# 

**Initial Design Document**

**Finder Matchmaking Web Application**

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Version 1.0 approved**

**Authors**

Trần Thái Dương

Đỗ Đức Cường

Hoàng Hữu Phước

**Members**

Trần Thái Dương

Đỗ Đức Cường

Đỗ Đình Dũng

Nguyễn Thành Huy

Nguyễn Hữu Phước

**Table of content**

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 0](#_heading=)

[1. Introduction 1](#_heading=)

[1.1 Purpose 1](#_heading=)

[2. Architectural Design 1](#_heading=)

[2.1 Introduction 1](#_heading=)

[2.1 Purpose 1](#_heading=h.zbfkh1chw87v)

[2.2 Scope 1](#_heading=h.5jq1gkze780l)

[2.2 System Architecture 1](#_heading=)

[2.1 Overview Diagram 1](#_heading=h.rylkov687umn)

[2.2 Details Diagram 4](#_heading=h.31b9edsle554)

[2.3 Components Description 5](#_heading=h.gtudvtbaxiv8)

[2.3.1 Presentation 5](#_heading=h.2yc2rufgakeh)

[2.3.2 Admin/User Activities 5](#_heading=h.ylshni58rci0)

[2.3.3 Domain 5](#_heading=h.h4i7qz71jm7a)

[2.3.4 Processor 5](#_heading=h.x4jktzhp2ud8)

[2.3.5 External Interface 5](#_heading=h.ludnbyib7qiy)

[2.3.6 Data Access Object 5](#_heading=h.yxt8xhy4cqik)

[2.3.7 Database Management System 6](#_heading=h.fd41dctf44j6)

[**3. Diagrams 6**](#_heading=h.fpcq934z7kxp)

[3.1 Sequence Diagram 6](#_heading=h.ba0grxi1kylc)

[3.1.1 Host Event 6](#_heading=h.bkixwbmib4qm)

[3.1.2 Join Request 7](#_heading=h.5dztpo7k2l41)

[3.1.3 Login 7](#_heading=h.26byacb352cm)

[3.1.4 Manage Event 8](#_heading=h.hxgmd2qb2w6i)

[3.1.5 Register 8](#_heading=h.vye1gd46dtdu)

[3.1.6 Update User’s Info 9](#_heading=h.1h04d7rbr7z9)

[3.1.7 View Event Info 9](#_heading=h.zxal5m896qa)

[3.2 Class Diagram 10](#_heading=h.5yk9ghcni2u)

[4. Demo UI 10](#_heading=)

[**5. Database Schema 11**](#_heading=h.vw830i93dxs0)

**Revision History**

| **Name** | **Date** | **Reason for Changes** | **Version** |
| --- | --- | --- | --- |
| Assignment 2 | 5/5/2024 |  | 1.0 |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

This document outlines the preliminary design for an event scheduling web application. This application will provide a platform for users to create, manage, and attend events of various types. It aims to simplify the event planning process, streamline communication between organizers and attendees, and ultimately increase event participation and success.

# 2. Architectural Design

## 2.1 Introduction

### 2.1 Purpose

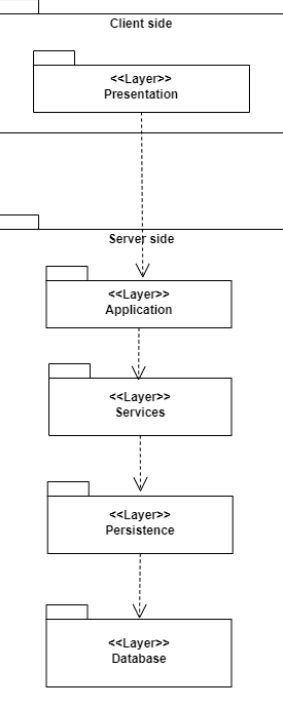
The system architecture document serves as a detailed map, outlining the structure, components, and operation of a software system. It not only creates unity within the development team by ensuring everyone has a clear understanding of the system but also supports the design, development, maintenance, and upgrade processes. Additionally, this document helps identify potential risks, ensures consistency, and serves as a crucial reference throughout the project's lifecycle. In essence, the system architecture document acts as a solid foundation, contributing to the overall success of a software project.

### 2.2 Scope

This document outlines the scope of Finder Application, a platform designed to facilitate event organization, participation, and social interaction. Three core functionalities are encompassed within the system. Firstly, the event organization functionality allows users to create and manage their own events, defining event details, quantities, and other options. They can track registrations, manage guest lists, and interact with attendees. Secondly, the event participation functionality enables users to search for and register to attend events organized by others. The system provides detailed information about events and allows participants to interact with both organizers and other attendees. Finally, the social interaction functionality fosters a sense of community by enabling users to comment on events, share events on social media,...

## 2.2 System Architecture

### 2.1 Overview Diagram



* **Presentation Layer (or User Interface Layer):**

This layer encompasses components such as the user interface, web browsers, mobile applications, or any other element that interacts with users. Its primary functions are to display information and results from the Business Layer to users and to transfer user input to the Business Layer for processing.

* **Application Layer:**

The Application layer acts as a bridge, connecting the user interface (Presentation layer) with the business logic (Business Services layer). It handles application logic, coordinates tasks, and prepares data for both sides. It also manages user sessions and ensures smooth application operation.

* **Services Layer:**

Responsible for handling business logic, this layer receives requests from the Presentation Layer, processes them, and sends corresponding requests to the Persistence Layer. It acts as the intermediary between the user interface and data storage.

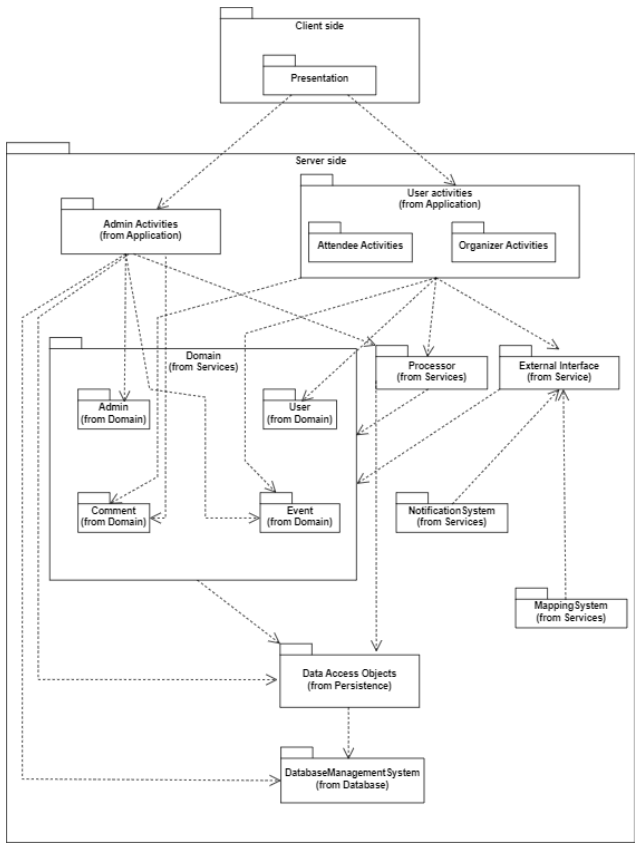
* **Persistence Layer:**

This layer is tasked with sending requests to the Database Layer to perform data-related operations. It acts as an abstraction layer, separating the business logic from the specifics of data access and storage.

* **Database Layer:**

Comprising the database and related components such as the database management system, this layer is responsible for managing the database, executing read and write operations on data, and implementing queries and data storage as defined by the Persistence Layer.

### 2.2 Details Diagram

****

### 2.3 Components Description

#### 2.3.1 **Presentation**

The Presentation encompasses components such as the user interface, web browser, mobile application, or any other element that interacts with users. Its primary functions are to display information and results from the Logic Layer to the user and to transfer user input data to the Logic Layer for processing.

#### 2.3.2 Admin/User Activities

User Activities and Admin Activities represent the actions taken by users and administrators, respectively, as they interact with the application through the Presentation Layer. These activities are analyzed to determine the type of request and then routed to the appropriate components within the Services Layer for processing and fulfillment.

#### 2.3.3 Domain

The Domain Layer houses and manages the core business objects of the system. In this specific system, it encompasses:

1. **Admin:** This component handles the management of administrator information and actions within the system.
2. **Event:** This component manages all event-related information, including details, schedules, and other relevant data.
3. **User:** This component oversees user activities and information within the system.
4. **Comment:** This component manages user comments associated with events, allowing for interaction and feedback.

#### 2.3.4 Processor

The Processor utilizes information from the Domain layer and requests from the Application layer to execute business logic calculations. Based on these calculations, it generates corresponding requests to the Persistence layer for data access and manipulation.

#### 2.3.5 External Interface

The External Interface is responsible for facilitating communication with external systems. In this context, it specifically interacts with map systems and notification systems such as email and SMS services. This allows the application to leverage external functionalities and integrate with other platforms.

#### 2.3.6 Data Access Object

The Data Access Object (DAO) receives requests from the Services layer and provides methods to perform CRUD (Create, Read, Update, Delete) operations on the corresponding entities. It encapsulates the details of database access (e.g., SQL queries) from the Services layer, promoting modularity and ease of maintenance.

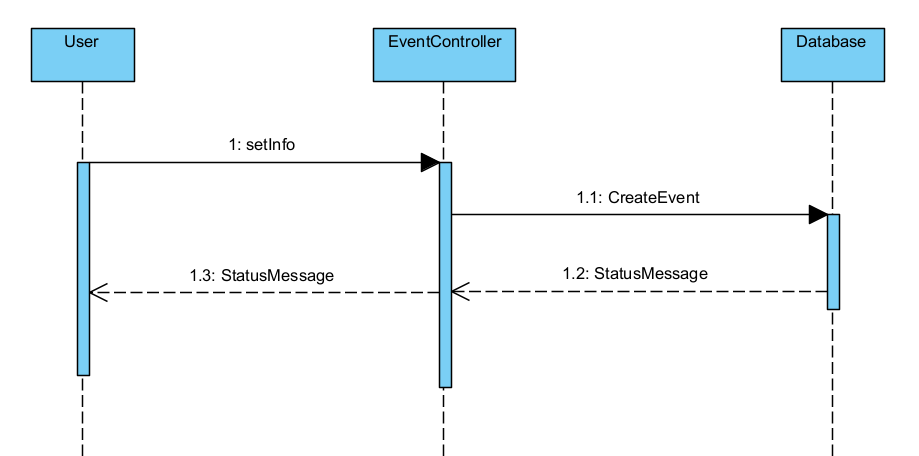
#### 2.3.7 Database Management System

Directly interacts with the database, receiving queries from the Persistence layer and returning corresponding data. It also executes operations such as adding, deleting, and updating data within the database.

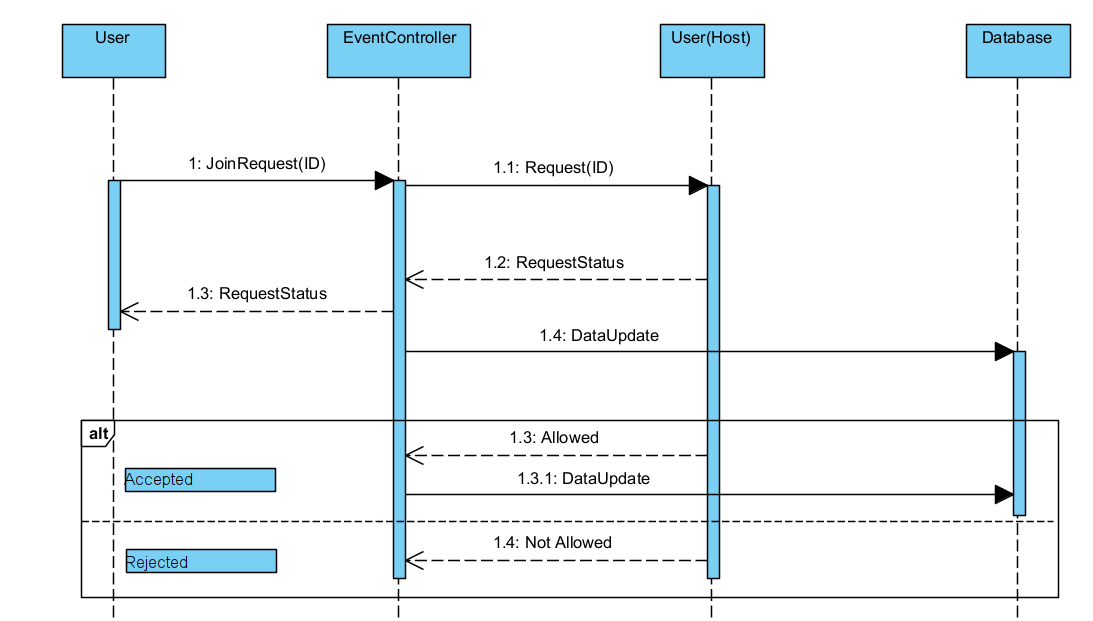
# 3. Diagrams

## 3.1 Sequence Diagram

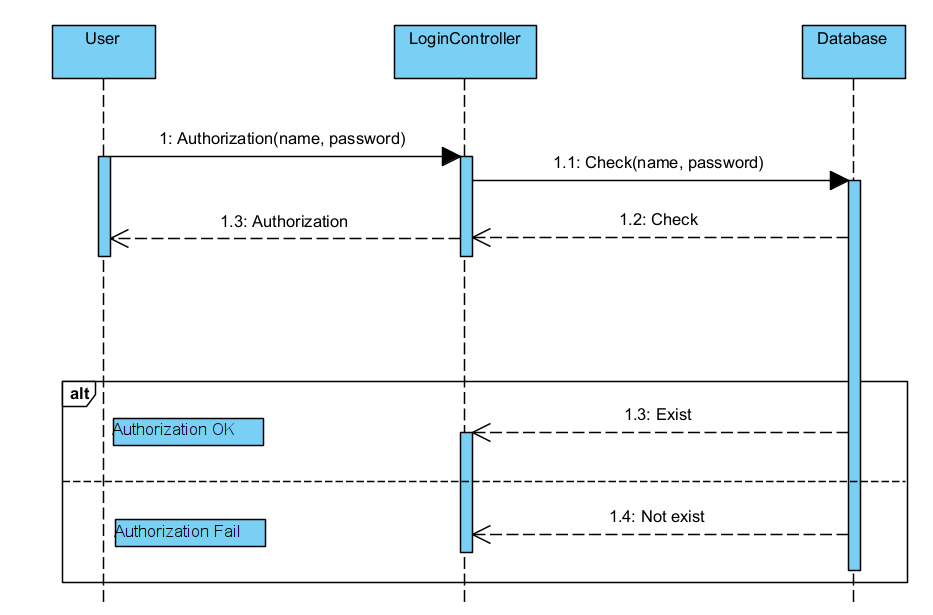
### 3.1.1 Host Event



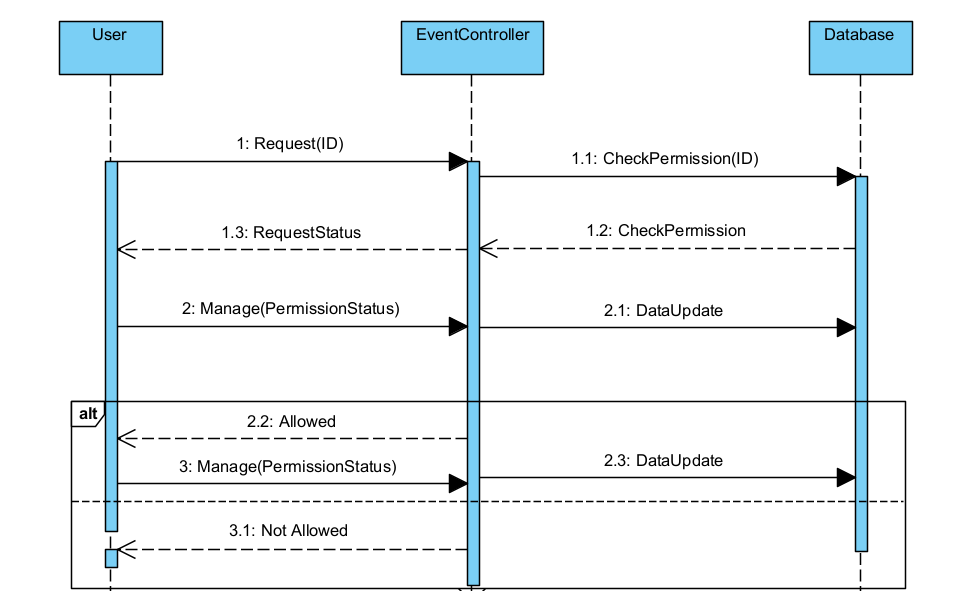
### 3.1.2 Join Request



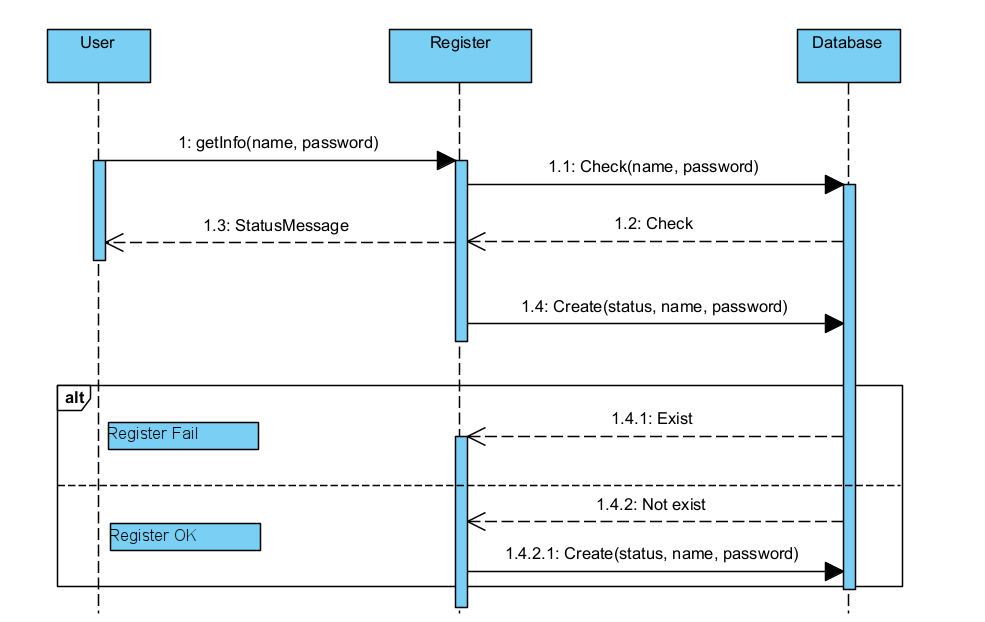
### 3.1.3 Login



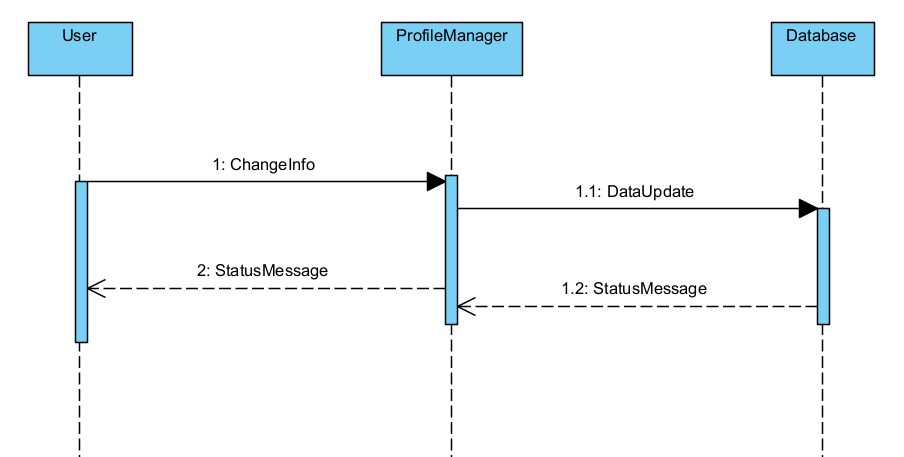
### 3.1.4 Manage Event



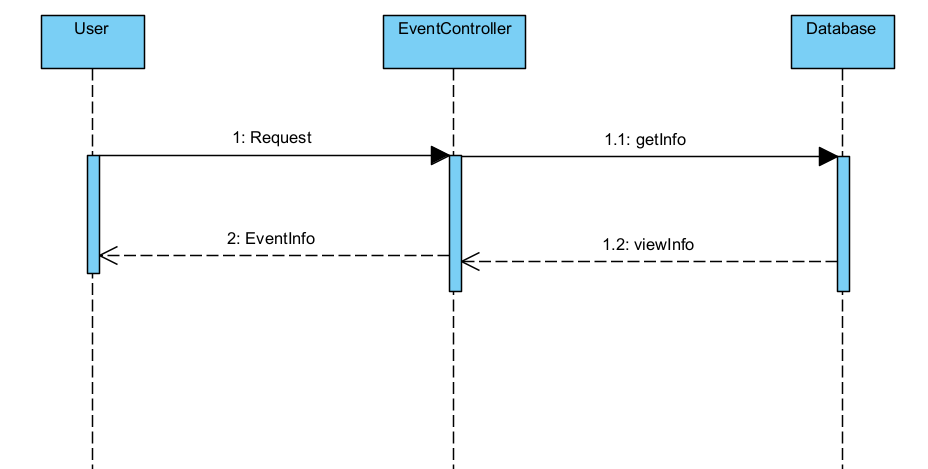
### 3.1.5 Register



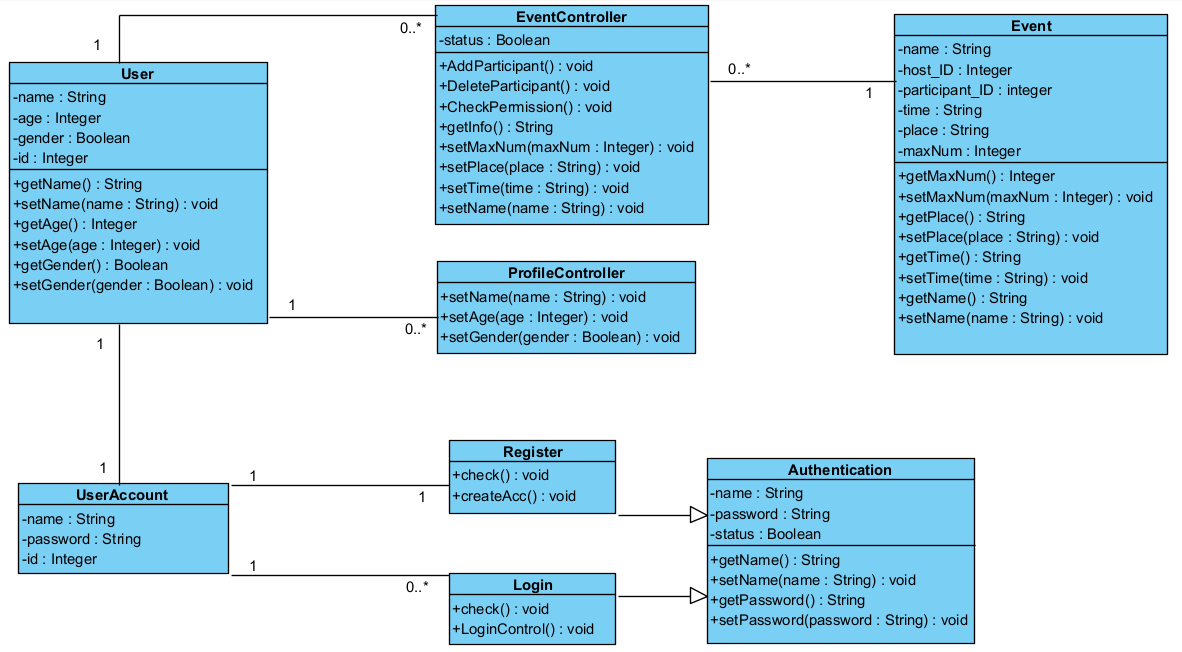
### 3.1.6 Update User’s Info



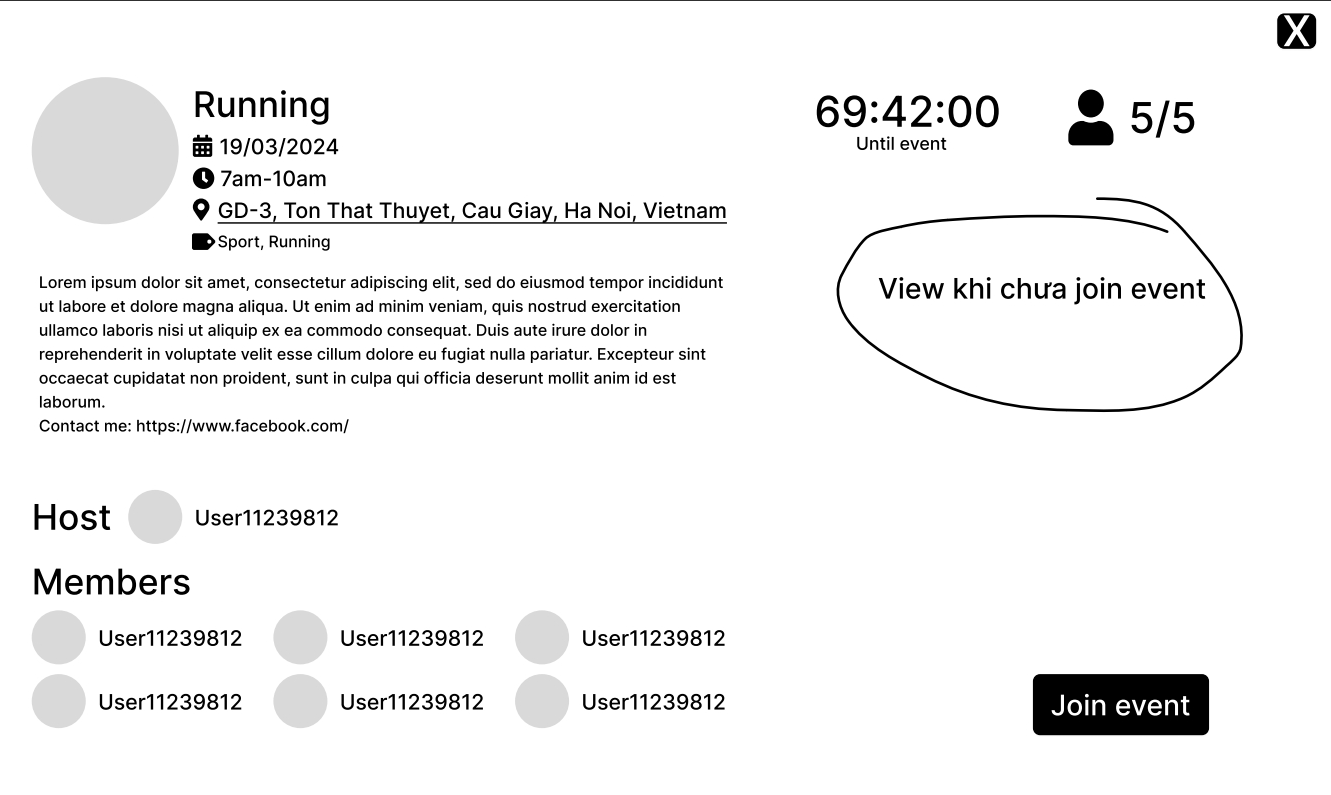
### 3.1.7 View Event Info

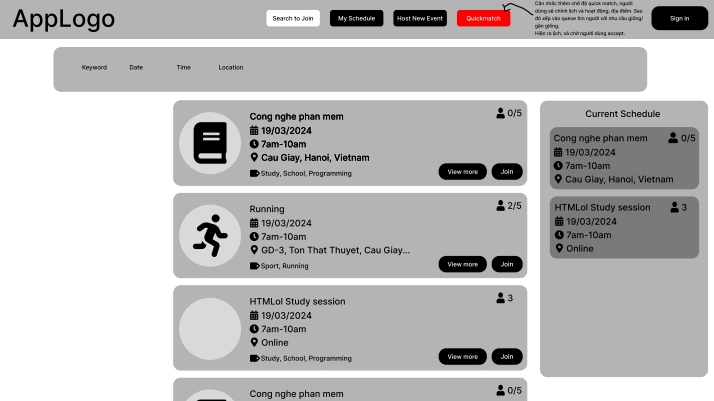


## 3.2 Class Diagram

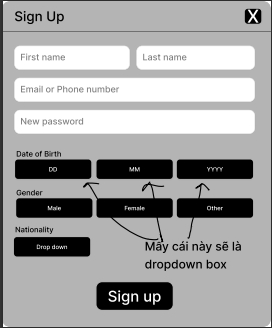


# 4. Demo UI









# 5. Database Schema

