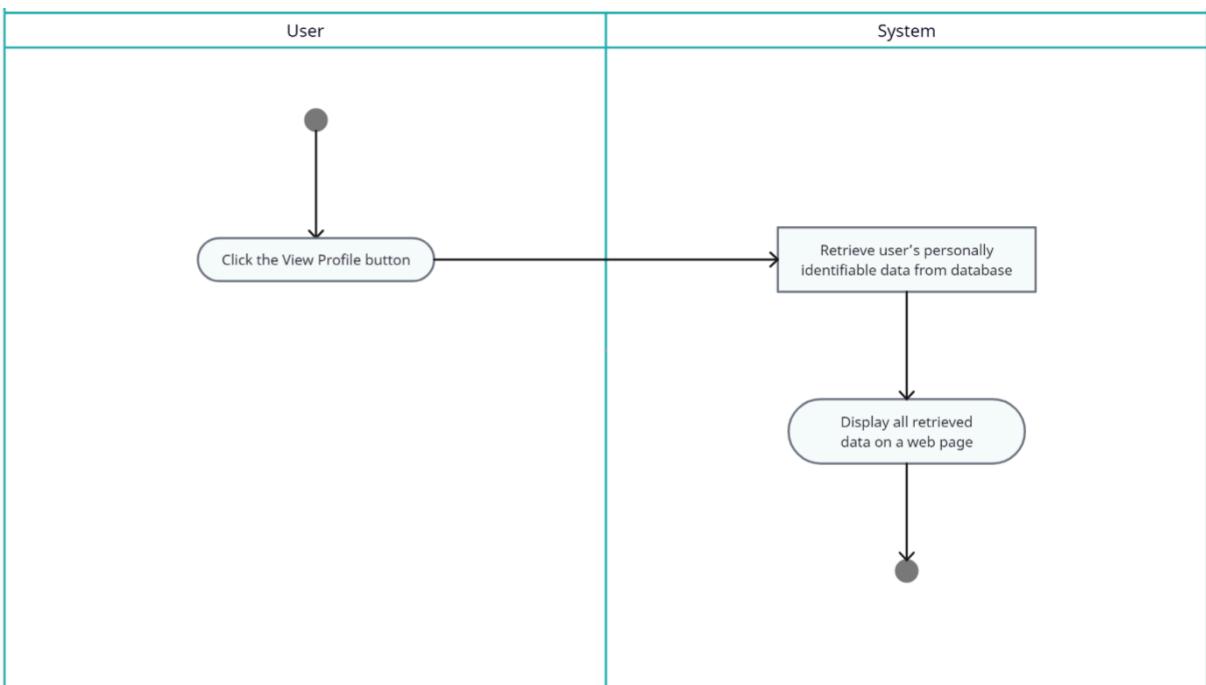


Figure 5: Activity Diagram - View Profile

Description: Short description for this activity diagram...

❖ **Activity 5. Start Workout Session**

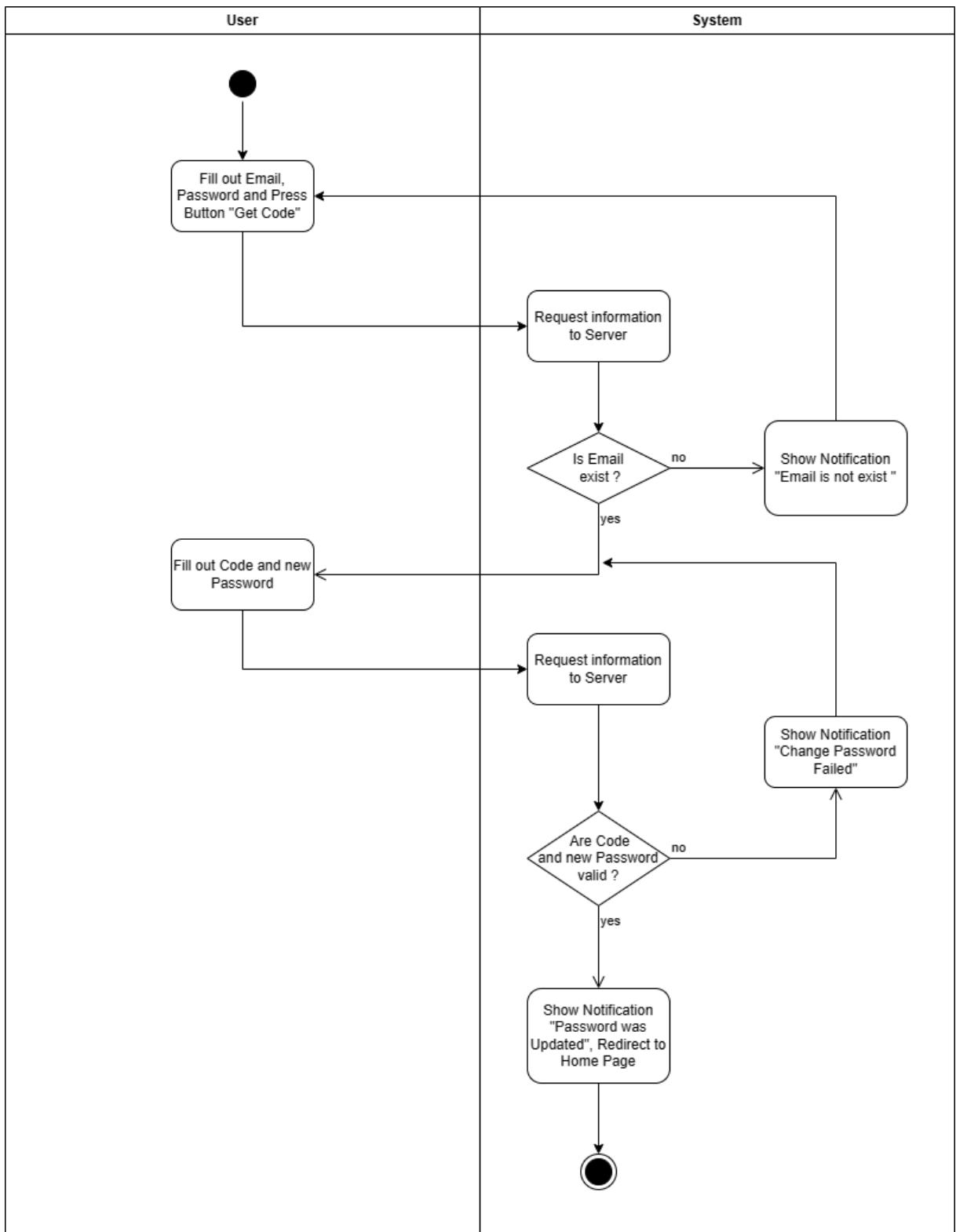


Figure 6: Activity Diagram - Start Workout Session

Description: Short description for this activity diagram...

❖ **Activity 6. Change Password**

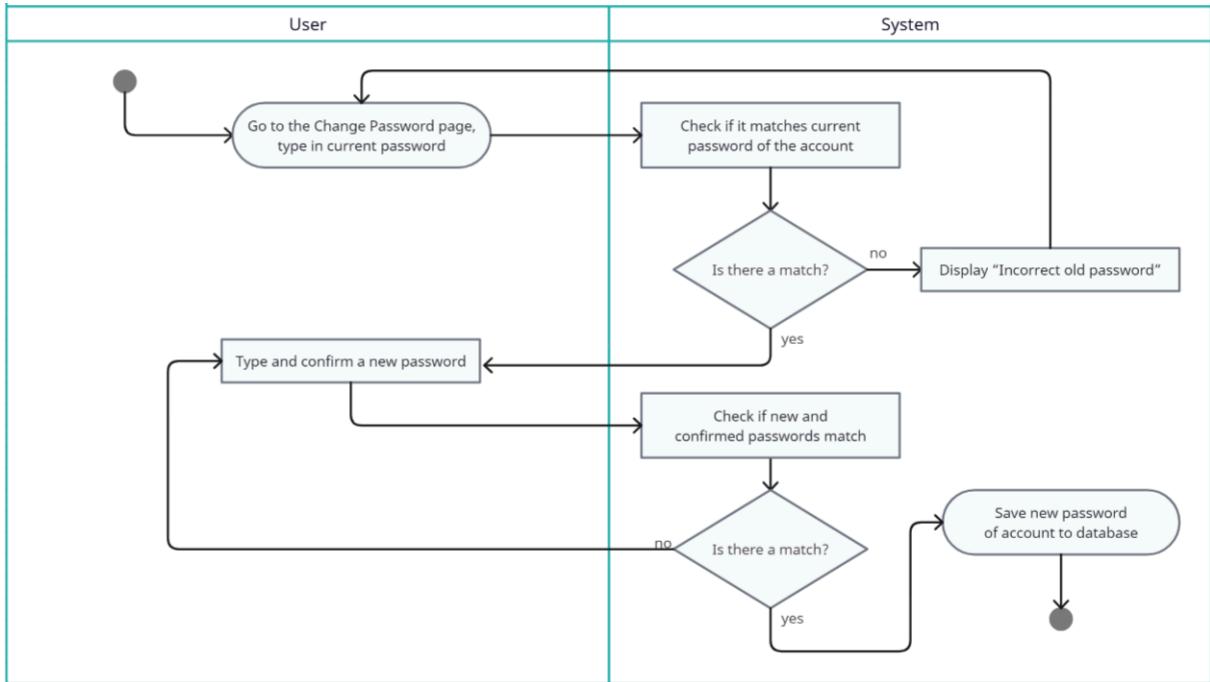


Figure 7: Activity Diagram - Change Password

Description: Short description for this activity diagram...

❖ **Activity 7. Update Profile**

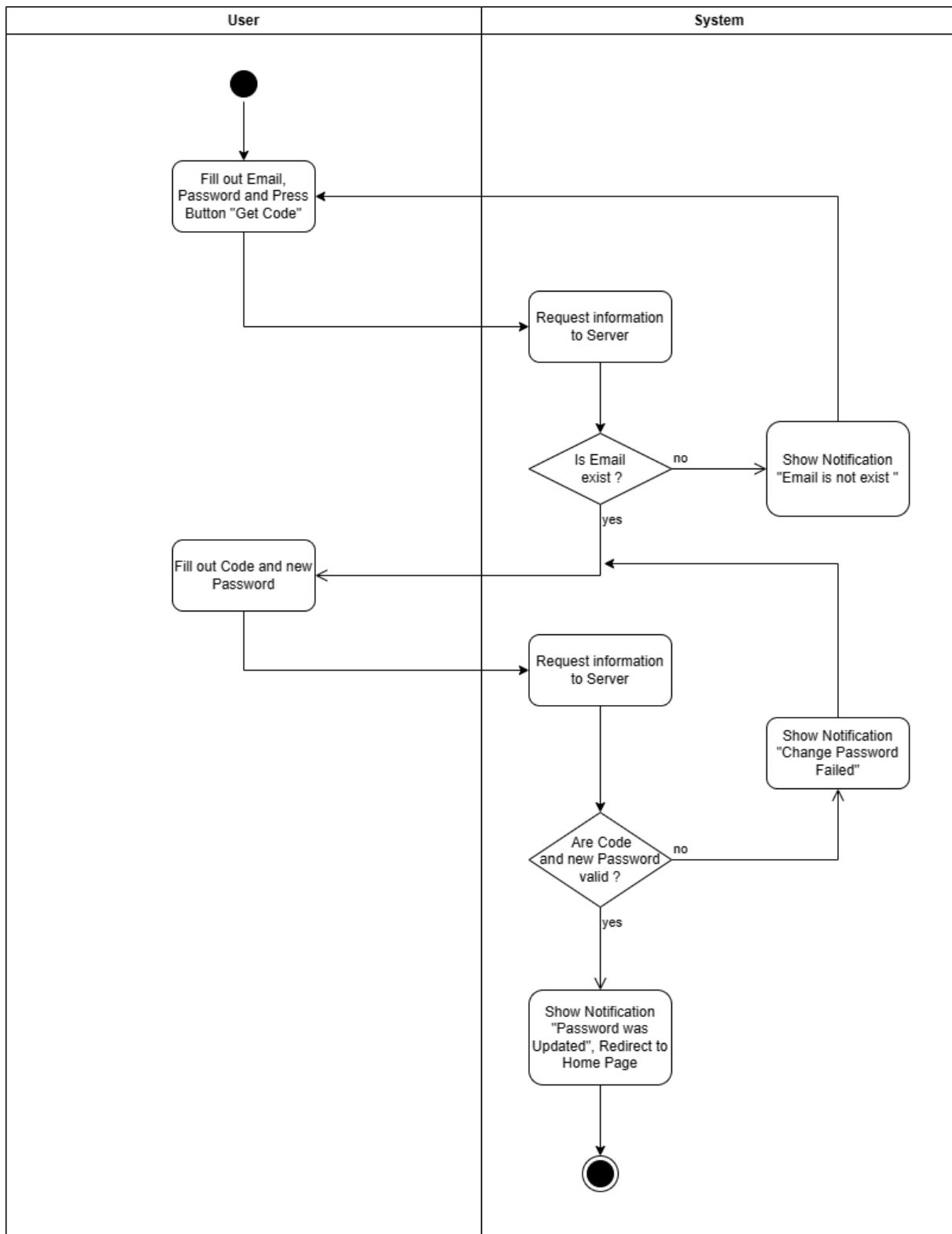


Figure 8: Activity Diagram - Update Profile

Description: Short description for this activity diagram...

❖ Activity 8. View Statistics

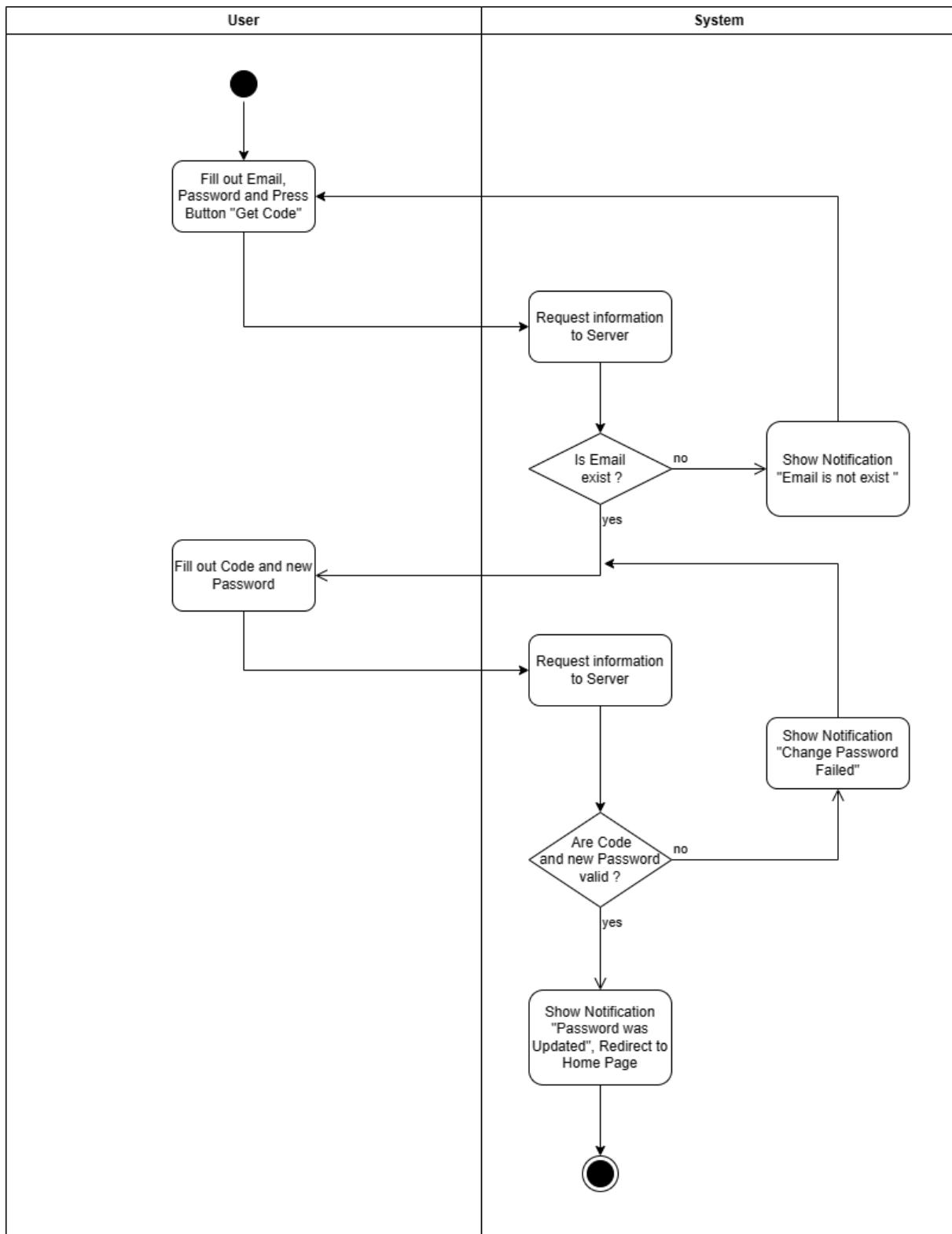


Figure 9: Activity Diagram - View Statistics

Description: Short description for this activity diagram...

VI. Class Diagram

The class diagram for the system is as follows:

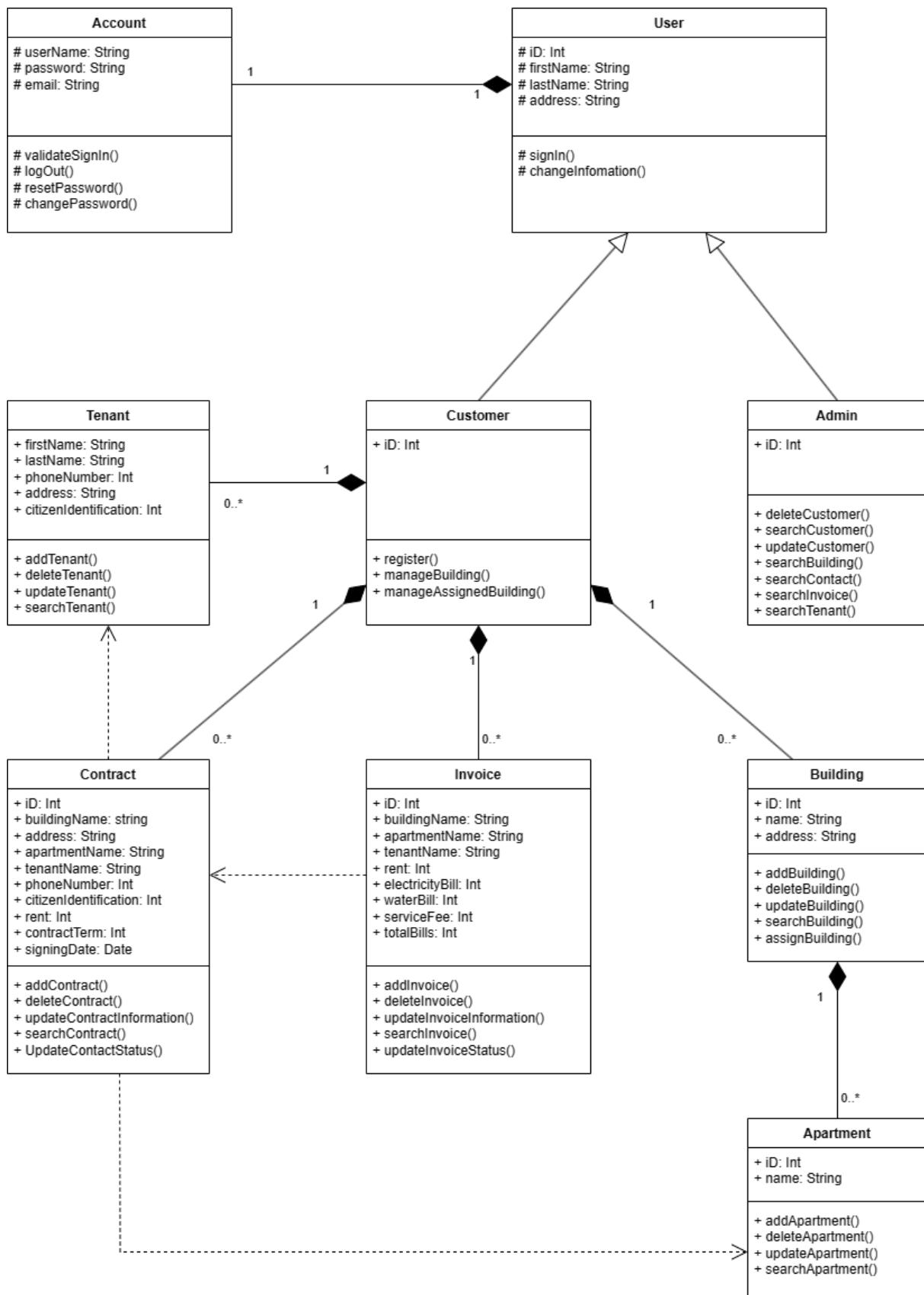


Figure 10: Class Diagram

Description: Short description for this class diagram...

VII. Sequence Diagram

❖ Sequence 1. Sign in

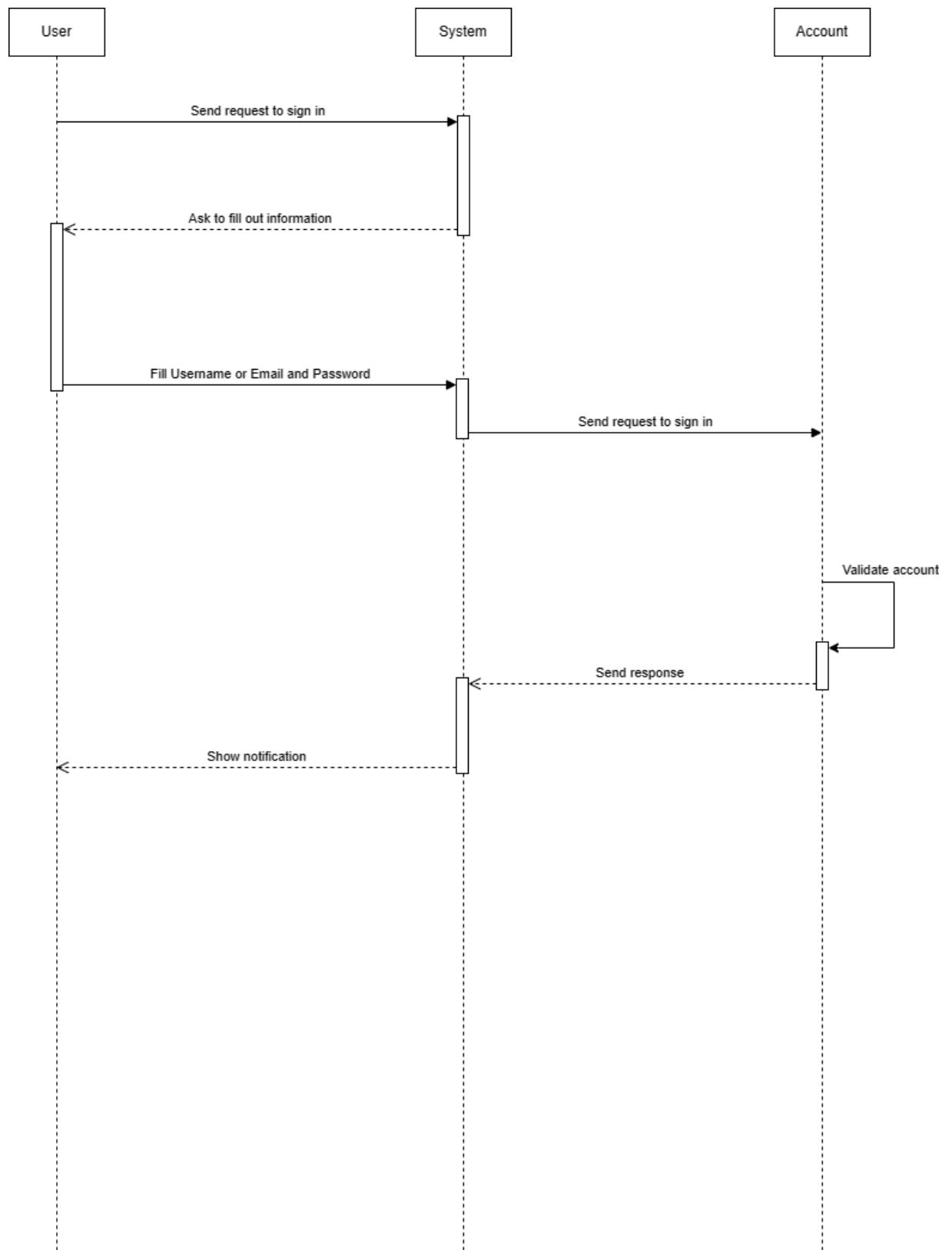


Figure 11: Sequence Diagram - Sign in

Description: Short description for this sequence diagram...

❖ **Sequence 2. Log out**

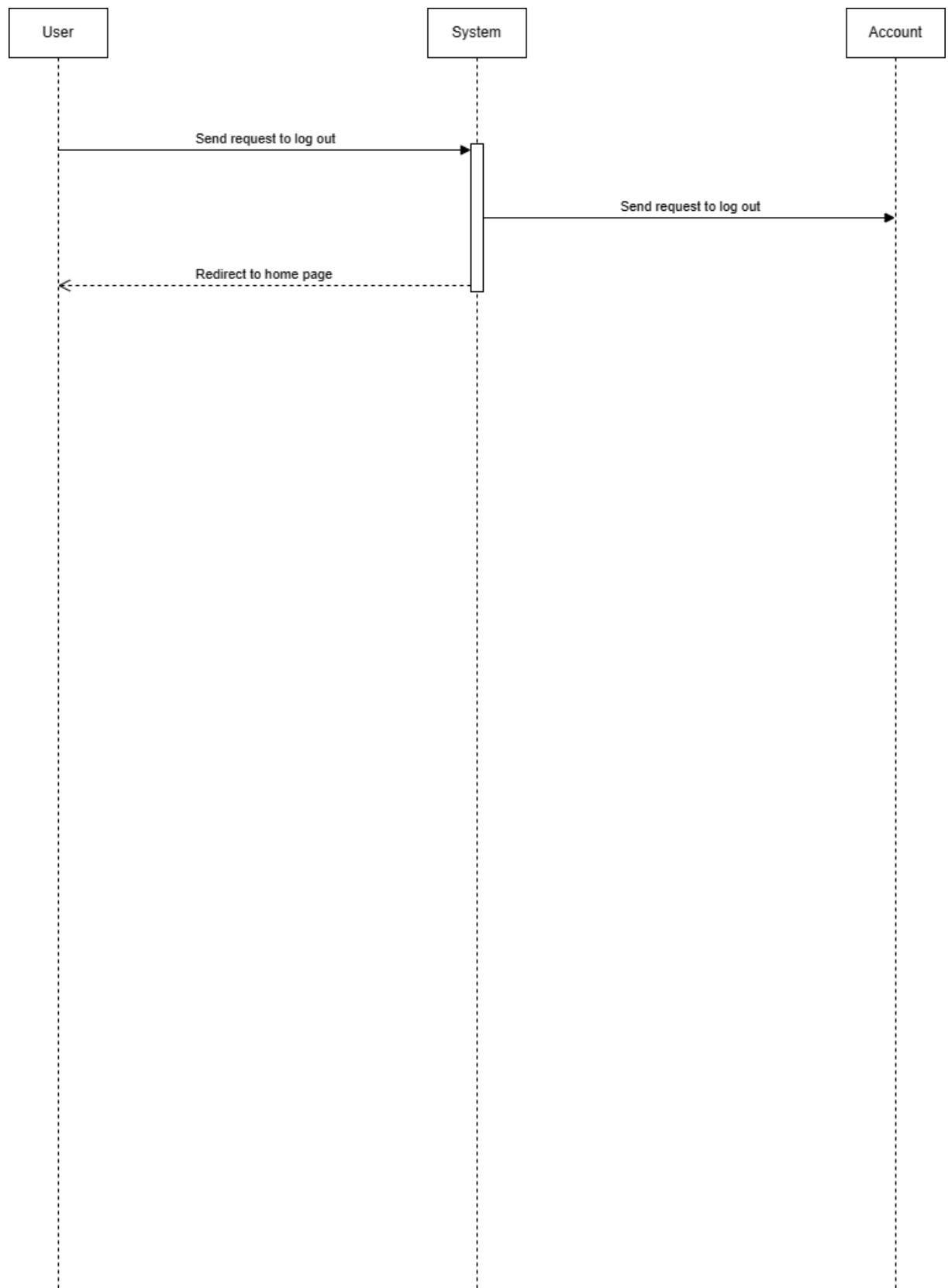


Figure 12: Sequence Diagram - Log out

Description: Short description for this sequence diagram...

❖ **Sequence 3. Reset Password**

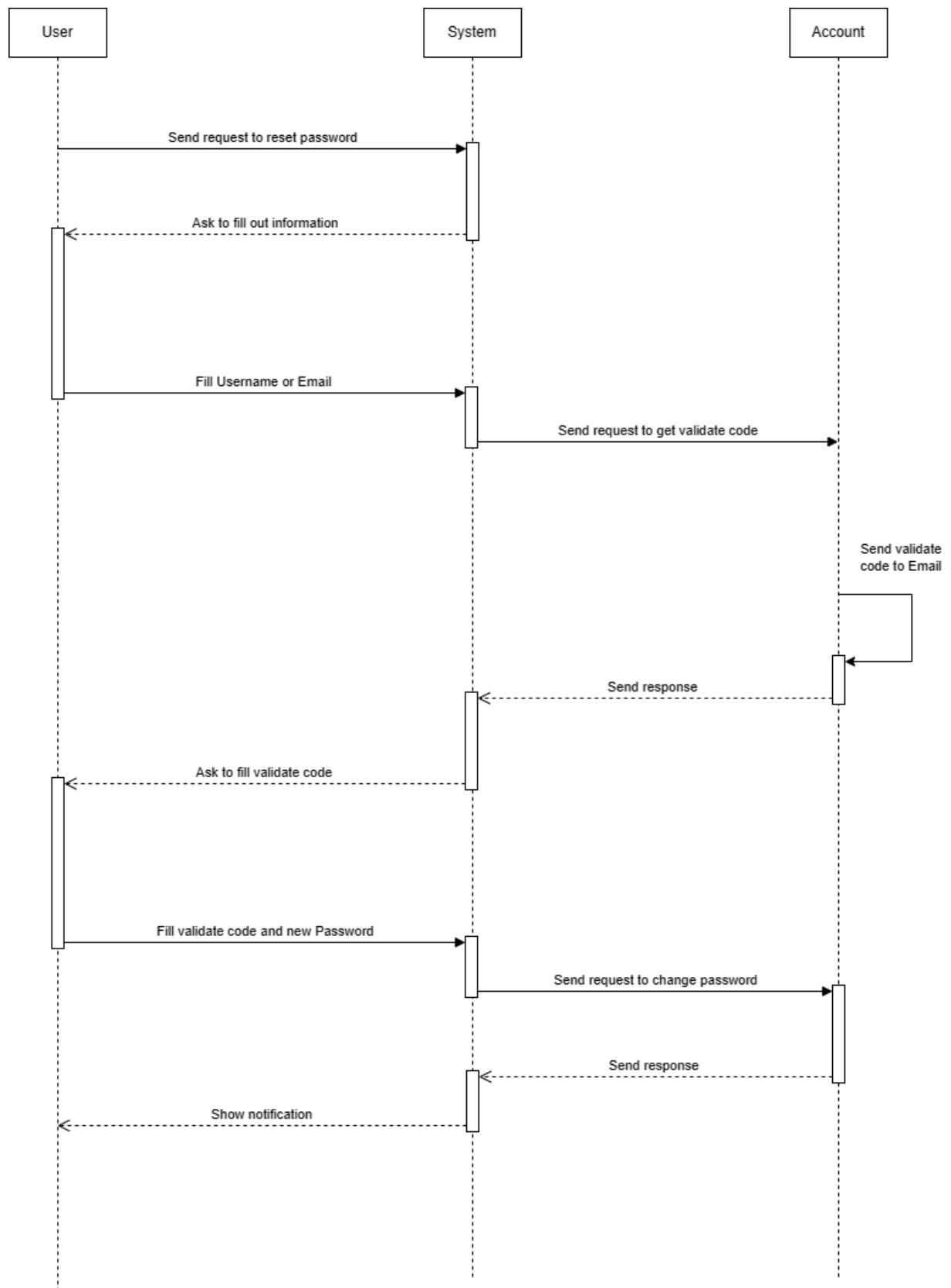


Figure 13: Sequence Diagram - Reset Password

Description: Short description for this sequence diagram...

CHAPTER 3 SYSTEM DESIGN

I. Architectural design

1. Introduction

Track Me Run is a web application that allows users to log workout sessions and manage their Calorie consumption. The web app utilizes Arduino Cloud server, Python Flask library for hooking to HTML pages, and Supabase remote server for managing external databases.

2. Architecture Overview

The web application follows a microservices architecture, allowing for scalable and independent deployment and addition of features.

3. System Components

Frontend:

- HTML templates coupled with custom CSS tweaks

Backend:

- Flask Session module for Python
- functools module for Python
- IoT API Client module for Python
- Time and Datetime modules for Python

Database:

- Supabase remote database
- Local database for backup

Third-Party Services:

- Arduino Cloud API for data sends by iPhone and storage on server
- Supabase for remote database management

4. Architecture Diagram

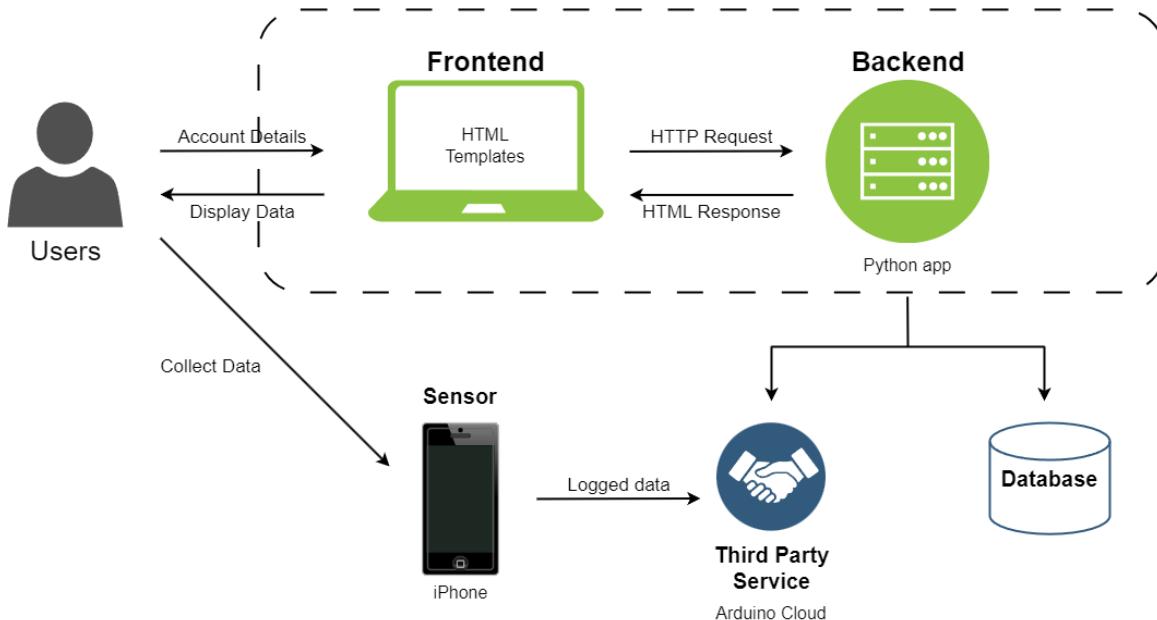


Figure 14: Architecture Diagram for Track Me Run

5. Detailed Component Descriptions

Frontend Components:

- HTML templates for different page navigation

Backend Components:

- Arduino_cloud.py: Handles communication with Arduino Cloud server
- app.py: Handles user program session

Database Schema:

- Users Table: Fields for user information
- Sessions Table: Fields for workout sessions details and status

6. Data Flow

Data flows from the web app UI to the backend via API calls, processed by the main Python app, and stored in the Supabase database.

7. Scalability Considerations

Web app features are designed in a modular fashion to enable future feature additions. These modules are toggled in the main function upon app start.

8. Performance Considerations

Database indexing for faster query performance

9. Deployment Architecture

A terminal environment is needed to execute the main process.

10. Technology Stack

Frontend: HTML, CSS

Backend: Python, Flask, Arduino Cloud API

11. Conclusion

The architecture design supports a scalable and high-performance web application, fulfilling the project's requirements effectively.

II. Database design

Database: SQLite

Database initialization:

```
CREATE TABLE users(id INTEGER PRIMARY KEY, name TEXT, username TEXT, password TEXT, weight INTEGER, height INTEGER, age INTEGER, gender TEXT, bmr INTEGER);
```

```
CREATE TABLE sessions(id INTEGER PRIMARY KEY, user_id INTEGER, datetime TEXT, duration FLOAT, distance FLOAT, avg_speed FLOAT, max_speed FLOAT, calories FLOAT, FOREIGN KEY (user_id) REFERENCES users(id));
```



Figure 15: Database Structure Diagram

The database is divided into two tables, Users table and Sessions table. The id field in Users serves as the primary key, which is automatically generated by enumeration when a new entry is created. This gives each user account a unique id number. The Users table also catalogs name, username, password, weight,

height, age, gender, and bmr values specific to each user in either numeric or text form.

At login of any user account, the `id` field is recorded by Flask and is used to retrieve all running records belonging to that account. The Flask app filters all Sessions entries that have a matching `user_id`. Those entries are then cached into a global dynamic memory variable `running_sessions` for operating use by the Flask web application. Various telemetry data saved in the Sessions table include `datetime`, `duration`, `distance`, `avg_speed`, `max_speed`, and `calories`.

Data examples:

- User example:

```
sqlite> select * from users;
1|Trương|Trương Tuấn Anh|1234|63|170|31|male|1570.4917628
```

- Session example:

```
sqlite> select * from sessions;
1|1|16-07-2024 11:39:59|61.9|123.76|1.998|3.224|7.57
2|1|16-07-2024 11:49:18|57.9|120.01|2.074|6.512|7.527
3|1|16-07-2024 11:54:03|62.9|169.27|2.691|5.363|10.78
```

III. UI design

Track Me Run utilizes a series of static and dynamic HTML pages to provide a web user interface. The pages that encompass this GUI are as follows:

HTML Page	Usage	Reference (s)	Login Required
<code>login.html</code>	for accepting login credentials	<i>Figure 16</i>	No
<code>signup.html</code>	for creating a new login credential, username and password	<i>Figure 17</i>	No
<code>fillprofile.html</code>	for inputting user personal measurements	<i>Figure 18</i>	Yes
<code>viewprofile.html</code>	for viewing user's personal information	<i>Figure 19</i>	Yes
<code>updateprofile.html</code>	for accepting new user's personal information	<i>Figure 20</i>	Yes
<code>home.html</code>	default directing page for displaying user's workout sessions	<i>Figure 21</i>	Yes

changepass.html	for accepting new password sets	<i>Figure 22</i>	Yes
setgoal.html	for setting distance and calorie goals before a workout session	<i>Figure 23</i>	Yes
startsession.html	for monitoring a live session and finishing that session	<i>Figure 24</i>	Yes
status.html	for displaying whether the user achieves their goals after a workout session	<i>Figure 25</i> and <i>Figure 26</i>	Yes

Table 9: UI Design - HTML pages

Every page, except for `login.html` and `signup.html`, requires a session token to access. A session token can only be obtained by logging in or signing up. Once the user has input a valid username and password, a `user_id` is saved into the `session` object. This object is checked at every `@login_required` decorator, and the function resorts to redirecting back to the login page if the session token is empty.

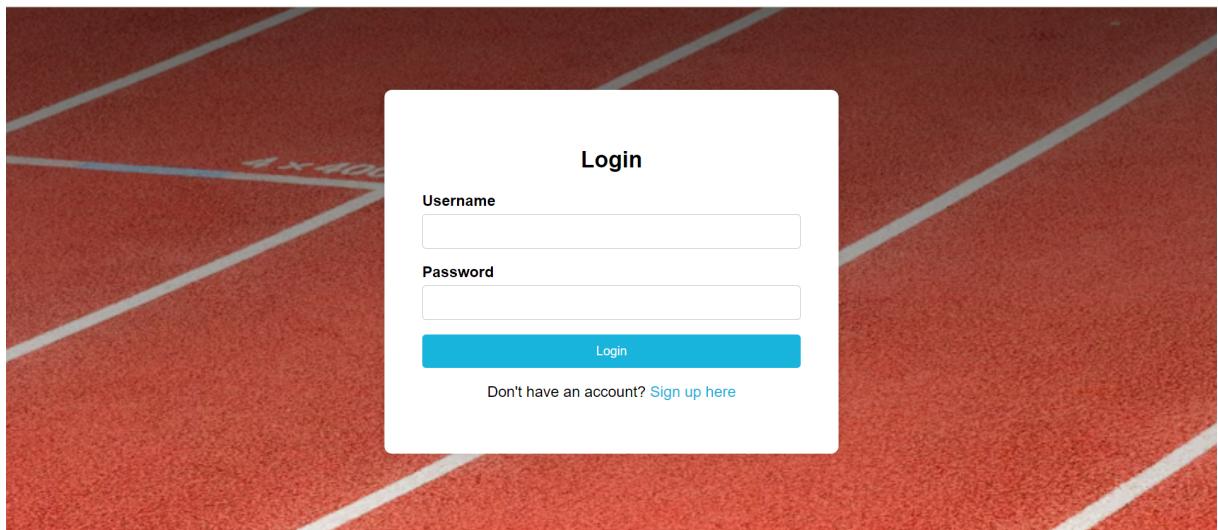


Figure 16: Login Interface

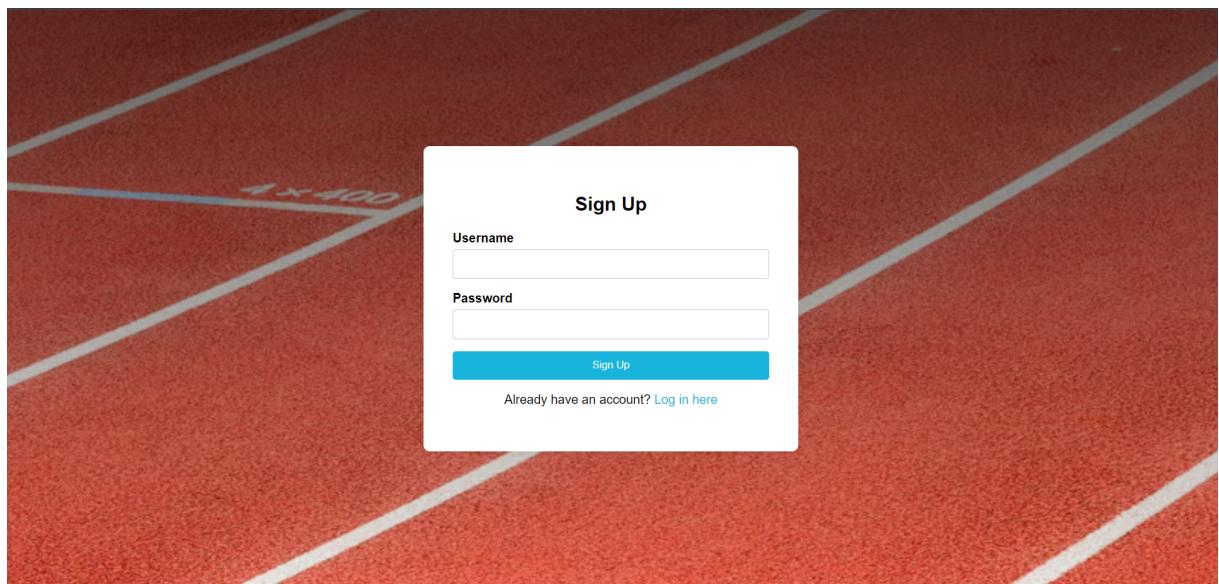


Figure 17: Sign up

A profile填写 form titled "Profile" is shown. It includes fields for Name, Weight (kg), Height (cm), Age, and Gender (Male selected). A "Submit" button is at the bottom.

Figure 18: Fill Profile

The screenshot shows the 'User Profile' section of the Track Me Run application. At the top, there is a navigation bar with links for 'Home', 'My Profile' (which is the active tab), 'Change Password', and 'Logout'. The main content area is titled 'User Profile' and contains fields for personal information:

- Name:** Truong Tuân Anh
- Weight (kg):** 63
- Height (cm):** 170
- Age:** 31
- Gender:** male
- BMR (kcal/day):** 1570.4917628

At the bottom of the form is a blue 'Update Profile' button.

Figure 19: User Profile

The screenshot shows the 'Update Profile' page of the Track Me Run application. At the top, there is a navigation bar with links for 'Home', 'My Profile' (which is the active tab), 'Change Password', and 'Logout'. The main content area is titled 'Update Profile' and contains fields for personal information, identical to Figure 19:

- Name:** Truong
- Weight (kg):** 63
- Height (cm):** 170
- Age:** 31

Below these fields is a 'Gender' section with radio buttons for 'Male' (selected) and 'Female'. At the bottom of the form is a blue 'Submit' button.

At the very bottom of the page, there is a footer bar with the text '© 2024 Track Me Run. All rights reserved.'

Figure 20: Update Profile

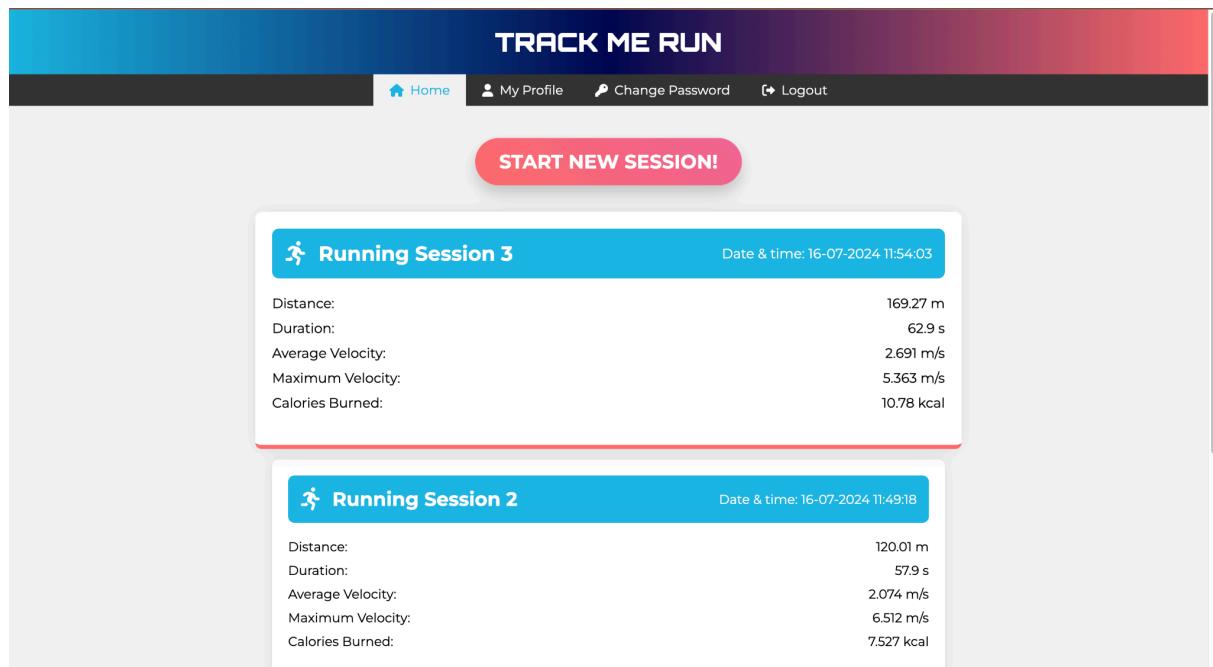


Figure 21: Home Page

The screenshot shows the "Change Password" page of the TRACK ME RUN application. At the top, there is a navigation bar with links for Home, My Profile, Change Password, and Logout. Below the navigation bar, a form titled "Update Password" is displayed.

Update Password

Current Password

New Password

Confirm New Password

Update Password

Figure 22: Change Password

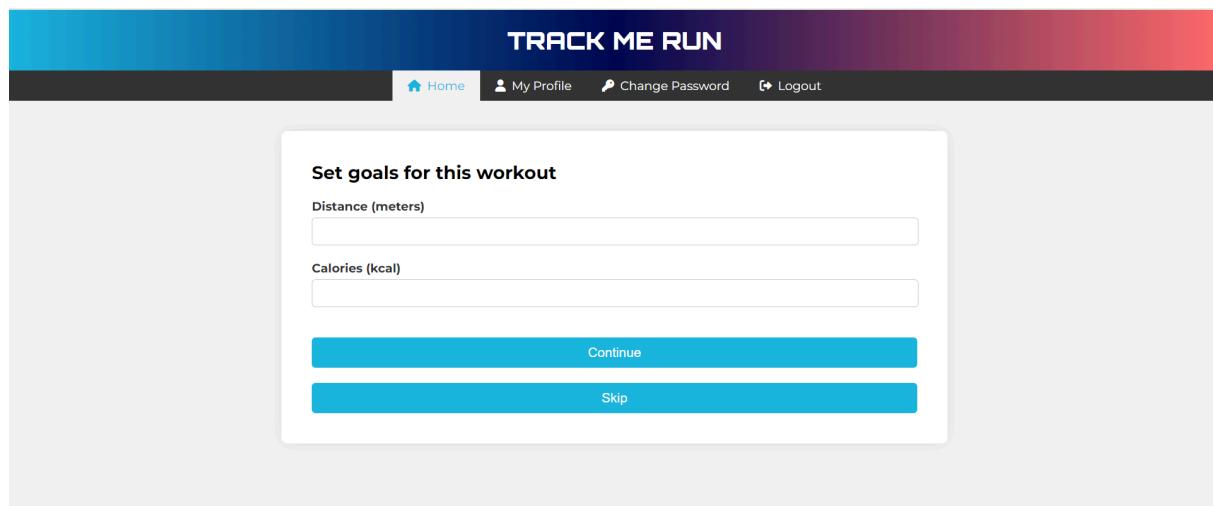


Figure 23: Set Goal

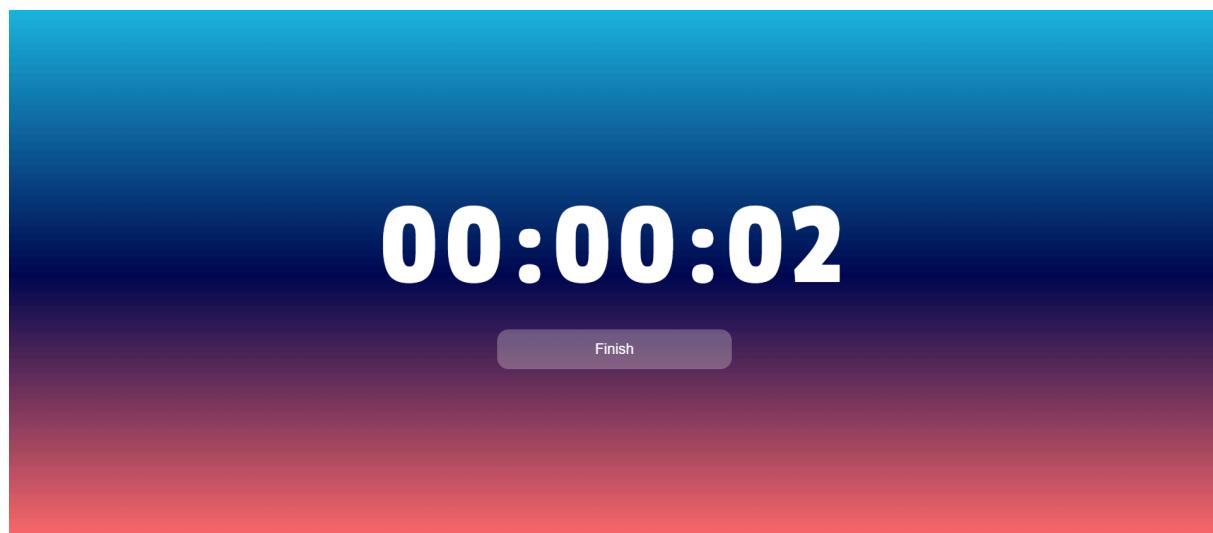


Figure 24: Start Session

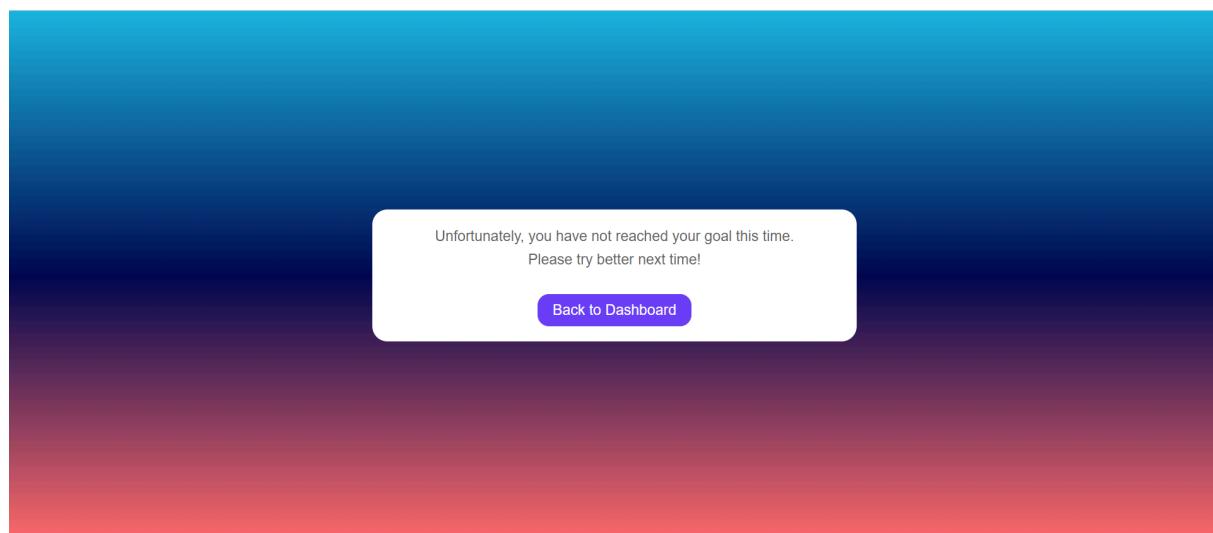


Figure 25: Incompleted Goal

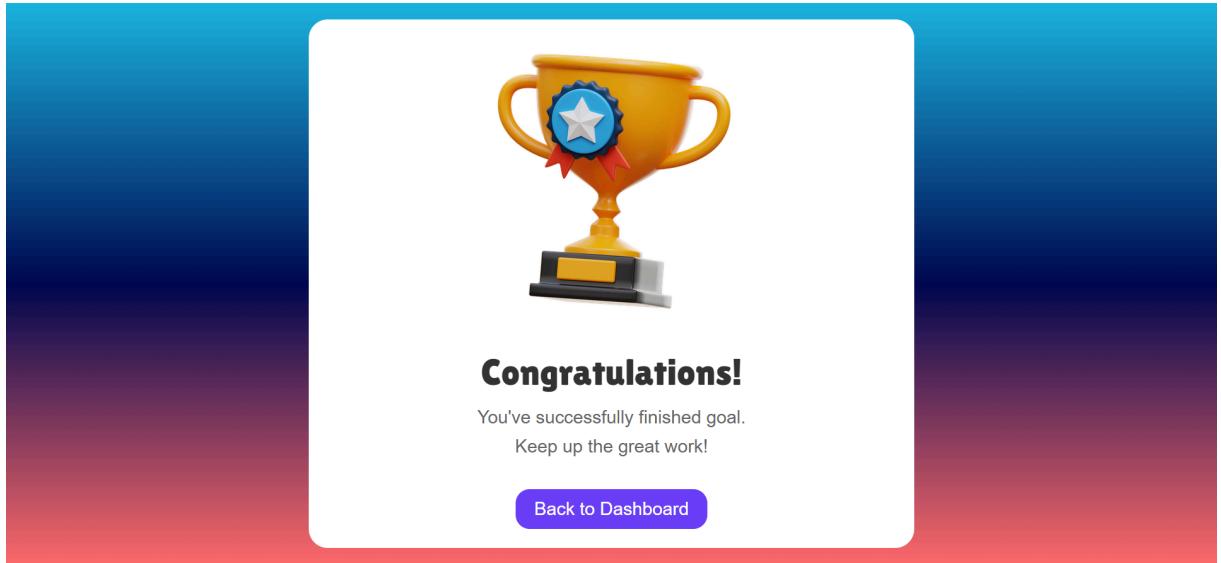


Figure 26: Completed Goal

CHAPTER 4 SYSTEM IMPLEMENTATION

CHAPTER 5 SYSTEM TESTING

CHAPTER 6 SYSTEM DEPLOYMENT

CHAPTER 7 CONCLUSIONS

REFERENCES

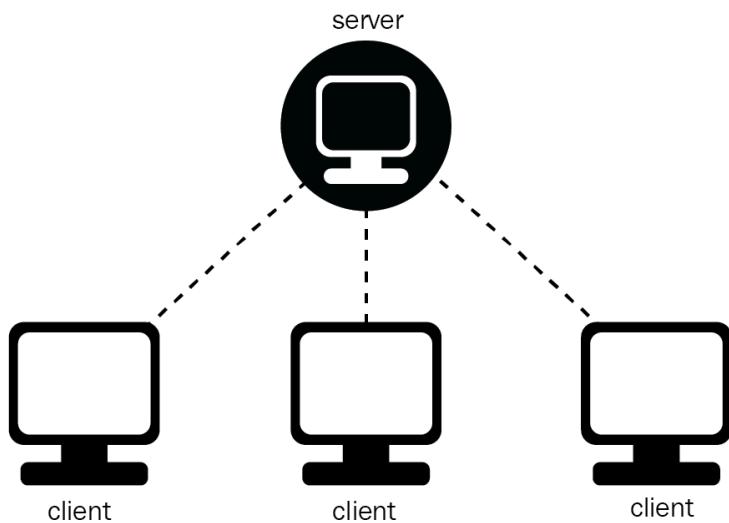
- [1] React. “LEARN REACT”. 2023, từ: <https://react.dev/learn/>
- [2] Redux. “API Reference”. 2023, từ: <https://redux.js.org/api/api-reference/>
- [3] Redux-Saga. “API Reference”. 2023, từ: <https://redux-saga.js.org/docs/api/>
- [4] PostgreSQL. “Documentation”. 2023, từ: <https://postgresql.org/docs/>
- [5] JSON Web Token. “Introduction”. 2023, từ: <https://jwt.io/introduction/>
- [6] Express. “API”. 2023, từ: [https://expressjs.com/en/5x/api.html/](https://expressjs.com/en/5x/api.html)

[7] W3school. “Learn to Code”. 2023, từ: <https://w3schools.com/>

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Mô hình mạng Client - Server

Mô hình Client Server là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.



Hình 2.1.1: Mô hình Client – Server

Nguyên tắc hoạt động

Trong mô hình Client Server, server chấp nhận tất cả các yêu cầu hợp lệ từ mọi nơi khác nhau trên Internet, sau đó trả kết quả về máy tính đã gửi yêu cầu đó

Máy tính được coi là máy khách khi chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ và đợi câu trả lời được gửi về.

Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn đó được gọi là giao thức. (Giao thức là gì thì các bạn có thể tham khảo tại đây) Một số giao thức được sử dụng phổ biến hiện nay như: HTTPS, TCP/IP, FTP,...

Nếu máy khách muốn lấy được thông tin từ máy chủ, chúng phải tuân theo một giao thức mà máy chủ đó đưa ra. Nếu yêu cầu đó được chấp nhận thì máy chủ sẽ thu thập thông tin và trả về kết quả cho máy khách yêu cầu. Bởi vì Server - máy chủ luôn luôn trong trạng thái sẵn sàng để nhận request từ client nên chỉ cần client gửi yêu cầu tín hiệu và chấp nhận yêu cầu đó thì server sẽ trả kết quả về phía client trong thời gian ngắn nhất.

Ưu điểm

Ưu điểm mô hình Client - Server bao gồm:

Giúp chúng ta có thể làm việc trên bất kì một máy tính nào có hỗ trợ giao thức truyền thông. Giao thức chuẩn này cũng giúp các nhà sản xuất tích hợp lên nhiều sản phẩm khác nhau mà không gặp phải khó khăn gì.

Có thể có nhiều server cùng làm một dịch vụ, chúng có thể nằm trên nhiều máy tính hoặc một máy tính.

Chỉ mang đặc điểm của phần mềm mà không hề liên quan đến phần cứng, ngoài yêu cầu duy nhất là server phải có cấu hình cao hơn các client.

Hỗ trợ người dùng nhiều dịch vụ đa dạng và sự tiện dụng bởi khả năng truy cập từ xa.

Cung cấp một nền tảng lý tưởng, cho phép cung cấp tích hợp các kỹ thuật hiện đại như mô hình thiết kế hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS).

Nhược điểm

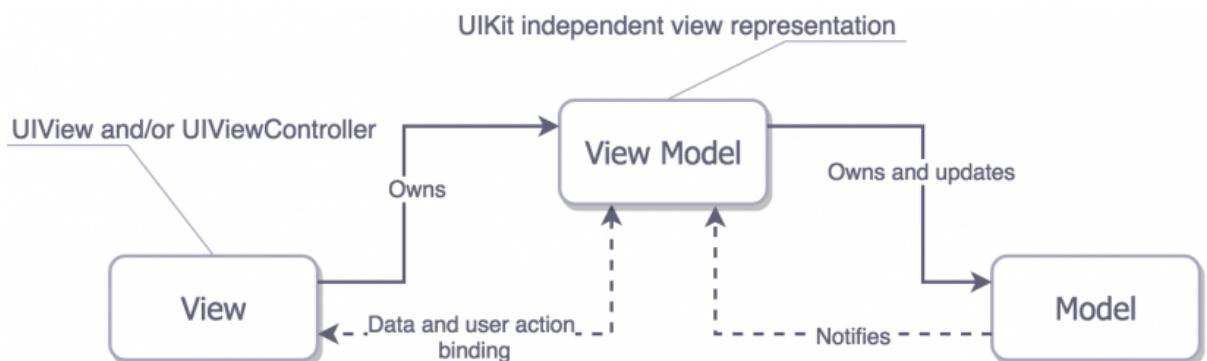
Nhược điểm mô hình Client - Server đó là:

Vấn đề bảo mật dữ liệu thông tin đôi khi còn chưa được an toàn lắm vì do phải trao đổi dữ liệu giữa 2 máy tính khác nhau ở 2 khu vực địa lý cách xa nhau. Và đây cũng nhược điểm duy nhất của mô hình này.

Tuy nhiên vấn đề này thì có một số giao thức đã hỗ trợ bảo mật dữ liệu khi truyền tải. Giao thức được sử dụng phổ biến như HTTPS.

Mô hình MVVM là một mô hình định nghĩa cấu trúc ứng dụng, được phát triển dựa trên kiến trúc của MVP. Mô hình này cho phép người dùng tách biệt giữa Model (dữ liệu), ViewModel (mã thực thi) và View (giao diện người dùng).

Khác với các mô hình truyền thống, mô hình MVVM không cho phép người dùng xử lý sự kiện Click và các hoạt động khác bằng cách viết mã trực tiếp trên một Button. Nguyên nhân là vì trong MVVM, các control như ListView, Button hay SearchBar,... không thể kết nối với dữ liệu một cách trực tiếp. Thay vào đó, bạn sẽ phải sử dụng thuộc tính Command (thuộc kiểu ICommand) để kết nối các hành động với ViewModel.



Hình 2.2.1: Mô hình Model - View - ViewModel (MVVM)

View

View được hiểu là thành phần giao diện mô tả dữ liệu của một ứng dụng, đồng thời đây cũng là thành phần duy nhất cho phép người dùng tương tác trong chương trình.

View trong MVVM được đánh giá là tích cực hơn so với các mô hình khác nhờ vào khả năng thực hiện hành vi và cung cấp phản hồi cho người dùng với một số tính năng nổi bật như Command, Binding,...

Model

Giống với mô hình MVC, Model là đối tượng cho phép bạn truy xuất dữ liệu và thao tác trên dữ liệu thật sự.

ViewModel

ViewModel là lớp trung gian của View với Model. Thành phần này tương đương với Controller trong mô hình MVC, có nhiệm vụ chứa mã lệnh cần thiết để thực hiện các lệnh Command, Data Binding,...

Cơ sở dữ liệu quan hệ

Cơ sở dữ liệu quan hệ là một loại cơ sở dữ liệu tổ chức dữ liệu thành một hoặc nhiều bảng hoặc quan hệ, mỗi bảng có một tên duy nhất và bao gồm một tập hợp các

hàng và cột. Dữ liệu trong cơ sở dữ liệu quan hệ được cấu trúc và tổ chức, giúp dễ dàng tìm kiếm, truy xuất và quản lý.

Dữ liệu thường được lưu trữ ở dạng chuẩn hóa trong cơ sở dữ liệu quan hệ. Dữ liệu được chia thành các bảng nhỏ hơn, có liên quan với nhau, mỗi bảng có khóa hoặc mã định danh duy nhất. Mỗi quan hệ giữa các bảng này được xác định thông qua việc sử dụng các khóa ngoại, khóa này liên kết dữ liệu trong một bảng với dữ liệu trong một bảng khác.

Cơ sở dữ liệu quan hệ được sử dụng rộng rãi trong các ứng dụng khác nhau, bao gồm hệ thống kinh doanh và tài chính, nghiên cứu khoa học và thương mại điện tử. Chúng cung cấp một cách linh hoạt và có thể mở rộng để lưu trữ và quản lý lượng lớn dữ liệu trong khi vẫn đảm bảo tính toàn vẹn và nhất quán của dữ liệu thông qua các ràng buộc, chẳng hạn như khóa chính và khóa ngoại.

Cấu trúc

Cơ sở dữ liệu quan hệ được cấu trúc bằng cách sử dụng bảng, còn được gọi là quan hệ. Mỗi bảng bao gồm các hàng và cột, với mỗi hàng đại diện cho một bản ghi hoặc phiên bản dữ liệu và mỗi cột đại diện cho một thuộc tính hoặc trường cụ thể của dữ liệu. Một tập hợp các thuộc tính hoặc kiểu dữ liệu, chẳng hạn như văn bản, số, ngày hoặc Boolean, xác định các cột trong bảng. Mỗi cột cũng có một tên duy nhất, giúp xác định kiểu dữ liệu được lưu trữ trong cột đó.

Các hàng trong bảng đại diện cho các bản ghi hoặc phiên bản dữ liệu riêng lẻ. Mỗi hàng có một mã định danh duy nhất, được gọi là khóa chính. Khóa chính được sử dụng để liên kết các bản ghi trên các bảng khác nhau trong cơ sở dữ liệu. Mỗi quan hệ giữa các bảng trong cơ sở dữ liệu quan hệ được xác định bằng khóa ngoại. Khóa ngoại là một cột trong một bảng tham chiếu đến khóa chính của một bảng khác. Điều này cho phép dữ liệu liên quan được liên kết và truy cập từ các bảng khác nhau trong cơ sở dữ liệu.

Ngoài các bảng, cơ sở dữ liệu quan hệ cũng sử dụng các ràng buộc để đảm bảo tính toàn vẹn và nhất quán của dữ liệu. Các ràng buộc là các quy tắc hoặc điều kiện phải được đáp ứng trước khi dữ liệu có thể được chèn, cập nhật hoặc xóa khỏi cơ sở dữ liệu. Ví dụ về các ràng buộc bao gồm khóa chính, khóa ngoại, ràng buộc duy nhất và ràng buộc kiểm tra.

Mô hình quan hệ

Mô hình quan hệ là một mô hình dữ liệu được sử dụng để thiết kế và quản lý dữ liệu trong cơ sở dữ liệu quan hệ. Mô hình quan hệ được Edgar F. Codd giới thiệu vào năm 1970 và kể từ đó nó đã trở thành mô hình dữ liệu được sử dụng rộng rãi nhất cho cơ sở dữ liệu hiện đại.

Mô hình quan hệ dựa trên khái niệm bảng, còn được gọi là quan hệ. Mỗi bảng trong cơ sở dữ liệu đại diện cho một tập hợp dữ liệu có liên quan và mỗi hàng trong bảng đại diện cho một bản ghi hoặc phiên bản của dữ liệu đó. Mỗi cột trong bảng đại diện cho một thuộc tính hoặc trường cụ thể của dữ liệu.

Mỗi quan hệ giữa các bảng trong cơ sở dữ liệu được xác định bằng các khóa. Khóa chính là một cột hoặc tập hợp các cột trong bảng xác định duy nhất từng hàng trong bảng đó. Khóa ngoại là một cột trong một bảng tham chiếu đến khóa chính của một bảng khác, cho phép dữ liệu liên quan được liên kết trên các bảng khác nhau trong cơ sở dữ liệu.

Mô hình quan hệ cũng hỗ trợ các hoạt động truy vấn và thao tác dữ liệu trong cơ sở dữ liệu, chẳng hạn như CHỌN, CHÈN, CẬP NHẬT và XÓA. Các thao tác này được thực hiện bằng một ngôn ngữ đặc biệt gọi là Ngôn ngữ truy vấn có cấu trúc (SQL), ngôn ngữ này xác định các truy vấn và câu lệnh tương tác với cơ sở dữ liệu.

Một trong những lợi ích chính của mô hình quan hệ là tính linh hoạt và khả năng mở rộng của nó. Các bảng có thể được thêm, xóa hoặc sửa đổi để phù hợp với các yêu cầu dữ liệu đang thay đổi và mối quan hệ giữa các bảng có thể được xác định hoặc cập nhật dễ dàng khi cần. Ngoài ra, mô hình quan hệ cung cấp một cách tổ chức và quản lý dữ liệu nhất quán và chuẩn hóa, giúp dễ dàng duy trì và cập nhật cơ sở dữ liệu lớn và phức tạp theo thời gian.

Lợi ích của hệ quản trị cơ sở dữ liệu quan hệ

Hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) mang lại nhiều lợi ích, một số lợi ích bao gồm:

Tính toàn vẹn của dữ liệu : RDBMS sử dụng nhiều ràng buộc khác nhau như khóa chính, khóa ngoại và ràng buộc kiểm tra để đảm bảo tính toàn vẹn của dữ liệu, giúp duy trì tính chính xác và nhất quán của dữ liệu.

Khả năng mở rộng : RDBMS có thể xử lý một lượng lớn dữ liệu và có thể dễ dàng tăng hoặc giảm quy mô khi cần. Họ cũng có thể hỗ trợ đồng thời nhiều người dùng và ứng dụng.

Tính linh hoạt : RDBMS cung cấp một cách linh hoạt để tổ chức và lưu trữ dữ liệu, vì các bảng có thể được thêm, xóa hoặc sửa đổi để đáp ứng các yêu cầu dữ liệu thay đổi.

Dễ sử dụng : Ngôn ngữ SQL được sử dụng trong RDBMS rất dễ học và dễ sử dụng, đồng thời cung cấp một cách tiêu chuẩn và nhất quán để tương tác với cơ sở dữ liệu.

Bảo mật dữ liệu : RDBMS cung cấp các tính năng bảo mật tích hợp như kiểm soát truy cập và xác thực người dùng để đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập và sửa đổi dữ liệu.

Tính nhất quán của dữ liệu : RDBMS sử dụng các giao dịch để đảm bảo rằng dữ liệu nhất quán và đáng tin cậy, ngay cả khi hệ thống bị lỗi hoặc gián đoạn.

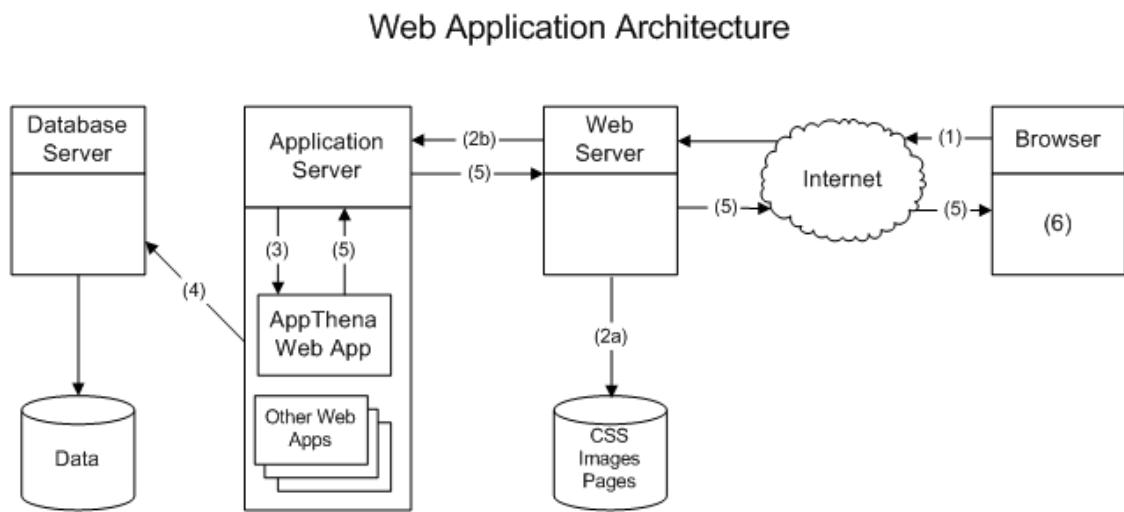
Chia sẻ dữ liệu : RDBMS có thể chia sẻ dữ liệu trên các ứng dụng và nền tảng khác nhau, cải thiện sự hợp tác và năng suất trong các tổ chức.

RDBMS cung cấp một cách mạnh mẽ và đáng tin cậy để quản lý dữ liệu và được sử dụng rộng rãi trong nhiều ứng dụng, bao gồm các hệ thống kinh doanh và tài chính, nghiên cứu khoa học và thương mại điện tử.

Kiến trúc hệ thống

Kiến trúc ứng dụng Website (Website application architecture) mô tả sự tương tác giữa các ứng dụng, cơ sở dữ liệu (database) và các hệ thống phần mềm trung gian

trên web. Kiến trúc này đảm bảo nhiều ứng dụng có thể hoạt động đồng thời và trơn tru.



Hình 2.4.1: Kiến trúc ứng dụng Website

Cách hoạt động

Với bất kỳ ứng dụng Website điển hình nào, bạn cũng sẽ cần hai mã (chương trình con) khác nhau chạy song song. Đó là:

Code từ phía máy khách (frontend): Đoạn code này nằm trong Browser và phản hồi input (đầu vào) của người dùng.

Code phía máy chủ (backend): Code nằm trên máy chủ và phản hồi các yêu cầu HTTP

Một nhà phát triển web sẽ quyết định xem mã trên phía máy chủ sẽ làm gì với mã trong trình duyệt - phía máy khách. Để viết mã từ phía máy chủ, bạn sẽ cần sử dụng các ngôn ngữ lập trình như C#, Java, JavaScript, Python, PHP, Ruby,...

Bất kỳ code nào có thể đáp ứng các yêu cầu HTTP đều có khả năng chạy trên phía máy chủ. Mã phía máy chủ chịu trách nhiệm tạo trang mà người dùng yêu cầu cũng như lưu trữ các loại dữ liệu khác nhau, bao gồm hồ sơ người dùng và đầu vào của họ.

Sự kết hợp của CSS, HTML, JavaScript được sử dụng để viết mã phía máy khách. Mã này sẽ được trình duyệt website phân tích cú pháp. Không giống mã phía máy chủ, mã phía máy khách có thể được người dùng nhìn thấy và chỉnh sửa. Mã phía máy khách chỉ giao tiếp thông qua các yêu cầu HTTP và không thể đọc trực tiếp các tệp từ máy chủ.

Các thành phần

Khi nhắc đến các thành phần web application, chúng ta có thể hiểu đó là bất kỳ phần nào trong số 2 thành phần sau:

Thành phần UI/UX của ứng dụng Website: Bao gồm nhật ký hoạt động, trang tổng quát, thông báo, cài đặt, thống kê,... Các thành phần này không liên quan gì đến hoạt động của kiến trúc ứng dụng web. Thay vào đó, chúng là một phần của sơ đồ bố cục giao diện của ứng dụng web.

Các thành phần cấu trúc: Hai thành phần cấu trúc chính của Website là phía máy khách (Frontend) và máy chủ (Backend):

Thành phần phía máy khách: Các component phía máy khách (frontend) được phát triển trong CSS, HTML và JS. Vì nó tồn tại trong trình duyệt web của người dùng nên không cần có những điều chỉnh liên quan đến hệ điều hành hoặc thiết bị. Phần frontend đại diện cho chức năng của ứng dụng web mà người dùng tương tác cùng.

Thành phần phía máy chủ: Phần backend có thể được xây dựng bằng cách sử dụng một hoặc kết hợp nhiều ngôn ngữ lập trình cùng các framework, bao gồm các ngôn ngữ như Java, .NET, NodeJS, PHP, Python, Ruby on Rails,... Component của máy chủ phải có ít nhất hai thành phần là logic ứng dụng và cơ sở dữ liệu. Logic ứng dụng là trung tâm điều khiển chính của ứng dụng web trong khi cơ sở dữ liệu là nơi lưu trữ tất cả các dữ liệu một cách liên tục.

Ngôn ngữ **HTML**

HTML hay HyperText Markup Language - Ngôn ngữ đánh dấu siêu văn bản, là ngôn ngữ được sử dụng cho các tài liệu web.

Trên thực tế, đây không phải là một ngôn ngữ lập trình, HTML giống như một ngôn ngữ xác định đâu là ý nghĩa, mục đích và cấu trúc của một tài liệu. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các website.

Vai trò của HTML

Một website thường chứa nhiều trang con. Mỗi trang con sẽ có một tập tin HTML riêng. Dù bạn lập trình bằng bất cứ ngôn ngữ nào, trên bất cứ Framework nào, khi chạy trên nền website, chúng đều được biên dịch ra ngôn ngữ HTML. Đa số các trình soạn thảo văn bản trên website đều có 2 chế độ xem: HTML và văn bản thường.

Chức năng chính của HTML là xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc. HTML thường được dùng để phân chia các đoạn văn, heading, links, blockquotes,...

Với HTML, bạn sẽ có thể:

Thêm tiêu đề, định dạng đoạn văn, ngắt dòng điều khiển.

Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh, tạo liên kết.

Xây dựng bảng, điều khiển một số kiểu mẫu

Như đã đề cập ở trên, HTML không phải là ngôn ngữ lập trình. Do đó, HTML cũng không thể tạo ra các chức năng “động” được. Hiểu đơn giản, HTML giống như Microsoft Word, dùng để bố cục và định dạng trang web.

Tóm lại, HTML là ngôn ngữ markup, dễ học, dễ hiểu, dễ áp dụng. Tuy nhiên, một website được viết bằng HTML rất đơn giản, nếu không muốn nói là nhảm chán. Để gây hứng thú với người truy cập, website cần có sự hỗ trợ của CSS và JavaScript.

Nếu HTML cung cấp cấu trúc cơ bản của các trang web thì CSS và JavaScript sẽ cải tiến và sửa đổi website sinh động hơn.

CSS

CSS là viết tắt của cụm từ “Cascading Style Sheets”, tạm dịch: ngôn ngữ tạo phong cách cho trang web. Ngôn ngữ lập trình này quy định cách các thành phần HTML của trang web thực sự sẽ xuất hiện trên frontend như thế nào.

Hiểu đơn giản, CSS sẽ giúp webmaster xác định styles và định nghĩa nhiều loại nội dung của website. CSS được tạo ra để kết hợp với ngôn ngữ markup HTML nhằm tạo nên phong cách cho trang web.

Có 3 loại style CSS chính:

Style CSS Internal: là style được tải lên mỗi khi trang web được refresh.

Style CSS Inline: bạn có thể chỉnh sửa một yếu tố nào đó mà không cần truy cập trực tiếp vào file CSS.

External style: bạn có thể tạo phong cách ở file khác áp dụng CSS vào trang bạn muốn. External style sẽ cải thiện thời gian tải trang rất nhiều.

Đặc biệt, bạn có thể sử dụng External Style CSS để tạo phong cách cho nhiều trang cùng lúc.

Vai trò của CSS

Tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như HTML.

Tiết kiệm công sức của lập trình viên nhờ điều khiển định dạng của nhiều website

Phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc và font chữ.

Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một website thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn.

Javascript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi

nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

Vai trò của Javascript

Trước đây, các trang web có dạng tĩnh, tương tự như các trang trong một cuốn sách. Một trang tĩnh chủ yếu hiển thị thông tin theo một bố cục cố định và không làm được mọi thứ mà chúng ta mong đợi như ở một trang web hiện đại. JavaScript dần được biết đến như một công nghệ phía trình duyệt để làm cho các ứng dụng web linh hoạt hơn. Sử dụng JavaScript, các trình duyệt có thể phản hồi tương tác của người dùng và thay đổi bố cục của nội dung trên trang web.

Khi ngôn ngữ này phát triển hoàn thiện, các nhà phát triển JavaScript đã thiết lập các thư viện, khung và cách thức lập trình cũng như bắt đầu sử dụng ngôn ngữ này bên ngoài trình duyệt web. Ngày nay, bạn có thể sử dụng JavaScript để thực hiện hoạt động phát triển cả ở phía máy khách và máy chủ:

JavaScript phía máy khách

JavaScript phía máy khách để cập đến cách thức JavaScript hoạt động trong trình duyệt của bạn. Trong trường hợp này, công cụ JavaScript nằm trong mã trình duyệt. Tất cả các trình duyệt web phổ biến đều có các công cụ JavaScript tích hợp riêng.

Các nhà phát triển ứng dụng web viết mã JavaScript với các hàm khác nhau được liên kết với nhiều loại sự kiện, chẳng hạn như nhấp chuột hoặc khi di chuột đến. Các hàm này thực hiện các thay đổi đối với HTML và CSS.

JavaScript phía máy chủ

JavaScript phía máy chủ để cập đến việc sử dụng ngôn ngữ viết mã trong logic máy chủ back-end. Trong trường hợp này, công cụ JavaScript nằm trực tiếp trên máy chủ. Hàm JavaScript phía máy chủ có thể truy cập cơ sở dữ liệu, thực hiện các phép toán logic khác nhau và phản hồi nhiều loại sự kiện do hệ điều hành của máy chủ kích hoạt. Ưu điểm chính của viết tập lệnh phía máy chủ là bạn có khả năng tùy chỉnh linh hoạt phản hồi của trang web dựa trên yêu cầu và quyền truy cập của mình cũng như theo yêu cầu thông tin từ trang web.

Công nghệ sử dụng

Node.js

Node.js là một platform (nền tảng) độc lập được xây dựng trên môi trường Javascript Runtime - trình thông dịch ngôn ngữ Javascript cực nhanh trên trình duyệt

Chrome. Dựa trên Node.js, chúng ta có thể xây dựng các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Node.js được thiết kế bởi Ryan Lienhart Dahl - một kỹ sư phần mềm người Mỹ vào năm 2009 và phát triển dưới sự bảo trợ của Joyent.

Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Các ứng dụng được tạo ra bởi Node.js có tốc độ xử lý nhanh, realtime thời gian thực.

Node.js sẽ thích hợp cho các sản phẩm có lưu lượng truy cập lớn, cần sự thay đổi về công nghệ và cần được mở rộng hoặc tạo ra các dự án Startup một cách nhanh nhất có thể.

Vai trò của Node.js

Các ứng dụng Nodejs được viết bằng javascript, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ Javascript, Ryan Dahl: “Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”

Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.

Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac - Window - Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí. Các bạn có thể thấy cộng đồng Nodejs lớn như thế nào tại đây, các package đều hoàn toàn free.

Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

Express

Express hay còn được viết là Express Js, Express.js. Đây là một framework mã nguồn mở miễn phí cho Node.js. Express được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.

Vì Express chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển. Express cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

Nhờ có Express mà các nhà lập trình có thể dễ dàng tạo các ứng dụng một website, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Express giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn. Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

Express cũng là một phần của công nghệ giúp quản lý các ứng dụng web một cách dễ dàng hơn hay còn được gọi là ngăn xếp phần mềm MEAN. Nhờ có thư viện Javascript của Express đã giúp cho các nhà lập trình xây dựng nền các ứng dụng web hiệu quả và nhanh chóng hơn. Express cũng được sử dụng để nâng cao các chức năng của Node.js.

Trên thực tế, nếu không sử dụng Express, bạn sẽ phải thực hiện rất nhiều bước lập trình phức tạp để xây dựng nền một API hiệu quả. Express đã giúp cho việc lập trình trong Node.js trở nên dễ dàng hơn và có nhiều tính năng mới bổ sung.

Vai trò của Express

Phát triển máy chủ nhanh chóng: Express cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian.

Phần mềm trung gian Middleware: Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.

Định tuyến - Routing: Express cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL.

Tạo mẫu - Templating: Các công cụ tạo khuôn mẫu được Express cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ.

Gỡ lỗi - Debugging: Để phát triển thành công các ứng dụng web không thể thiết đi việc gỡ lỗi. Giờ đây với Express việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi.

PostgreSQL

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.

PostgreSQL được phát triển dựa trên POSTGRES 4.2 tại phòng khoa học máy tính Berkeley, Đại học California.

PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.

PostgreSQL là một phần mềm mã nguồn mở miễn phí. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do. Theo đó, bạn sẽ được tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức.

PostgreSQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên PostgreSQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

JSON Web Token (JWT)

JWT (JSON Web Tokens) là một chuỗi mã hoá hay một tiêu chuẩn mở RFC 7519 được sử dụng như một phương tiện đại diện nhỏ gọn có khả năng thông qua chuỗi JSON để truyền đạt thông tin giữa server và client.

JSON là một dạng dữ liệu được sử dụng theo một quy luật chung nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể dễ dàng đọc và tìm hiểu. Nó được xem là một tiêu chuẩn mở với mục đích sử dụng là trao đổi dữ liệu trong website.

Token là một chữ ký điện tử được mã hoá thành các con số khác nhau và các con số này tạo thành một dãy số ẩn tượng. Do được tạo dưới dạng OTP nên loại mã số này chỉ được tạo ngẫu nhiên và sử dụng trong một lần cho từng lần giao dịch khác nhau.

Các thông tin trong chuỗi JWT được định dạng bằng JSON đồng thời được đảm bảo tính bảo mật và gia tăng mức độ tin tưởng dựa trên chữ ký điện tử. Chữ ký điện tử này sẽ sử dụng các Public Key (khóa công khai) và Private Key (khóa riêng tư) có ứng dụng ECDSA và RSA với thuật toán HMAC.

Lợi ích của việc sử dụng JWT:

Ủy quyền - Authorization: Đây là trường hợp nên sử dụng JWT. Khi người dùng đã đăng nhập, mỗi request tiếp theo được gửi từ Client sẽ bao gồm JWT, cho phép người dùng access vào routes, services, and resources được phép với token đó. Single Sign ON là tính năng sử dụng JWT rộng rãi hiện nay, vì chi phí thấp và dễ dàng sử dụng trên các domains khác nhau.

Trao đổi thông tin - Information Exchange: JSON Web Tokens là một cách tốt để truyền thông tin an toàn giữa các bên Client và Server. Vì JWT có thể signed. Ví dụ, sử dụng các cặp public/private key, bạn có thể biết chắc người gửi. Ngoài ra, vì signature được xác định dựa vào header và payload, bạn cũng có thể xác minh rằng nội dung chưa bị giả mạo.

Reactjs

ReactJS được hiểu nôm na là một thư viện mã nguồn mở chứa nhiều JavaScript và người tạo ra ReactJS chính là ông trùm với cái tên quen thuộc Facebook, được ra đời vào năm 2013. Mục đích của việc tạo ReactJS là tạo ra các ứng dụng web nhanh, hiệu quả và hấp dẫn với nỗ lực viết mã tối thiểu. Mục tiêu chính của ReactJS là bất kỳ trang web nào sử dụng ReactJS phải mượt mà, nhanh chóng, khả năng mở rộng cao và dễ thực hiện.

Nhìn chung, các tính năng và điểm mạnh của ReactJS thường đến từ việc tập trung vào các phần riêng lẻ. Do đó, khi làm việc với web thay vì toàn bộ ứng dụng của một trang web bằng ReactJS, các nhà phát triển có thể tách rời và chuyển đổi giao diện người dùng từ những cách phức tạp và biến nó thành những phần đơn giản hơn. Điều này có nghĩa là kết xuất dữ liệu không chỉ ở phía máy chủ. Thực hiện với các vị trí, nhưng bạn cũng có thể thực hiện tại vị trí khách hàng khi sử dụng ReactJS.

Redux

Nói một cách đơn giản, Redux là một công cụ quản lý trạng thái. Mặc dù nó được sử dụng chủ yếu với React, nhưng nó có thể được sử dụng với bất kỳ khung hoặc thư viện JavaScript nào khác. Nó rất nhẹ ở mức 2KB (bao gồm cả phụ thuộc), vì vậy bạn không phải lo lắng về việc nó làm cho ứng dụng của bạn có kích thước lớn.

Với Redux, trạng thái ứng dụng của bạn được giữ trong một "store" và mỗi thành phần có thể truy cập bất kỳ trạng thái nào mà nó cần từ "store" này. Sâu hơn một chút để xem tại sao bạn có thể cần một công cụ quản lý trạng thái.

Redux - Saga

Redux saga là một thư viện giúp các side effect của ứng dụng tức là những thứ không đồng bộ như tìm nạp dữ liệu và những thứ không tinh khiết như truy cập bộ nhớ cache của trình duyệt dễ quản lý hơn, thực thi hiệu quả hơn, dễ kiểm tra và xử lý lỗi tốt hơn.

Redux-saga là một phần mềm trung gian của redux (Redux là một thư viện JavaScript mã nguồn mở để quản lý trạng thái ứng dụng. Nó được sử dụng phổ biến nhất với các thư viện như React hoặc Angular để xây dựng giao diện người dùng). Phần mềm Redux này có thể được bắt đầu, tạm dừng và hủy khỏi ứng dụng chính với các hành động redux bình thường.

Redux saga có quyền truy cập vào trạng thái ứng dụng redux đầy đủ và gửi các hành động Redux.

Redux saga sử dụng một tính năng ES6 được gọi là Generators khiến các luồng không đồng bộ đó dễ đọc, ghi và kiểm tra. Bằng cách sử dụng Redux saga, các luồng không đồng bộ này trông giống như mã JavaScript đồng bộ tiêu chuẩn của bạn.

Ant Design

Trong quá trình phát triển các ứng dụng trên máy tính, có nhiều thông số thiết kế và hệ thống xử lý khác nhau rất phức tạp, điều này gây khó khăn gấp đôi cho các nhà thiết kế và lập trình viên, đồng thời sẽ làm giảm hiệu quả cho việc phát triển này. Sau khi có cho mình các dự án lớn và sơ lược lại, đội ngũ Ant UED đã cho ra mắt ngôn ngữ thiết kế Ant Design dành cho giao diện của ứng dụng, nhằm mục đích nhất quán các thông số giao diện người dùng trên nền tảng dự án nội bộ, giảm chi phí không cần thiết của việc triển khai ý tưởng và cung cấp nhiều tài nguyên cho việc thiết kế và lập trình front-end.

Ant Design đặc biệt được tạo ra cho các ứng dụng máy tính nội bộ, cam kết trải nghiệm của người dùng và các nhà thiết kế sản phẩm. Những người thiết kế giao diện và thiết kế trải nghiệm cho người dùng được gọi chung là thiết kế sản phẩm, Ant Design sẽ xóa bỏ ranh giới giữa việc quản lý sản phẩm, thiết kế tương tác, thiết kế hình ảnh, phát triển giao diện và quản trị dữ liệu người dùng. Nhờ việc tận dụng sự đồng nhất về đặc điểm kỹ thuật, Ant Design đơn giản hóa việc thiết kế và lập trình cho những người đang thực hiện Project, điều này sẽ giúp họ có thêm nhiều kinh nghiệm và tăng hiệu quả phát triển giao diện của sản phẩm và ứng dụng.

Material UI

Material Design là một phiên bản phát triển hơn so với Flat Design (thiết kế phẳng) và được các designer vô cùng yêu thích trong thời gian gần đây. Không giống hoàn toàn với Flat Design khi mang đến cảm giác cả thiết kế là một mặt phẳng, Material Design thực chất là các lớp Flat xếp chồng lên nhau để tạo điểm nhấn và chiều sâu cho thiết kế của bạn.

Đối với phong cách thiết kế này, các designer thường tập trung chủ yếu vào các chi tiết như đường nét tối giản hay màu sắc được chọn là màu đậm để mang lại sự nổi bật cho tổng thể thiết kế. Ngoài ra các yếu tố đồ họa tạo cảm giác 3D cũng được sử dụng để tạo điểm nhấn cần thiết cho thiết kế, khiến người xem có cảm giác như hiệu ứng của bạn nổi lên trên giao diện.

Material Design tập trung vào đường nét tối giản và màu sắc đậm nét

Không chỉ gây ấn tượng với người xem bằng những yếu tố đồ họa 3D, Material Design còn sử dụng những chuyển động tự nhiên, ví dụ như các menu hay các nút hiện trên màn hình. Nhờ vậy mà người dùng có thêm những trải nghiệm mới mẻ và gần gũi hơn với đời sống thực.

Font Awesome

Font awesome là một thư viện để tạo ra những icon tiện lợi, nhẹ nhàng có tính tùy chỉnh cao trong website. Thay vì bạn phải mất công design icon bằng file ảnh nặng nề thì giờ đây có hàng ngàn icon miễn phí sử dụng nhanh chóng.