UTM UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

## Semester II 2024/2025

**Subject : Database (SECD2523)**

**Section : 06**

**Task : PHASE 3**

**Lecturer : Dr LAYLA HASAN**

**Group : 05**

| | **Name** | **Matric Number** |
|---|---|---|
| 1 | RAMI YASSEIN ELTAYEB MOHAMED | A23CS0022 |
| 2 | AHMED MAHGOUB SAIDAHMED MAHGOUB | A23CS3023 |
| 3 | MOHAMED ADEL ABDALLA MAHMOUD | A23CS4061 |
| 4 | AHMED ALI ABDALLA MUSA | A23CS4036 |
| 5 | MUSAB ELSADIG ELMAKI AHMED | A23CS4049 |

## *Table Of Content*

# 1. Introduction

Currently in today’s age, it is quite useful and helpful to adopt technology and let it simplify lives. In recent years, the world has undergone a transformation as it has now become an important aspect of almost everything. Our university is one of those places where it is also a key area of concentration.

Many of the students on the UTM campus suffer from either lack of access or sometimes even the complete absence of updated relevant information about the various clubs and societies. So students tend to miss events, opportunities and stay out of the loop a lot because of the current forms of communication like campus group mass messages being cluttered. In addition, a lot of clubs are not very well known, which Increasingly prevents the students from participating in different activities.

Our team has embarked on a project to tackle these challenges through the development of a mobile application that will allow the Students to access Clubs Platforms and Explore various clubs Activities , Get in Touch with Clubs Announcements, and Registers for Events . The result will be a new dimension as it will provide students with options of engaging into many activities outside classes .

## 2. Overview of the project

During the database logical design phase, the conceptual Entity Relationship Diagram developed earlier is transformed into logical ERD . Adjustments are made to accommodate the complexities involved in managing Events by creating a more structured and efficient database. Each entity in the logical ERD is mapped to a relational schema, with attributes defined and primary keys established for every table. This process is critical for defining and relating entities such as Student, Event, Club, Organizer, payment, And administrator within the ClubHub System.

Normalization is an important focus at this stage, as it eliminates data redundancies and ensures proper dependencies within the relational schema, updates to the data dictionary are made to reflect the changes brought about by normalization.

Additionally, SQL DDL and DML statements are proposed to validate the logical ERDs against the system's transactional requirements. Lastly User interface design is developed to ensure the database can efficiently handle main operations such as managing, scheduling, Registering, and approving the events and payment processing.

In the context of ClubHub system, these validations are essential, as they directly impact the system's responsiveness, reliability, and its ability to enhance Student’s campus experiences.

# 3. Database conceptual design

## 3.1 Updated business rule

**Regular User :**

- View a list of all approved events.
- Can filter the event based on preferences like category , cost or date.
- Can view the event details such as the title, requirement of joining the activity, cost, description of the activity, total capacity.
- Receive a recommendation list based on their preferences by the system .
- Register for the event by providing event details .
- Do an online payment for an event that they registered for .
- View details by their profiles of their registered events , Notifications about event, attendance record , Event certificates .

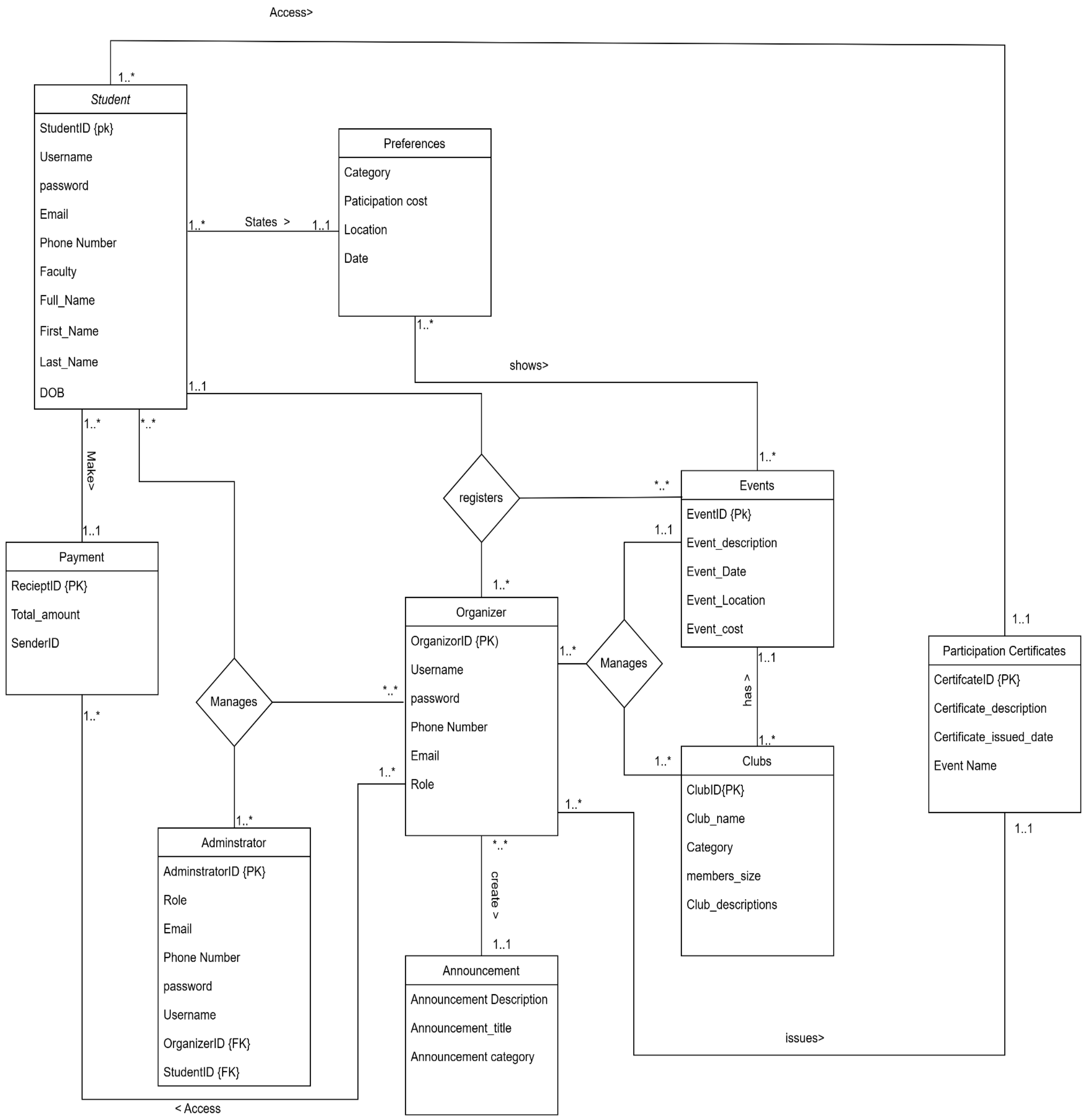
**Organizer :**

- Create new events with details like title, description, requirement, cost .
- View The Validation Status of an Event.
- Provide Instruction of User Credentials to Login .
- Track the attendance and communicate with the participants .
- Review the student's Registration payment .

**Administrator :**

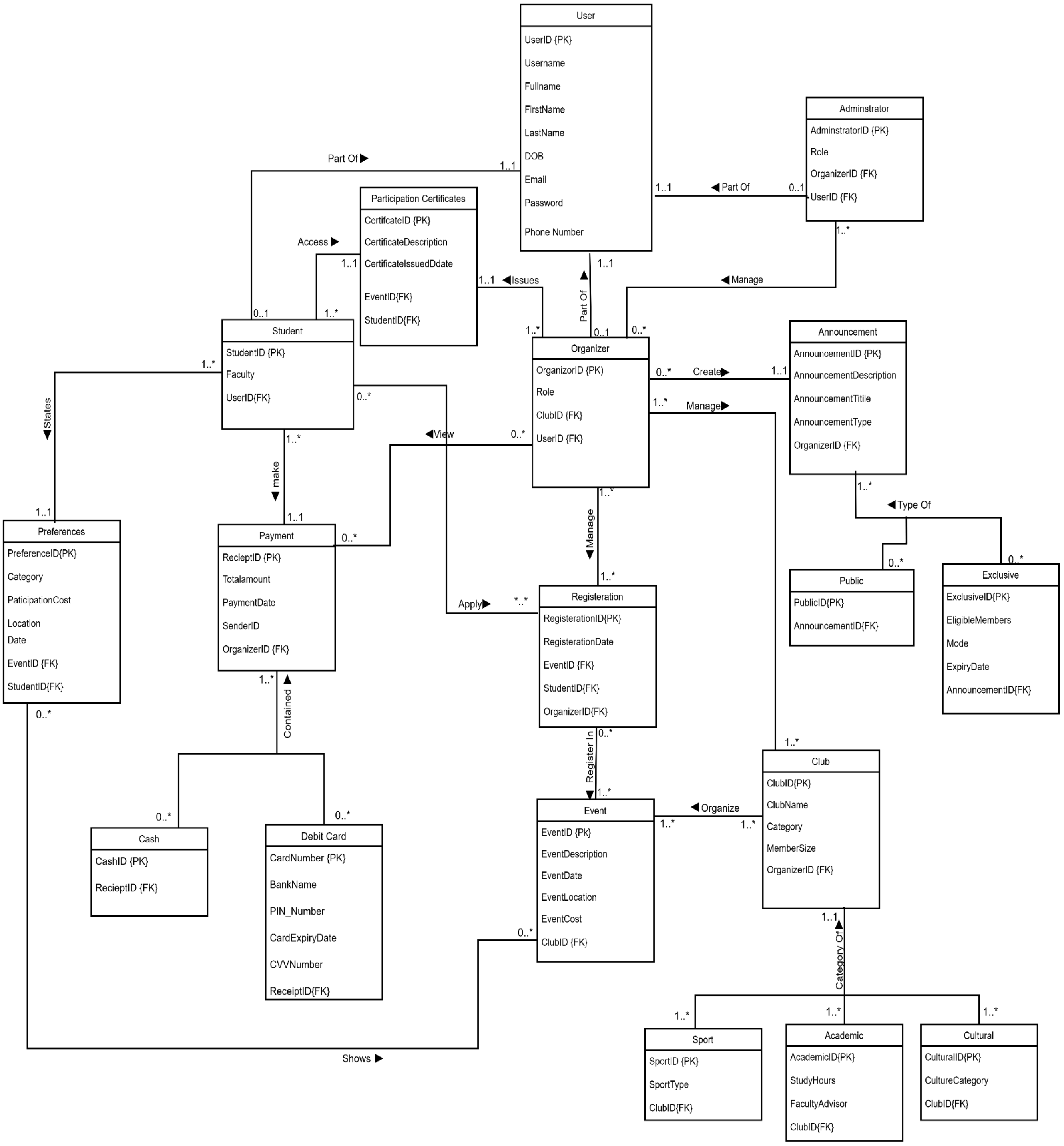
- Can approve or reject events submitted by the Organizer .
- Update the status of an event .
- Manage User credentials to Log in .

## 3.2 Conceptual ERD



# 4.0 DB logical design

## 4.1 Logical ERD



## 4.2 Updated Data Dictionary

### 4.2.1 Description of Entity

| Entity | Description | Occurrence |
|---|---|---|
| User | User’s Information | User entity contains Administrator , Student , and Organizer. |
| Student | Student’s Information | The Student states preferences and then he can access his participation certificates after his/her Apply for Registration is Accepted and after he/she Makes payment for Registration , A Student is a part of User. |
| Event | Events Information | Events are Organized by Clubs and the students Registers in Events that reflect their Preferences . |
| Club | Clubs Information | Clubs organize Events and the Clubs are managed by the Organizer. |
| Announcement | Announcement Information | Announcement is created by the Organizer . |
| preferences | Preferences Information | Preferences are stated by Student and it shows the Events . |
| Organizer | Organizer’s Information | The Organizer manages Clubs and he is managed by the Administrator , the Organizer Manages registration. |
| payment | Payment’s Information | Payment is made by the Student and it can be accessed by the Organizator. |
| Administrator | Administrator's Information | Administrator manages both the Organizer. |
| Participation Certificates | Participation Certificates Information | Participation Certificates are issued by the Organizer and can be accessed by the Student . |
| Cash | Hold payment information when paying by cash. | One of the selections available to Students when making payment. |

| | | |
|---|---|---|
| Debit Card | Hold payment information when paying by online transfer | One of the selections available to Students when making payment. |
| Registration | Hold Registeration's information based on Preferences. | Registration will be accomplished when Student Applies for his/her Preferences and it's managed by the Organizer and it will result in Registering Student in the Event. |
| Public | Hold Public Announcements information . | One of the selections available to Organizer's when creating an announcement. |
| Exclusive | Hold Exclusive Announcements information. | One of the selections available to Organizer's when creating an announcement. |
| Sport | Hold Sport's Club information . | One of the selections available to Student's when considering a Club. |
| Cultural | Hold Cultural Club information . | One of the selections available to Student's when considering a Club. |
| Academic | Hold Academic's Club information . | One of the selections available to Student's when considering a Club. |

### 4.2.2 Description of Relationship

| Entity | Multiplicity | Relationship | Multiplicity | Entity |
|---|---|---|---|---|
| Student | 1..* | Access | 1..1 | Participation Certificates |
|  | 1..* | States | 1..1 | Preferences |
|  | 1..* | Make | 1..1 | Payment |
|  | 0..* | Apply | *..* | Registration |
|  | 0..1 | Part Of | 1..1 | User |
| Organizer | 1..* | Manages | 1..* | Club |
|  | 0..* | View | 0..* | Payment |
|  | 1..* | Issues | 1..1 | Participation Certificates |
|  | 0..* | Create | 1..1 | Announcement |
|  | 1..* | Manage | 1..* | Registration |
|  | 0..1 | Part Of | 1..1 | User |
| Preferences | 0..* | Shows | 0..* | Event |
| Club | 1..* | Organize | 1..* | Event |
| Administrator | 1..* | Manages | 0..* | Organizer |
|  | 0..1 | Part Of | 1..1 | User |
| Registration | 0..* | Register In | 1..* | Event |
| Cash | 0..* | Contained | 1..* | Payment |
| Debit Card | 0..* | Contained | 1..* | Payment |
| Public | 0..* | Type Of | 1..* | Announcement |
| Exclusive | 0..* | Type Of | 1..* | Announcement |
| Sport | 1..* | Category Of | 1..1 | Club |
| Cultural | 1..* | Category Of | 1..1 | Club |
| Academic | 1..* | Category Of | 1..1 | Club |

### 4.2.3 Description of Attributes

| **Entity** | **Attributes** | **Description** | **Data Type** | **Null** | **Multi-Valued** |
|---|---|---|---|---|---|
| User | UserID | Uniquely identify the User . | VARCHAR(10) | NO | NO |
| | Username | User name of the User. | VARCHAR(20) | NO | NO |
| | FullName | Full Name of the User . | VARCHAR(30) | NO | NO |
| | FirstName | First Name of the User . | VARCHAR (15) | NO | NO |
| | LastName | Last Name of the User . | VARCHAR (15) | NO | NO |
| | DOB | the date of birth of user | DATE | NO | NO |
| | Password | User’s password. | VARCHAR(100) | NO | NO |
| | Email | The Email Address of the User . | VARCHAR(320) | YES | NO |
| | Phone Number | the phone number of user | VARCHAR (13) | NO | YES |
| Student | StudentID {PK} | Uniquely identify a Student . | VARCHAR(10) | NO | NO |
| | Faculty | Name of Student’s Faculty . | VARCHAR(50) | NO | NO |
| | UserID {FK} | Foreign Key of User which Uniquely identify the User. | VARCHAR(10) | NO | NO |
| Preferences | PreferenceID {PK} | Uniquely Identify the Preference. | VARCHAR(10) | NO | NO |
| | Category | The Category of Student’s Preferences . | VARCHAR(10) | YES | YES |
| | ParticipationCost | The cost of participation . | FLOAT(10) | YES | NO |
| | Location | The location of Student’s preferences . | VARCHAR (50) | YES | YES |
| | Date | The date of Student’s preferences . | DATE | YES | YES |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  | StudentID {FK} | Foreign Key of Student which Uniquely identify the Student. | VARCHAR(10) | NO | NO |
|  | EventID{FK} | Foreign Key of Event which Uniquely identify the Event. | VARCHAR(10) | NO | NO |
| Payment | ReceiptID {PK} | Uniquely identify a Student . | VARCHAR(10) | NO | NO |
|  | TotalAmount | The amount of the Payment . | FLOAT (5) | NO | NO |
|  | PaymentDate | The Date of the Payment. | DATE | NO | NO |
|  | SenderID | Uniquely identify the sender . | VARCHAR(10) | NO | NO |
|  | OrganizerID {FK} | Foreign Key of Organizer which Uniquely identify the Organizer. | VARCHAR(10) | NO | NO |
| Organizer | OrganizerID {PK} | Uniquely identify the Organizer . | VARCHAR(10) | NO | NO |
|  | Role | The Role of the Organizer . | VARCHAR(20) | NO | NO |
|  | UserID {FK} | Foreign Key of User which Uniquely identify the User. | VARCHAR(10) | NO | NO |
|  | ClubID {FK} | Foreign Key of Club which Uniquely identify the Club. | VARCHAR(10) | NO | NO |
| Event | EventID{PK} | Uniquely identify the Event . | VARCHAR(10) | NO | NO |
|  | EventDescription | The description of the Event . | VARCHAR(500) | NO | NO |
|  | EventDate | The Date of the Event . | DATE | NO | NO |
|  | EventLocation | The location of the Event . | VARCHAR(30) | NO | NO |
|  | EventCost | The Cost of the Event . | FLOAT (5) | NO | NO |

| | | | | | |
|---|---|---|---|---|---|
| | ClubID {FK} | Foreign Key of Club which Uniquely identify the Club. | VARCHAR(10) | NO | NO |
| Club | ClubID{PK} | Uniquely identify the Club . | VARCHAR(10) | NO | NO |
| | ClubName | The Name of the Club . | VARCHAR(20) | NO | NO |
| | Category | The Category of the Club . | VARCHAR(10) | NO | YES |
| | MemberSize | The members size of the Club . | INT | NO | NO |
| | OrganizerID {FK} | Foreign Key of Organizer which Uniquely identify the Organizer. | VARCHAR(10) | NO | NO |
| Announcement | AnnouncementID {PK} | Uniquely identify the Announcement . | VARCHAR(10) | NO | NO |
| | Announcement Description | The Description of the Announcement . | VARCHAR(1000) | NO | NO |
| | Announcement title | The Title of the Announcement . | VARCHAR(50) | NO | NO |
| | Announcement Type | the Type of the Announcement . | VARCHAR(10) | NO | NO |
| | OrganizerID {FK} | Foreign Key of Organizer which Uniquely identify the Organizer. | VARCHAR(10) | NO | NO |
| Administrator | AdministratorID {PK} | Uniquely identify the Administrator. | VARCHAR(10) | NO | NO |
| | Role | The Role of the Administrator . | VARCHAR(20) | NO | YES |
| | OrganizerID {FK} | Foreign Key of Organizer which Uniquely identifies the Organizer. | VARCHAR(10) | NO | NO |
| Participation Certificates | CertificateID {PK} | Uniquely identify the Certificate. | VARCHAR(10) | NO | NO |
| | Certificate | The Description of the | VARCHAR(1000) | NO | NO |

| | | | | | |
|---|---|---|---|---|---|
| | Description | Certificate. | | | |
| | CertificateIssued Date | The date of issuing the Certificate. | DATE | NO | NO |
| | EventID{FK} | Foreign Key of Event which Uniquely identify the Event. | VARCHAR(10) | NO | NO |
| | StudentID {FK} | Foreign Key of Student which Uniquely identify the Student. | VARCHAR(10) | NO | NO |
| Registration | Registration ID{PK} | Uniquely Identify the Registration. | VARCHAR(10) | NO | NO |
| | RegistrationDate | The Date of the Registration . | DATE | NO | NO |
| | OrganizerID {FK} | Foreign Key of Organizer which Uniquely identify the Organizer. | VARCHAR(10) | NO | NO |
| | StudentID {FK} | Foreign Key of Student which Uniquely identify the Student. | VARCHAR(10) | NO | NO |
| | EventID{FK} | Foreign Key of Event which Uniquely identify the Event. | VARCHAR(10) | NO | NO |
| Cash | CashID{PK} | Uniquely Identify the Cash payment . | VARCHAR(10) | NO | NO |
| | ReceiptID {FK} | Foreign Key of Payment which Uniquely identify the Payment. | VARCHAR(10) | NO | NO |
| DebitCard | CardNumber {PK} | The Number of The Card. | VARCHAR (16) | NO | NO |
| | BankName | The Name of The Bank | VARCHAR(10) | NO | NO |
| | PIN-Number | The Card PIN’s Number. | FLOAT(6) | NO | NO |
| | CardExpiryDate | The Expiry Date of the Card . | DATE | NO | NO |
| | CVVNumber | The CVV Number of the Card. | VARCHAR(4) | NO | NO |

| | | | | | |
|---|---|---|---|---|---|
| | ReceiptID {FK} | Foreign Key of Payment which Uniquely identify the Payment. | VARCHAR(10) | NO | NO |
| Public | PublicID{PK} | Uniquely Identify the Public Announcement. | VARCHAR(10) | NO | NO |
| | AnnouncementID {FK} | Foreign Key of Announcement Uniquely Identify the announcement. | VARCHAR(10) | NO | NO |
| Exclusive | ExclusiveID {PK} | Uniquely Identify the Exclusive Announcement. | VARCHAR(10) | NO | NO |
| | EligibleMembers | The Eligible Members for the Announcement. | VARCHAR(500) | NO | YES |
| | Mode | The Mode Of the Announcement. | VARCHAR(15) | NO | NO |
| | ExpiryDate | The Expiry Date of The Announcement. | DATE | YES | NO |
| | AnnouncementID {FK} | Foreign Key of Announcement Uniquely Identify the announcement. | VARCHAR(10) | NO | NO |
| Sport | SportID{PK} | Uniquely Identify the Club. | VARCHAR(10) | NO | NO |
| | SportType | The Type Of the Sport. | VARCHAR(30) | NO | NO |
| | ClubID {FK} | Foreign Key of Club which Uniquely identify the Club. | VARCHAR(10) | NO | NO |
| Academic | AcademicID {PK} | Uniquely Identify the Academic Club. | VARCHAR(10) | NO | NO |
| | StudyHour | The Hours of the Study in the Club. | FLOAT(2) | YES | NO |
| | FacultyAdvisor | The Faculty Advisor For the Club. | VARCHAR(20) | YES | YES |
| | ClubID {FK} | Foreign Key of Club which Uniquely identify the Club. | VARCHAR(10) | NO | NO |

| | | | | | |
|---|---|---|---|---|---|
| Cultural | CulturalID{PK} | Uniquely Identify the Cultural Club. | VARCHAR(10) | NO | NO |
| | CultureCategory | The Category of Culture. | VARCHAR(30) | NO | NO |
| | ClubID {FK} | Foreign Key of Club which Uniquely identify the Club. | VARCHAR(10) | NO | NO |

## 4.3 Normalization

1. User (UserID, Username, FullName, FirstName, LastName, DOB, Password, Email, PhoneNumber)
   FD1: UserID → Username, FullName, FirstName, LastName, DOB, Password, Email, PhoneNumber
   **1NF&2NF&3NF&BNCF:**
   User (UserID, Username, FirstName, LastName, DOB, Password, Email, PhoneNumber)

2. Student (StudentID,Faculty, UserID)
   FD1: StudentID → StudentID, Faculty, UserID
   **1NF&2NF&3NF&BNCF:**
   Student (StudentID, Faculty, UserID)

3. Preferences (PreferenceID ,Category, ParticipationCost, Location, Date, StudentID, EventID)
   FD1: PreferenceID → Category, ParticipationCost, Location, Date, StudentID, EventID
   **1NF&2NF&3NF&BNCF:**
   Preferences (PreferenceID, Category, ParticipationCost, Location, Date, StudentID, EventID)

4. Payment (ReceiptID, TotalAmount, PaymentDate, SenderID, OrganizerID)
   FD1: ReceiptID → TotalAmount, PaymentDate, SenderID, OrganizerID
   **1NF&2NF&3NF&BNCF:**
   Payment (ReceiptID, TotalAmount, PaymentDate, SenderID, OrganizerID)

5. Organizer (OrgnizerID, PhoneNumber, Role, UserID, ClubID)
   FD1: OrgnizerID → PhoneNumber, Role, UserID, ClubID
   **1NF&2NF&3NF&BNCF:**
   Organizer (OrgnizerID, PhoneNumber, Role, UserID, ClubID)

6. Event (EventID, EventDescription, EventDate, EventLocation, EventCost, ClubID)
   FD1: EventID → EventDescription, EventDate, EventLocation, EventCost, ClubID
   **1NF&2NF&3NF&BNCF:**
   Event (EventID, EventDescription, EventDate, EventLocation, EventCost, ClubID)

7. Club (ClubID, ClubName, Category, MemberSize,OrganizerID)
   FD1: ClubID → ClubName, Category, MemberSize, OrganizerID
   **1NF&2NF&3NF&BNCF:**
   Club (ClubID, ClubName, Category, MemberSize, OrganizerID)

8. Announcement (AnnouncementID, AnnouncementDescription, Announcement title, AnnouncementType, OrganizerID)  
   FD1: AnnouncementID → AnnouncementDescription, Announcement title, AnnouncementType, OrganizerID  
   **1NF&2NF&3NF&BNCF:**  
   Announcement (AnnouncementID, AnnouncementDescription, Announcement title, AnnouncementType, OrganizerID)

9. Administrator (AdministratorID, Role, OrganizerID)  
   FD1: AdministratorID → Role, OrganizerID  
   **1NF&2NF&3NF&BNCF:**  
   Administrator (AdministratorID, Role, OrganizerID)

10. Participation Certificates (CertificateID, CertificateDescription, CertificateIssuedDate, EventID, StudentID)  
    FD1: CertificateID → CertificateDescription, CertificateIssuedDate, EventID, StudentID  
    **1NF&2NF&3NF&BNCF:**  
    Participation Certificates (CertificateID, CertificateDescription, CertificateIssuedDate, EventName, EventID, StudentID)

11. Registration (RegistrationID, RegistrationDate, OrganizerID, StudentID, EventID)  
    FD1: RegistrationID → RegistrationDate, OrganizerID, StudentID, EventID  
    **1NF&2NF&3NF&BNCF:**  
    Registration (RegistrationID, RegistrationDate, OrganizerID, StudentID, EventID)

12. Cash (CashID, ReceiptID)  
    FD1: CashID → ReceiptID  
    **1NF&2NF&3NF&BNCF:**  
    Cash (CashID, ReceiptID)

13. Debit Card (CardNumber, BankName, PIN-Number, CardExpiryDate, CVVNumber, ReceiptID)  
    FD1: CardNumber → BankName, PIN-Number, CardExpiryDate, CVVNumber, ReceiptID)  
    **1NF&2NF&3NF&BNCF:**  
    Debit Card (CardNumber, BankName, PIN-Number, CardExpiryDate, CVVNumber, ReceiptID)

14. Public (PublicID, AnnouncementID)  
    FD1: PublicID → AnnouncementID  
    **1NF&2NF&3NF&BNCF:**  
    Public (PublicID, AnnouncementID)

15. Exclusive (ExclusiveID, EligibleMembers, Mode, ExpiryDate, AnnouncementID)
    FD1: ExclusiveID → EligibleMembers, Mode, ExpiryDate, AnnouncementID
    **1NF&2NF&3NF&BNCF:**
    Exclusive (ExclusiveID, Mode, ExpiryDate, AnnouncementID)
    EligibleMembers (AnnouncementID, StudentID)

16. Sport (SportID, SportType, ClubID)
    FD1: SportID → SportType, ClubID
    **1NF&2NF&3NF&BNCF:**
    Sport (SportID, SportType, ClubID)

17. Academic (AcademicID, StudyHour, FacultyAdvisor, ClubID)
    FD1: AcademicID → StudyHour, FacultyAdvisor, ClubID
    **1NF&2NF&3NF&BNCF:**
    Academic (AcademicID, StudyHour, ClubID)
    Club Advisor (ClubID, FacultyAdvisor)

18. Cultural (CulturalID, CultureCategory, ClubID)
    FD1: CulturalID → CultureCategory, ClubID
    **1NF&2NF&3NF&BNCF:**
    Cultural (CulturalID, CultureCategory, ClubID)

**Remark: Underline word is primary key.*

## 5.0 Relational DB Schemas (after normalization)

The relational database schema for the ClubHub System is a collection of relation schemas that includes the following components.

User (UserID, Username, FullName, FirstName, LastName, DOB, Password, Email, PhoneNumber)

Student (StudentID, Faculty , UserID)

Preferences (PreferenceID, Category, ParticipationCost, Location, Date, StudentID, EventID)

Payment (ReceiptID, TotalAmount, PaymentDate, SenderID, OrganizerID)

Organizer (OrganizerID, Role, UserID, ClubID)

Event (EventID, EventDescription, EventDate, EventLocation, EventCost, ClubID)

Club (ClubID, ClubName, Category, MemberSize, OrganizerID)

Announcement (AnnouncementID, AnnouncementDescription, AnnouncementTitle, AnnouncementType, OrganizerID)

Administrator (AdministratorID, Role, OrganizerID)

Participation Certificates (CertificateID, CertificateDescription, CertificateIssuedDate, EventID, StudentID)

Registration (RegistrationID, RegistrationDate, OrganizerID, StudentID, EventID)

Cash (CashID, ReceiptID)

Debit Card (CardNumber, BankName, PIN-Number, CardExpiryDate, CVVNumber, ReceiptID)

Public (PublicID, AnnouncementID)

Exclusive (ExclusiveID, Mode, ExpiryDate, AnnouncementID)

Eligible Members (AnnouncementID, EligibleMemberID)

Sport (SportID, SportType, ClubID)

Academic (AcademicID, StudyHour, ClubID)

Club Advisor (ClubID, FacultyAdvisor)

Cultural (CulturalID, CultureCategory, ClubID)

**Remark: Underline word is primary key.*

### User

| UserID | UserName | FirstName | LastName | DOB | Password | Email | PhoneNumber |
|---|---|---|---|---|---|---|---|

### Student

| StudentID | Faculty | UserID |
|---|---|---|

### Preferences

| PreferenceID | Category | ParticipationCost | Location | Date | StudentID | EventID |
|---|---|---|---|---|---|---|

### Payment

| ReceiptID | TotalAmount | PaymentDate | SenderID | OrganizerID |
|---|---|---|---|---|

### Organizer

| OrganizerID | Role | UserID | ClubID |
|---|---|---|---|

### Event

| EventID | EventDescription | EventDate | EventLocation | EventCost | ClubID |
|---|---|---|---|---|---|

### Club

| ClubID | ClubName | Category | MemberSize | OrganizerID |
|---|---|---|---|---|

### Announcement

| AnnouncementID | Announcement Description | AnnouncementTitle | Announcement Type | OrganizerID |
|---|---|---|---|---|

### Administrator

| AdministratorID | Role | OrganizerID |
|---|---|---|

### Participation Certificates

| CertificateID | CertificateDescription | CertificateIssuedDate | EventID | StudentID |
|---|---|---|---|---|

**Registration**

| RegistrationID | RegistrationDate | OrganizerID | StudentID | EventID |
|---|---|---|---|---|

**Cash**

| CashID | ReceiptID |
|---|---|

**Debit Card**

| CardNumber | BankName | PIN-Number | CardExpiryDate | CVVNumber | ReceiptID |
|---|---|---|---|---|---|

**Public**

| PublicID | AnnouncementID |
|---|---|

**Exclusive**

| ExclusiveID | Mode | ExpiryDate | AnnouncementID |
|---|---|---|---|

**Eligible Members**

| AnnouncementID | EligibleMemberID |
|---|---|

**Sport**

| SportID | SportType | ClubID |
|---|---|---|

**Academic**

| AcademicID | StudyHour | ClubID |
|---|---|---|

**Club Advisor**

| ClubID | FacultyAdvisor |
|---|---|

**Cultural**

| CulturalID | CultureCategory | ClubID |
|---|---|---|

# 6.0 SQL Statements (DDL & DML)

## DDL:-

```
-- User Table
CREATE TABLE User (
    UserID VARCHAR(10) PRIMARY KEY,
    Username VARCHAR(20) NOT NULL,
    FullName VARCHAR(100) NOT NULL,
    FirstName VARCHAR(30) NOT NULL,
    LastName VARCHAR(30) NOT NULL,
    DOB DATE NOT NULL,
    Password VARCHAR(100) NOT NULL,
    Email VARCHAR(320),
    PhoneNumber VARCHAR(15) NOT NULL
);

-- Student Table
CREATE TABLE Student (
    StudentID VARCHAR(10) PRIMARY KEY,
    Faculty VARCHAR(50) NOT NULL,
    UserID VARCHAR(10),
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);

-- Preferences Table
CREATE TABLE Preferences (
    PreferenceID VARCHAR(10) PRIMARY KEY,
    Category VARCHAR(20) NOT NULL,
    ParticipationCost FLOAT NOT NULL,
    Location VARCHAR(50) NOT NULL,
    Date DATE NOT NULL,
    StudentID VARCHAR(10),
    EventID VARCHAR(10),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    FOREIGN KEY (EventID) REFERENCES Event(EventID)
);

-- Payment Table
CREATE TABLE Payment (
    ReceiptID VARCHAR(10) PRIMARY KEY,
    TotalAmount FLOAT NOT NULL,
    PaymentDate DATE NOT NULL,
    SenderID VARCHAR(10),
    OrganizerID VARCHAR(10),
    FOREIGN KEY (OrganizerID) REFERENCES Organizer(OrganizerID)
);
```

```
-- Organizer Table
CREATE TABLE Organizer (
    OrganizerID VARCHAR(10) PRIMARY KEY,
    Role VARCHAR(30),
    UserID VARCHAR(10),
    ClubID VARCHAR(10),
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID)
);

-- New Event Table (Modified Structure)
CREATE TABLE Event (
    EventID VARCHAR(10) PRIMARY KEY,
    EventName VARCHAR(100) NOT NULL, -- Added new column
    EventDescription VARCHAR(300), -- Shortened the length
    EventLocation VARCHAR(100), -- Increased length
    EventStartDate DATE, -- Renamed from EventDate
    EventEndDate DATE, -- Added new column
    EventCost FLOAT
);

-- Club Table
CREATE TABLE Club (
    ClubID VARCHAR(10) PRIMARY KEY,
    ClubName VARCHAR(50) NOT NULL,
    Category VARCHAR(20) NOT NULL,
    MemberSize INT NOT NULL,
    OrganizerID VARCHAR(10),
    FOREIGN KEY (OrganizerID) REFERENCES Organizer(OrganizerID)
);

-- Announcement Table
CREATE TABLE Announcement (
    AnnouncementID VARCHAR(10) PRIMARY KEY,
    AnnouncementDescription VARCHAR(1000) NOT NULL,
    AnnouncementTitle VARCHAR(100) NOT NULL,
    AnnouncementType VARCHAR(30) NOT NULL,
    OrganizerID VARCHAR(10),
    FOREIGN KEY (OrganizerID) REFERENCES Organizer(OrganizerID)
);

-- Administrator Table
CREATE TABLE Administrator (
    AdministratorID VARCHAR(10) PRIMARY KEY,
    Role VARCHAR(30),
    OrganizerID VARCHAR(10),
    FOREIGN KEY (OrganizerID) REFERENCES Organizer(OrganizerID)
);
```

```
-- Participation Certificates Table
CREATE TABLE ParticipationCertificates (
    CertificateID VARCHAR(10) PRIMARY KEY,
    CertificateDescription VARCHAR(500) NOT NULL,
    CertificateIssuedDate DATE NOT NULL,
    EventID VARCHAR(10),
    StudentID VARCHAR(10),
    FOREIGN KEY (EventID) REFERENCES Event(EventID),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
);

-- Registration Table
CREATE TABLE Registration (
    RegistrationID VARCHAR(10) PRIMARY KEY,
    RegistrationDate DATE NOT NULL,
    OrganizerID VARCHAR(10),
    StudentID VARCHAR(10),
    EventID VARCHAR(10),
    FOREIGN KEY (OrganizerID) REFERENCES Organizer(OrganizerID),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    FOREIGN KEY (EventID) REFERENCES Event(EventID)
);

-- Cash Table
CREATE TABLE Cash (
    CashID VARCHAR(10) PRIMARY KEY,
    ReceiptID VARCHAR(10),
    FOREIGN KEY (ReceiptID) REFERENCES Payment(ReceiptID)
);

-- Debit Card Table
CREATE TABLE DebitCard (
    CardNumber VARCHAR(16) PRIMARY KEY,
    BankName VARCHAR(50) NOT NULL,
    PINNumber INT NOT NULL,
    CardExpiryDate DATE NOT NULL,
    CVVNumber VARCHAR(4) NOT NULL,
    ReceiptID VARCHAR(10),
    FOREIGN KEY (ReceiptID) REFERENCES Payment(ReceiptID)
);

-- Public Table
CREATE TABLE Public (
    PublicID VARCHAR(10) PRIMARY KEY,
    AnnouncementID VARCHAR(10),
    FOREIGN KEY (AnnouncementID) REFERENCES
Announcement(AnnouncementID)
);
```

```sql
-- Exclusive Table
CREATE TABLE Exclusive (
    ExclusiveID VARCHAR(10) PRIMARY KEY,
    Mode VARCHAR(20) NOT NULL,
    ExpiryDate DATE NOT NULL,
    AnnouncementID VARCHAR(10),
    FOREIGN KEY (AnnouncementID) REFERENCES
Announcement(AnnouncementID)
);

-- Eligible Members Table
CREATE TABLE EligibleMembers (
    AnnouncementID VARCHAR(10),
    EligibleMemberID VARCHAR(10),
    FOREIGN KEY (AnnouncementID) REFERENCES
Announcement(AnnouncementID),
    PRIMARY KEY (AnnouncementID, EligibleMemberID)
);

-- Sport Table
CREATE TABLE Sport (
    SportID VARCHAR(10) PRIMARY KEY,
    SportType VARCHAR(50) NOT NULL,
    ClubID VARCHAR(10),
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID)
);

-- Academic Table
CREATE TABLE Academic (
    AcademicID VARCHAR(10) PRIMARY KEY,
    StudyHour INT NOT NULL,
    ClubID VARCHAR(10),
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID)
);

-- Club Advisor Table
CREATE TABLE ClubAdvisor (
    ClubID VARCHAR(10),
    FacultyAdvisor VARCHAR(100),
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID),
    PRIMARY KEY (ClubID, FacultyAdvisor)
);

-- Cultural Table
CREATE TABLE Cultural (
    CulturalID VARCHAR(10) PRIMARY KEY,
    CultureCategory VARCHAR(50) NOT NULL,
    ClubID VARCHAR(10),
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID)
);
```

```
-- Step 1: Rename Column (EventStartDate -> EventDate)
ALTER TABLE Event
CHANGE EventStartDate EventDate DATE;

-- Step 2: Remove New Columns (EventName, EventEndDate)
ALTER TABLE Event
DROP COLUMN EventName,
DROP COLUMN EventEndDate;

-- Step 3: Modify Column Length (EventDescription)
ALTER TABLE Event
MODIFY EventDescription VARCHAR(500);

-- Step 4: Modify Column Length (EventLocation)
ALTER TABLE Event
MODIFY EventLocation VARCHAR(50);

-- Step 5: Add Foreign Key for ClubID
ALTER TABLE Event
ADD ClubID VARCHAR(10); -- Adding back the column

ALTER TABLE Event
ADD CONSTRAINT fk_club
FOREIGN KEY (ClubID) REFERENCES Club(ClubID);
```

## DML:
## insert statements:-

```
-- 1. USER MANAGEMENT
-- Initial users
INSERT INTO User (UserID, Username, FullName, FirstName, LastName, DOB,
Password, Email, PhoneNumber)
VALUES
('U001', 'john_doe', 'John Doe', 'John', 'Doe', '2000-05-16',
'password123', 'john.doe@example.com', '0123456789'),
('U002', 'jane_smith', 'Jane Smith', 'Jane', 'Smith', '2001-03-22',
'securepass', 'jane.smith@example.com', '0198765432'),
('U003', 'alex_jones', 'Alex Jones', 'Alex', 'Jones', '2002-07-10',
'mypassword', 'alex.jones@example.com', '0185647382');

-- Add new user
INSERT INTO User (UserID, Username, FullName, FirstName, LastName, DOB,
Password, Email, PhoneNumber)
VALUES ('U004', 'emma_davis', 'Emma Davis', 'Emma', 'Davis',
'2001-08-15', 'emma123', 'emma.davis@example.com', '0187654321');

-- View all users
SELECT UserID, Username, FullName, Email FROM User;

/* The Output:
UserID | Username    | FullName     | Email
--------|-------------|--------------|----------------------
U001   | john_doe   | John Doe     | john.doe@example.com
U002   | jane_smith | Jane Smith   | jane.smith@example.com
U003   | alex_jones | Alex Jones   | alex.jones@example.com
U004   | emma_davis | Emma Davis   | emma.davis@example.com
*/
```

```
-- 2. CLUB MANAGEMENT
-- Initial clubs and organizers
INSERT INTO Organizer (OrganizerID, Role, UserID, ClubID)
VALUES
('O001', 'President', 'U001', NULL),
('O002', 'Vice President', 'U002', NULL);

INSERT INTO Club (ClubID, ClubName, Category, MemberSize, OrganizerID)
VALUES
('C001', 'Tech Club', 'Technology', 100, 'O001'),
('C002', 'Science Society', 'Science', 50, 'O002');

-- Create new club
INSERT INTO Club (ClubID, ClubName, Category, MemberSize, OrganizerID)
VALUES ('C003', 'Photography Club', 'Arts', 30, 'O001');

-- View all clubs
SELECT c.ClubID, c.ClubName, c.Category, c.MemberSize,
CONCAT(u.FirstName, ' ', u.LastName) as Organizer
FROM Club c
JOIN Organizer o ON c.OrganizerID = o.OrganizerID
JOIN User u ON o.UserID = u.UserID;

/* The Output:
ClubID | ClubName         | Category   | MemberSize | Organizer
-------|-----------------|------------|------------|------------
C001   | Tech Club       | Technology | 100        | John Doe
C002   | Science Society | Science    | 50         | Jane Smith
C003   | Photography Club| Arts       | 30         | John Doe
*/
```

```
-- 3. EVENT MANAGEMENT
-- Initial events
INSERT INTO Event (EventID, EventDescription, EventDate, EventLocation,
EventCost, ClubID)
VALUES
('E001', 'Tech Workshop 2025', '2025-02-01', 'Lab 101', 50.00, 'C001'),
('E002', 'Science Exhibition', '2025-03-15', 'Main Hall', 75.00,
'C002');

-- Create new event
INSERT INTO Event (EventID, EventDescription, EventDate, EventLocation,
EventCost, ClubID)
VALUES ('E003', 'Photography Masterclass', '2025-04-20', 'Studio A',
100.00, 'C003');

-- Update existing event details
UPDATE Event
SET EventCost = 60.00,
    EventLocation = 'Lab 102'
WHERE EventID = 'E001';

-- Cancel an event
UPDATE Event
SET EventDescription = 'CANCELLED - ' || EventDescription
WHERE EventID = 'E002';

-- View all upcoming events
SELECT
    e.EventID,
    e.EventDescription,
    e.EventDate,
    e.EventLocation,
    e.EventCost,
    c.ClubName
FROM Event e
JOIN Club c ON e.ClubID = c.ClubID
WHERE e.EventDate >= CURRENT_DATE
ORDER BY e.EventDate;

/* The Output:
EventID | EventDescription         | EventDate  | EventLocation | EventCost | ClubName
--------|------------------------|------------|---------------|-----------|------------------
E001    | Tech Workshop 2025     | 2025-02-01 | Lab 102       | 60.00     | Tech Club
E002    | CANCELLED - Science... | 2025-03-15 | Main Hall     | 75.00     | Science Society
E003    | Photography Masterclass| 2025-04-20 | Studio A      | 100.00    | Photography Club
*/
```

```
-- 4. REGISTRATION MANAGEMENT
-- Initial registrations
INSERT INTO Registration (RegistrationID, RegistrationDate,
OrganizerID, StudentID, EventID)
VALUES
('R001', '2025-01-15', 'O001', 'S001', 'E001'),
('R002', '2025-01-16', 'O002', 'S002', 'E001');

-- Register new student for event
INSERT INTO Registration (RegistrationID, RegistrationDate,
OrganizerID, StudentID, EventID)
VALUES ('R003', CURRENT_DATE, 'O001', 'S003', 'E003');

-- Cancel registration for cancelled event
DELETE FROM Registration
WHERE EventID = 'E002';

-- View current registrations
SELECT
    r.RegistrationID,
    CONCAT(u.FirstName, ' ', u.LastName) as StudentName,
    e.EventDescription,
    c.ClubName,
    r.RegistrationDate
FROM Registration r
JOIN Student s ON r.StudentID = s.StudentID
JOIN User u ON s.UserID = u.UserID
JOIN Event e ON r.EventID = e.EventID
JOIN Club c ON e.ClubID = c.ClubID;



/* The Output:
RegistrationID | StudentName | EventDescription    | RegistrationDate
---------------|-------------|--------------------|-----------------
R001           | John Doe    | Tech Meetup        | 2025-01-25
R002           | Jane Smith  | Science Fair       | 2025-02-10
R003           | Alex Jones  | Photography Workshop| 2025-01-16
*/
```

```sql
-- 5. ANNOUNCEMENT MANAGEMENT
-- Initial announcements
INSERT INTO Announcement (AnnouncementID, AnnouncementDescription,
AnnouncementTitle, AnnouncementType, OrganizerID)
VALUES
('A001', 'Annual meeting for members', 'Annual Meeting', 'General',
'O001'),
('A002', 'Upcoming Science Fair details', 'Science Fair', 'Event',
'O002');

-- Set initial announcement visibility
INSERT INTO Public (PublicID, AnnouncementID)
VALUES ('PUB001', 'A001');

INSERT INTO Exclusive (ExclusiveID, Mode, ExpiryDate, AnnouncementID)
VALUES ('EXC001', 'Members Only', '2025-12-31', 'A002');

-- View initial announcements
SELECT
    a.AnnouncementTitle,
    a.AnnouncementDescription,
    a.AnnouncementType,
    CASE
        WHEN p.PublicID IS NOT NULL THEN 'Public'
        WHEN e.ExclusiveID IS NOT NULL THEN 'Exclusive'
    END as Visibility,
    CONCAT(u.FirstName, ' ', u.LastName) AS AnnouncedBy
FROM Announcement a
JOIN Organizer o ON a.OrganizerID = o.OrganizerID
JOIN User u ON o.UserID = u.UserID
LEFT JOIN Public p ON a.AnnouncementID = p.AnnouncementID
LEFT JOIN Exclusive e ON a.AnnouncementID = e.AnnouncementID;

/* The Output:
AnnouncementTitle | AnnouncementDescription    | AnnouncementType | Visibility | AnnouncedBy
-----------------|--------------------------|------------------|------------|-------------
Annual Meeting    | Annual meeting for members| General          | Public     | John Doe
Science Fair      | Upcoming Science Fair...  | Event            | Exclusive  | Jane Smith
*/

-- Create new announcement
INSERT INTO Announcement (AnnouncementID, AnnouncementDescription,
AnnouncementTitle, AnnouncementType, OrganizerID)
VALUES ('A003', 'Join us for a photography exhibition!', 'Photo
Exhibition', 'Event', 'O001');

-- Make new announcement public
INSERT INTO Public (PublicID, AnnouncementID)
VALUES ('PUB003', 'A003');
```

```sql
-- View all public announcements
SELECT
    a.AnnouncementTitle,
    a.AnnouncementDescription,
    a.AnnouncementType,
    CONCAT(u.FirstName, ' ', u.LastName) AS AnnouncedBy
FROM Announcement a
JOIN Public p ON a.AnnouncementID = p.AnnouncementID
JOIN Organizer o ON a.OrganizerID = o.OrganizerID
JOIN User u ON o.UserID = u.UserID;

/* The Output:
AnnouncementTitle | AnnouncementDescription        | AnnouncementType | AnnouncedBy
------------------|-------------------------------|------------------|-------------
Annual Meeting    | Annual meeting for members    | General          | John Doe
Photo Exhibition  | Join us for a photo exhibition| Event            | John Doe
*/
```

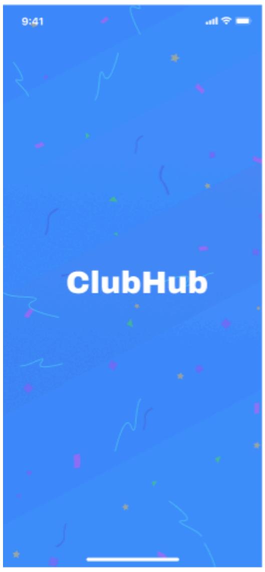
```sql
-- 6. PAYMENT PROCESSING
-- Initial payments
INSERT INTO Payment (ReceiptID, TotalAmount, PaymentDate, SenderID,
OrganizerID)
VALUES
('PAY001', 100.00, '2025-02-01', 'S001', 'O001'),
('PAY002', 50.00, '2025-03-15', 'S002', 'O002');

-- Record initial payment methods
INSERT INTO Cash (CashID, ReceiptID)
VALUES
('CASH001', 'PAY001'),
('CASH002', 'PAY002');

-- View initial payments
SELECT
    p.ReceiptID,
    p.TotalAmount,
    p.PaymentDate,
    'Cash' as PaymentMethod,
    CONCAT(u.FirstName, ' ', u.LastName) AS PaidBy
FROM Payment p
JOIN Cash c ON p.ReceiptID = c.ReceiptID
JOIN Student s ON p.SenderID = s.StudentID
JOIN User u ON s.UserID = u.UserID;

/* The Output:
ReceiptID | TotalAmount | PaymentDate | PaymentMethod | PaidBy
-----------|-------------|-------------|---------------|-------------
PAY001     | 100.00       | 2025-02-01 | Cash           | John Doe
PAY002     | 50.00        | 2025-03-15 | Cash           | Jane Smith
*/

-- Record new payment
INSERT INTO Payment (ReceiptID, TotalAmount, PaymentDate, SenderID,
OrganizerID)
VALUES ('PAY003', 75.00, CURRENT_DATE, 'S003', 'O001');

-- Record payment method (Debit Card)
INSERT INTO DebitCard (CardNumber, BankName, PINNumber, CardExpiryDate,
CVVNumber, ReceiptID)
VALUES ('4567890123456789', 'Bank C', 9012, '2026-12-01', '456',
'PAY003');
```

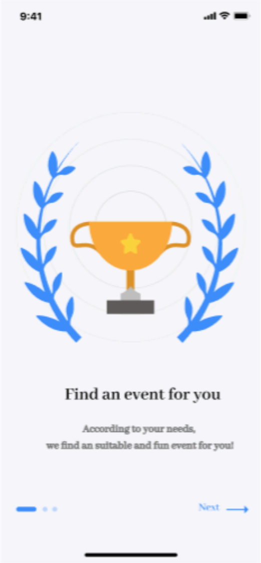
```
-- View all payments with payment methods
SELECT
    p.ReceiptID,
    p.TotalAmount,
    p.PaymentDate,
    CASE
        WHEN c.CashID IS NOT NULL THEN 'Cash'
        WHEN d.CardNumber IS NOT NULL THEN 'Debit Card'
    END AS PaymentMethod,
    CONCAT(u.FirstName, ' ', u.LastName) AS PaidBy
FROM Payment p
LEFT JOIN Cash c ON p.ReceiptID = c.ReceiptID
LEFT JOIN DebitCard d ON p.ReceiptID = d.ReceiptID
JOIN Student s ON p.SenderID = s.StudentID
JOIN User u ON s.UserID = u.UserID
ORDER BY p.PaymentDate;

/* The Output:
ReceiptID | TotalAmount | PaymentDate | PaymentMethod | PaidBy
-----------|-------------|-------------|---------------|-------------
PAY001     | 100.00      | 2025-02-01 | Cash          | John Doe
PAY002     | 50.00       | 2025-03-15 | Cash          | Jane Smith
PAY003     | 75.00       | 2025-01-16 | Debit Card    | Alex Jones
*/

-- Additional Payment Management Functions
-- Cancel/Refund a payment
UPDATE Payment
SET TotalAmount = 0.00,
    PaymentDate = CURRENT_DATE
WHERE ReceiptID = 'PAY002';

-- View payments by date range
SELECT
    p.ReceiptID,
    p.TotalAmount,
    p.PaymentDate,
    CASE
        WHEN c.CashID IS NOT NULL THEN 'Cash'
        WHEN d.CardNumber IS NOT NULL THEN 'Debit Card'
    END AS PaymentMethod,
    CONCAT(u.FirstName, ' ', u.LastName) AS PaidBy
FROM Payment p
LEFT JOIN Cash c ON p.ReceiptID = c.ReceiptID
LEFT JOIN DebitCard d ON p.ReceiptID = d.ReceiptID
JOIN Student s ON p.SenderID = s.StudentID
JOIN User u ON s.UserID = u.UserID
WHERE p.PaymentDate BETWEEN '2025-01-01' AND '2025-12-31'
ORDER BY p.PaymentDate;
```

```
/* The Output:

+------------+-------------+-------------+---------------+-------------
+
| ReceiptID  | TotalAmount | PaymentDate | PaymentMethod | PaidBy
|
+------------+-------------+-------------+---------------+-------------
+
| PAY003     | 75.00       | 2025-01-16  | Debit Card    | Alex Jones
|
| PAY002     | 0.00        | 2025-01-16  | Cash          | Jane Smith
|
| PAY001     | 100.00      | 2025-02-01  | Cash          | John Doe
|
+------------+-------------+-------------+---------------+-------------

*/+-+
```

# 7.0 Interface

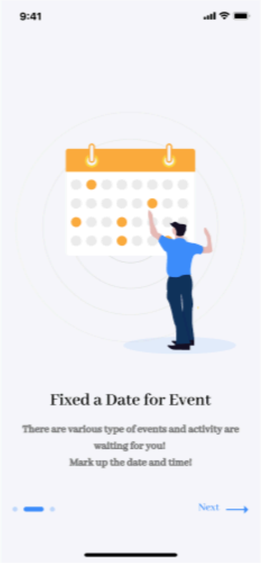
## 7.1 User Login


Splash screen


Onboarding 4

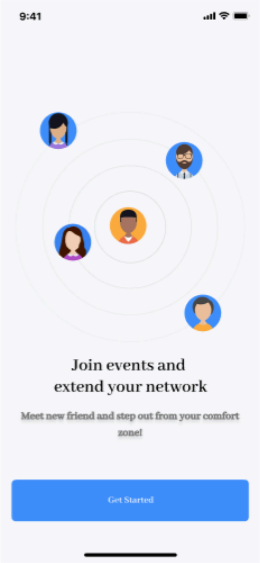
Find an event for you

According to your needs, we find an suitable and fun event for you!


Onboarding 2

Fixed a Date for Event

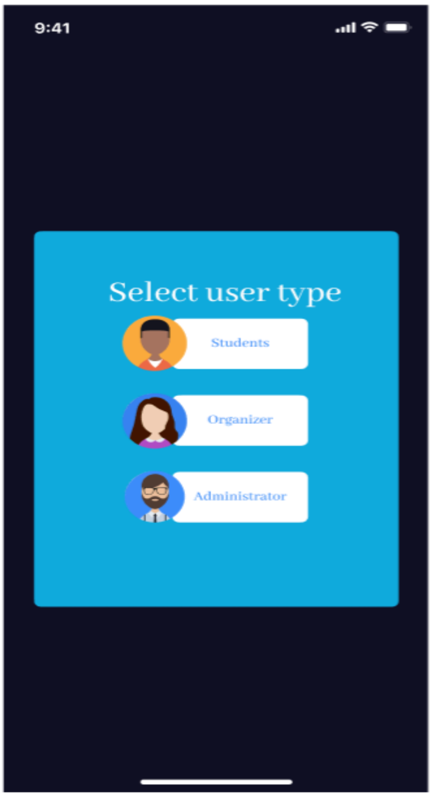
There are various type of events and activity are waiting for you! Mark up the date and time!

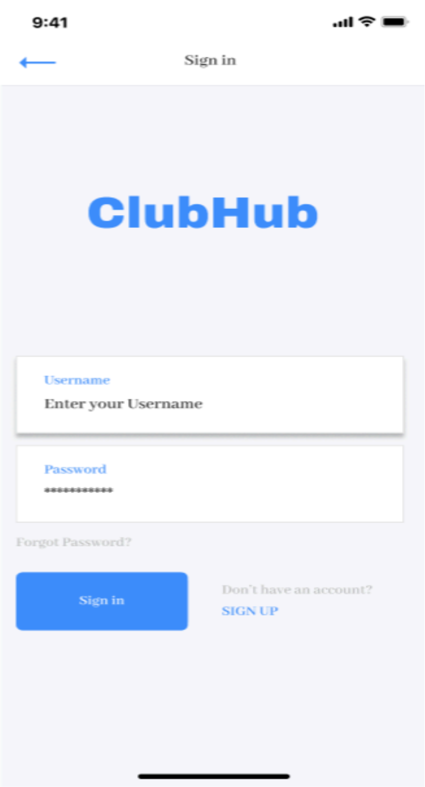

Onboarding 3

Join events and extend your network

Meet new friend and step out from your comfort zone!

## 7.2 User Sign In


Landing screen

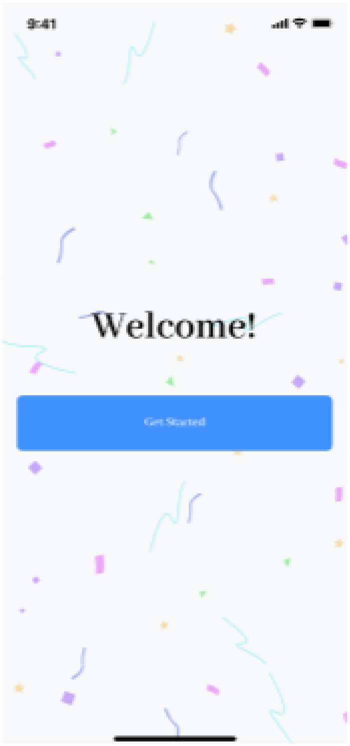

Sign in

Username: Enter your Username

Password: ************

## 7.3 Successful User Sign In

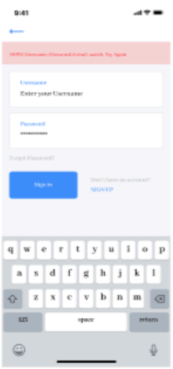
Sign up Success



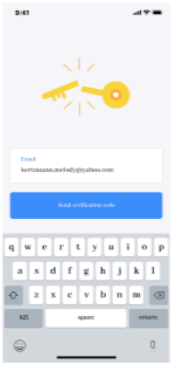
Welcome!

## 7.4 Failed User Sign in and Email verification pages

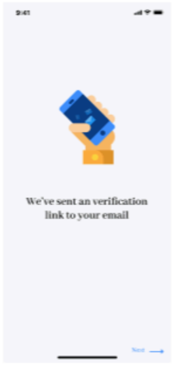
Sign in failure



Forgot password

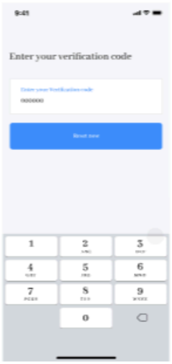


Email verificati...



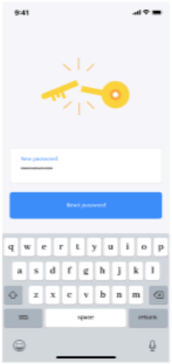
We've sent an verification link to your email

Forgot passwo...

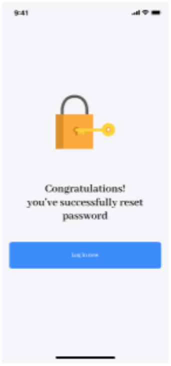


Enter your verification code

Reset password

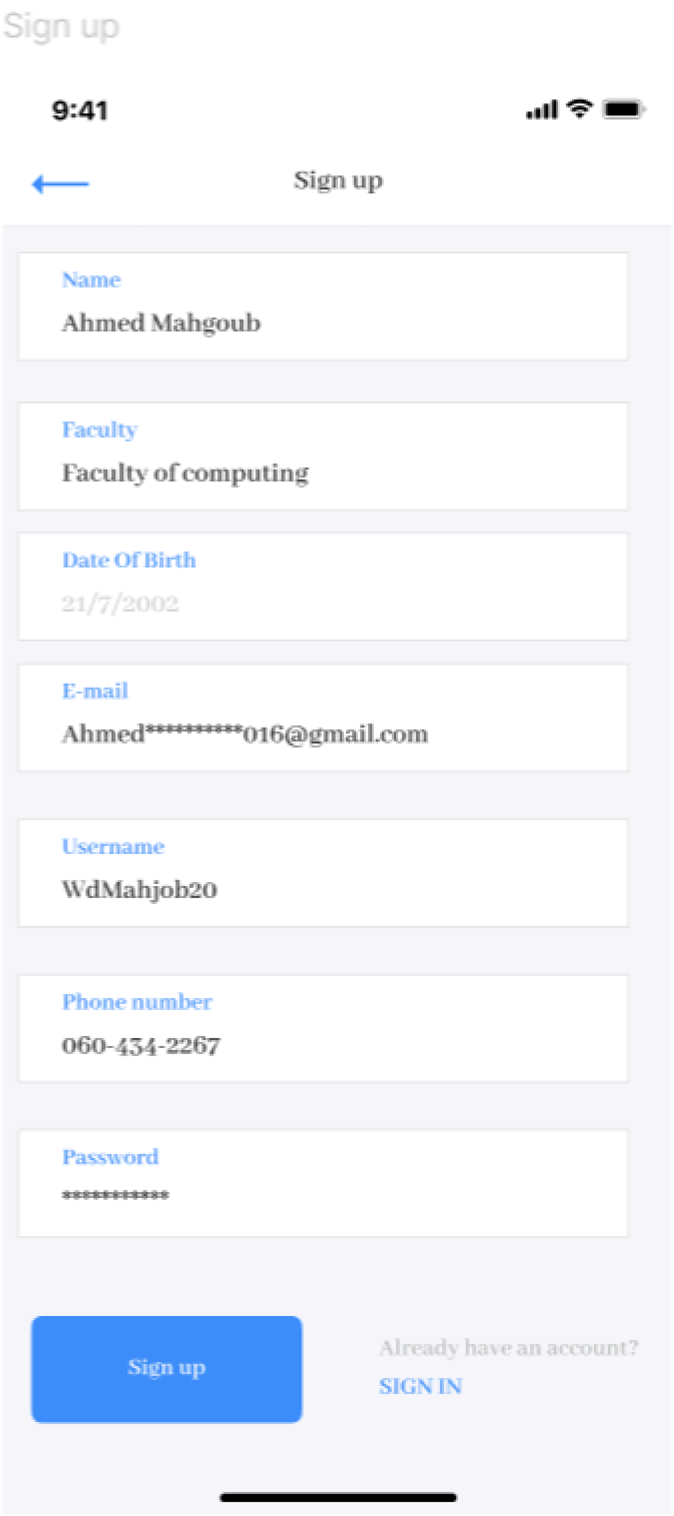


Forgot passwo...



Congratulations!
you've successfully reset password

7.5 User Sign-Up page



Sign up

9:41

Sign up

Name
Ahmed Mahgoub

Faculty
Faculty of computing

Date Of Birth
21/7/2002

E-mail
Ahmed**********016@gmail.com

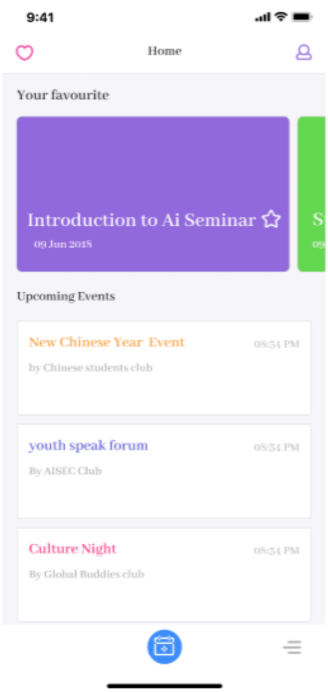
Username
WdMahjob20

Phone number
060-434-2267

Password
************

## 7.6 Student View (Home page)

Home User



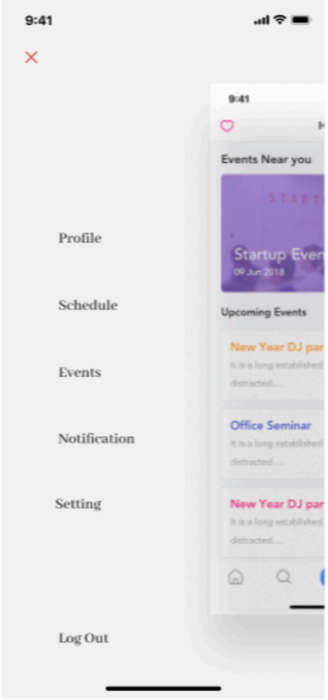
Introduction to Ai Seminar
09 Jun 2018

New Chinese Year Event 08:54 PM
by Chinese students club

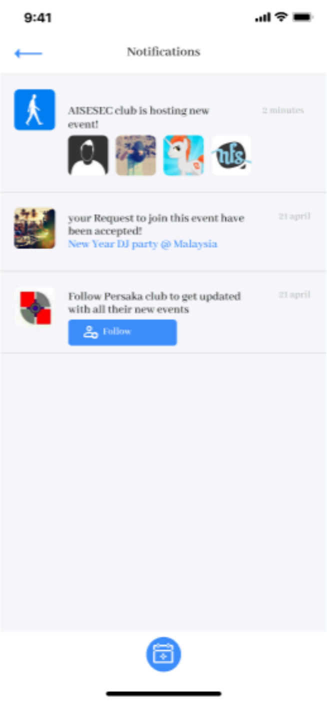
youth speak forum 08:54 PM
By AISEC Club

Culture Night 08:54 PM
By Global Buddies club

nav bar



Notification



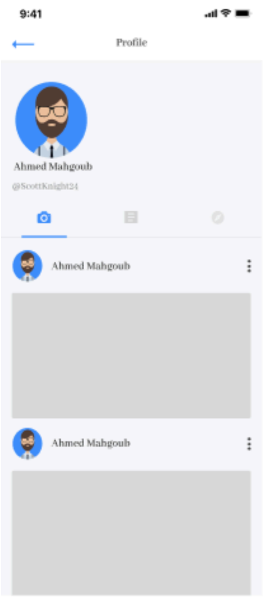
AISESEC club is hosting new event! 2 minutes

your Request to join this event have been accepted! 21 april
New Year DJ party @ Malaysia

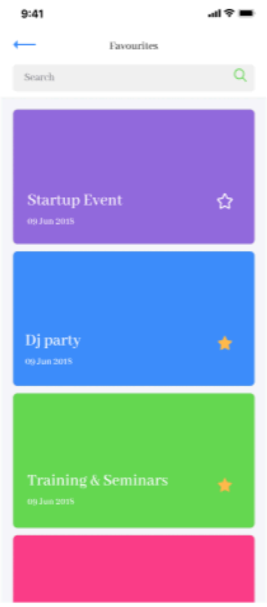
Follow Persaka club to get updated with all their new events 21 april

## 7.7 Student Profile

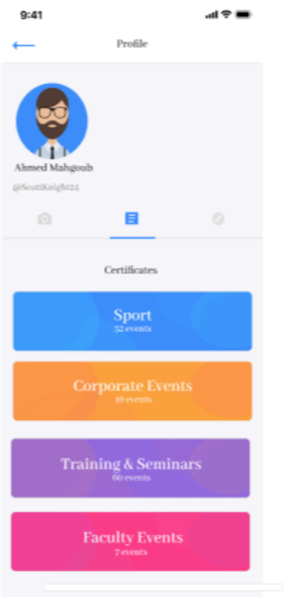
Account



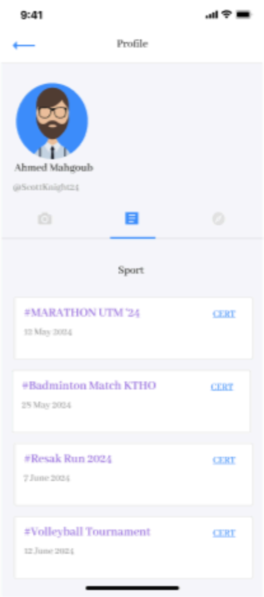
Favorite



Account



Account



#MARATHON UTM '24 CERT
12 May 2024

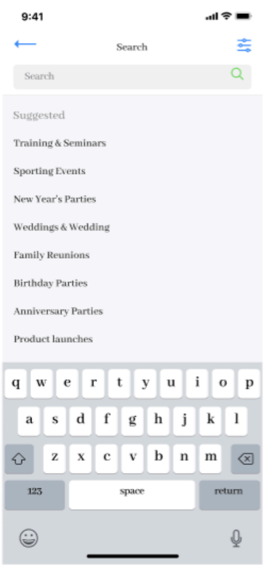
#Badminton Match KTHO CERT
25 May 2024

#Resak Run 2024 CERT
7 June 2024

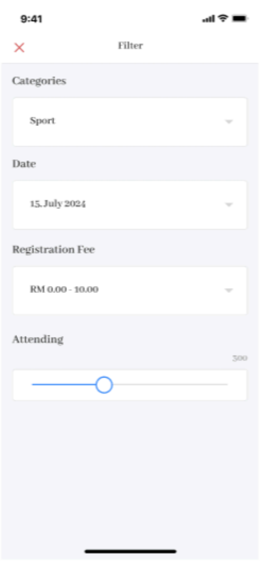
#Volleyball Tournament CERT
12 June 2024

## 7.8 Searching and Filtering Events (Preferences)

Search



Filtering



Categories

Sport

Date

15 July 2024

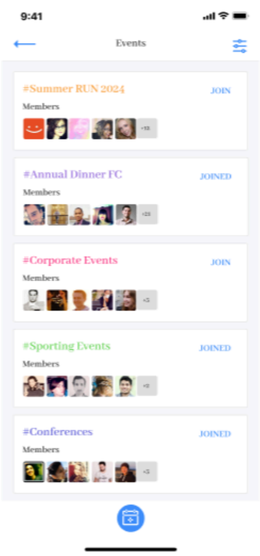
Registration Fee

RM 0.00 - 10.00

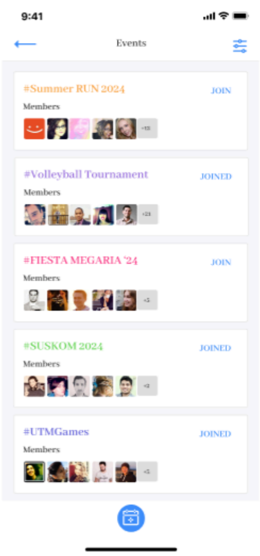
Attending

500

Events(before filter)

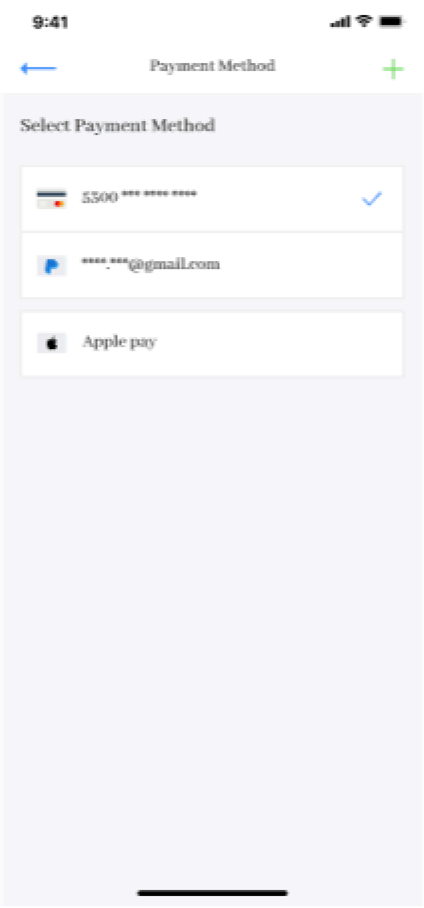


Events(after filter)



## 7.9 Making Payment

Payment Method (settin...



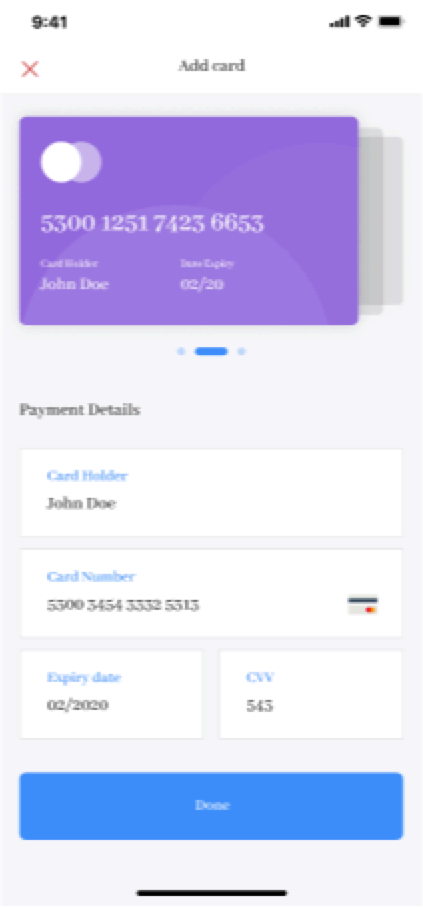
Select Payment Method

5300 **** **** **** ✓

****.***@gmail.com

Apple pay

Add Card(setting)



Payment Details

Card Holder
John Doe

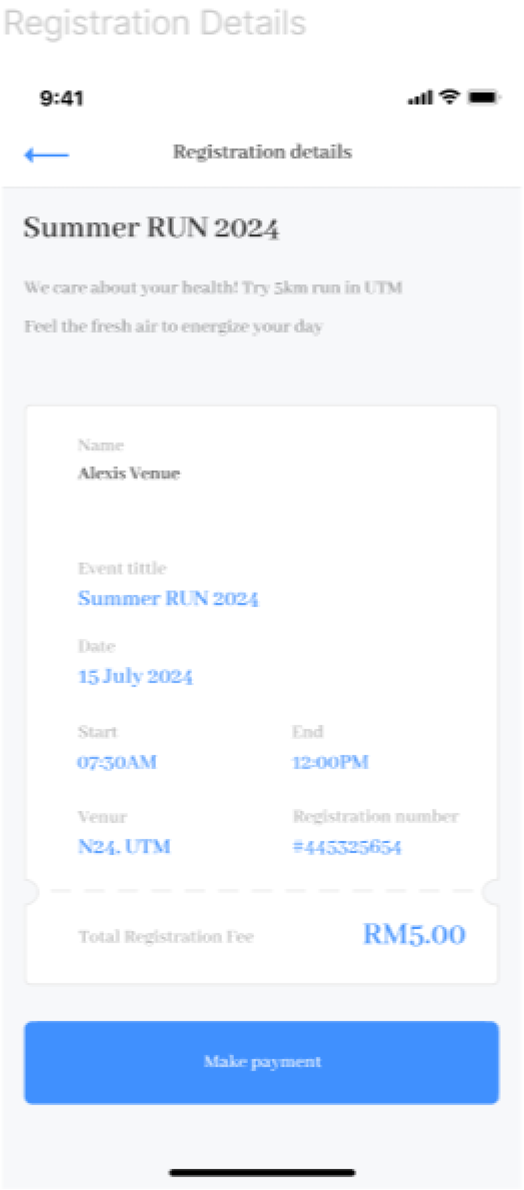
Card Number
5300 3454 3332 5515

Expiry date
02/2020

CVV
545

Done

7.10 Event Registration Details



Registration Details

9:41

Registration details

Summer RUN 2024

We care about your health! Try 5km run in UTM

Feel the fresh air to energize your day

Name
Alexis Venue

Event tittle
Summer RUN 2024

Date
15 July 2024

Start
07:30AM

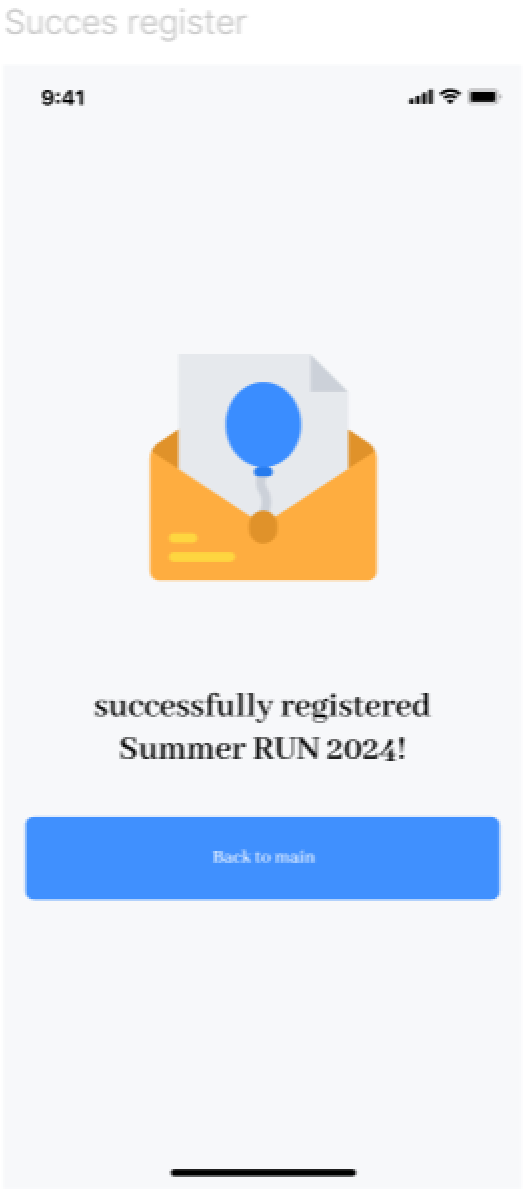
End
12:00PM

Venur
N24, UTM

Registration number
#445325654

Total Registration Fee
RM5.00

### 7.11 Successful Event Registration

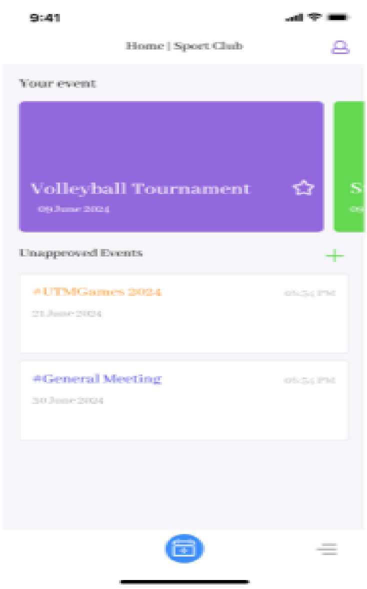


Succes register

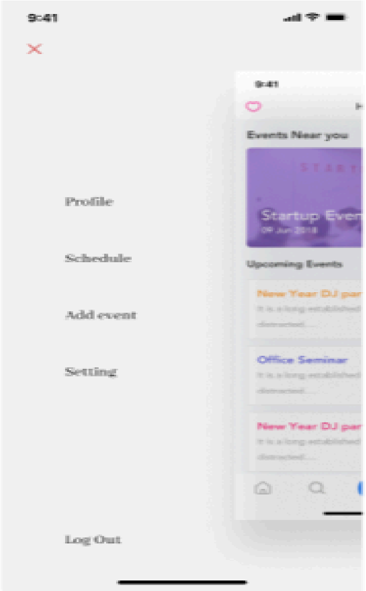
successfully registered
Summer RUN 2024!

## 7.12 Organizer View (Home Page)

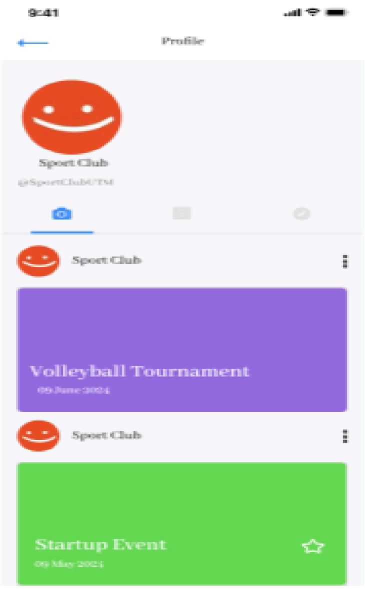
Home organizer



nav bar

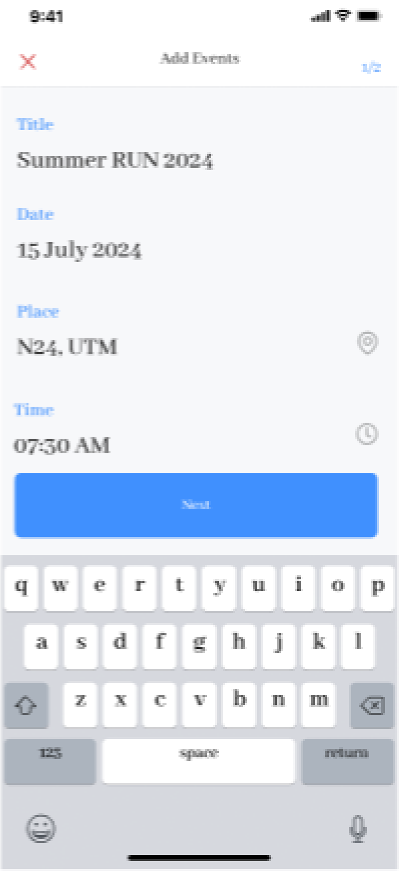


Account



## 7.13 Create New Event

Create Event



Title
Summer RUN 2024

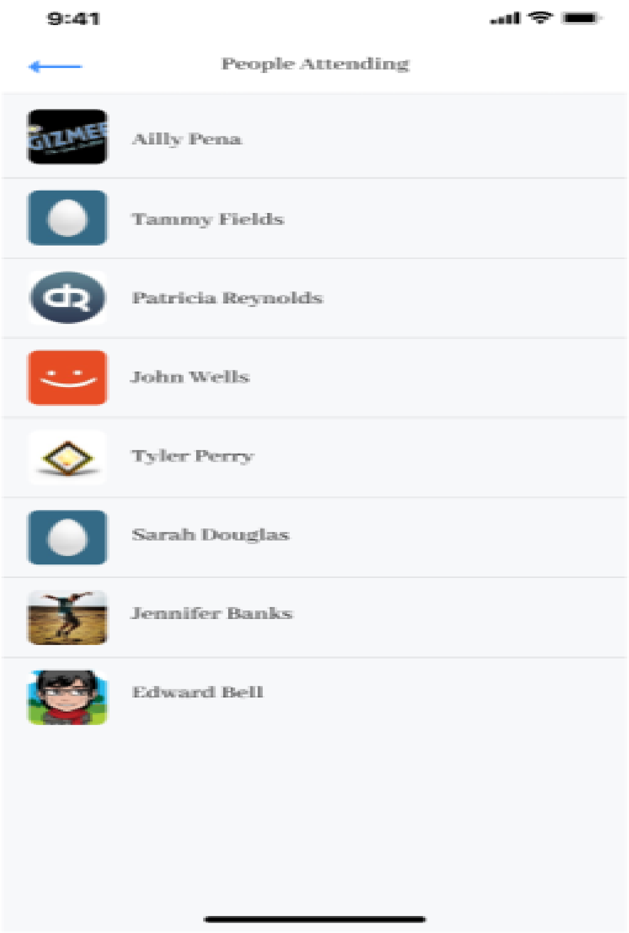
Date
15 July 2024

Place
N24, UTM

Time
07:30 AM

7.14 View Event Attendance List

Who is going list



9:41

People Attending

Ailly Pena

Tammy Fields

Patricia Reynolds

John Wells

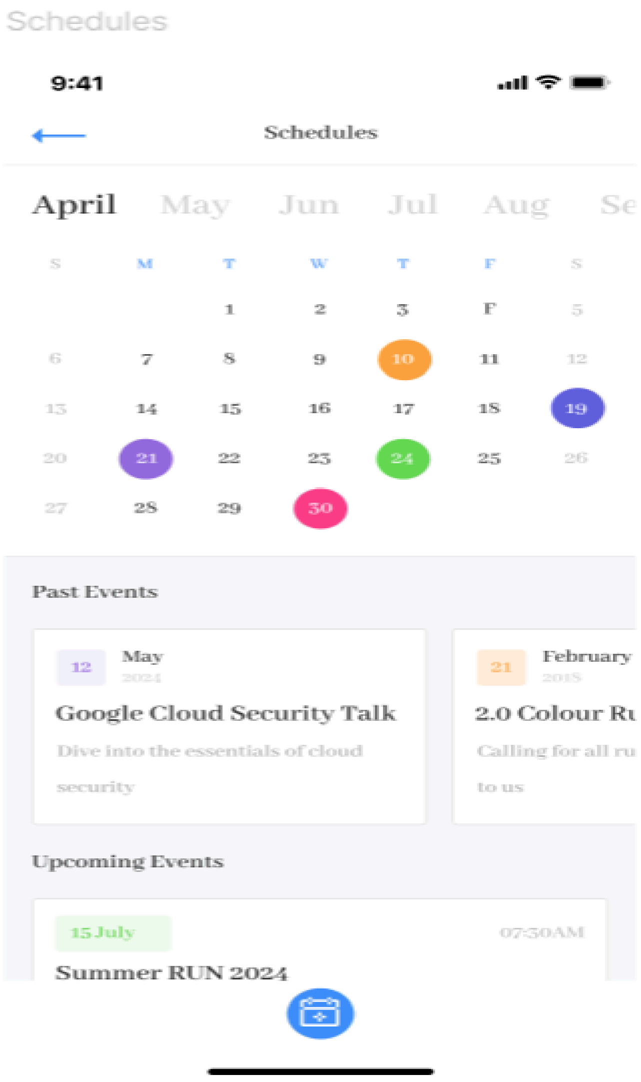
Tyler Perry

Sarah Douglas

Jennifer Banks

Edward Bell

7.15 Event Scheduling



Schedules

9:41

Schedules

April May Jun Jul Aug Se

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | F | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | | | |

Past Events

12 May 2024

Google Cloud Security Talk

Dive into the essentials of cloud security

21 February 2018

2.0 Colour Ru

Calling for all ru to us

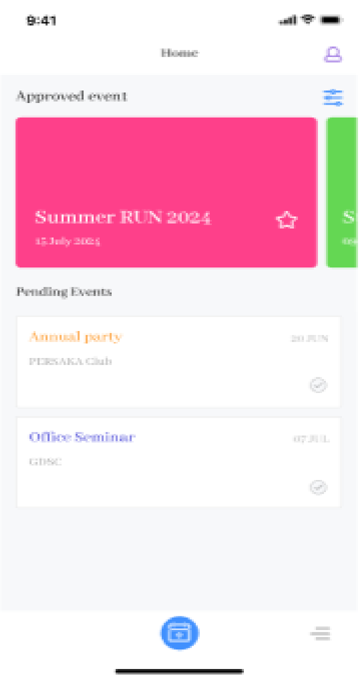
Upcoming Events

15 July 07:30AM

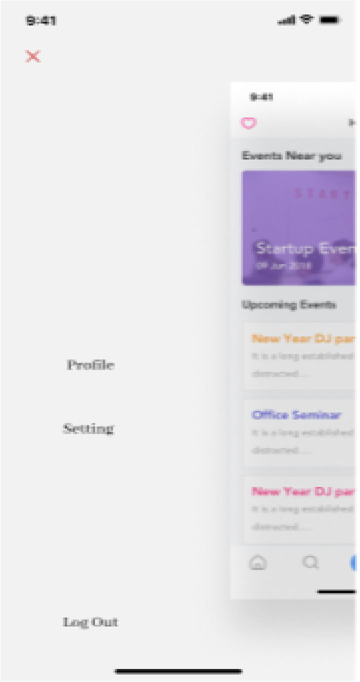
Summer RUN 2024

## 7.16 Administrator view (Home Page)

Home Administrator

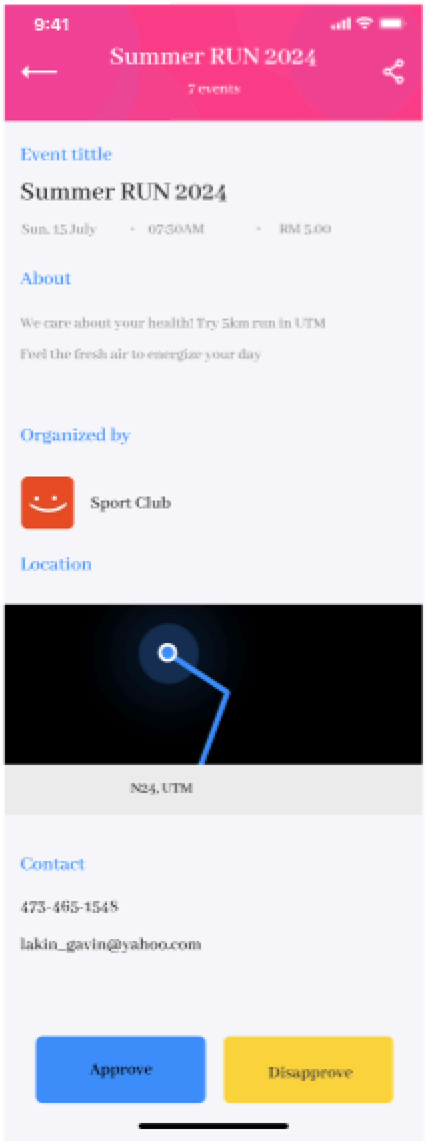


nav bar



### 7.17 Approve Event

details



9:41

Summer RUN 2024

7 events

Event tittle

Summer RUN 2024

Sun, 15 July · 07:30AM · RM 5.00

About

We care about your health! Try 5km run in UTM

Feel the fresh air to energize your day

Organized by

Sport Club

Location

N24, UTM

Contact

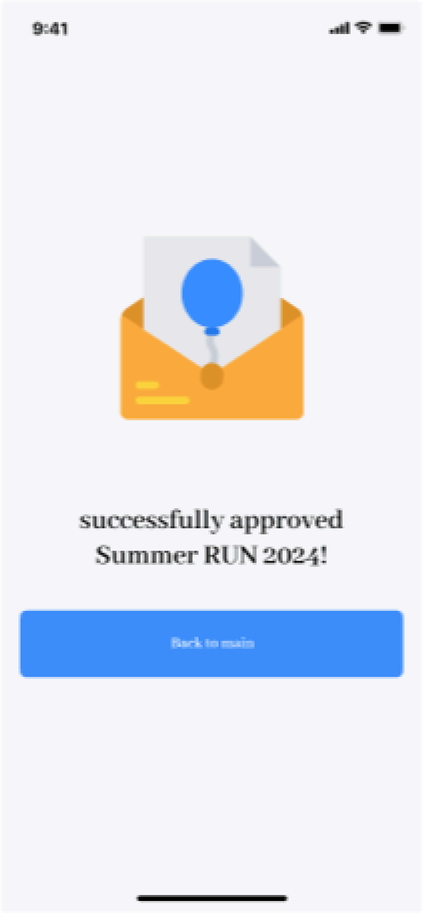
473-465-1548

lakin_gavin@yahoo.com

Approve

Disapprove

Succes register



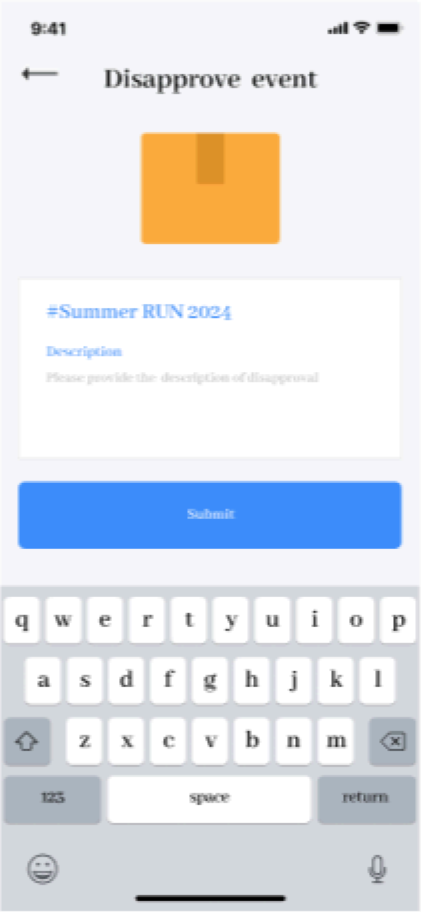
9:41

successfully approved
Summer RUN 2024!

Back to main

7.18 Disapprove Event

Disapprove Event



9:41

Disapprove event

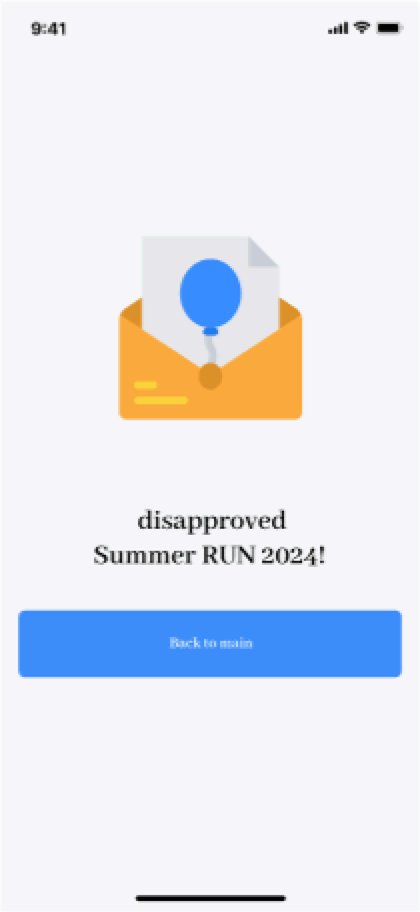
#Summer RUN 2024

Description

Please provide the description of disapproval

Submit

Succes register



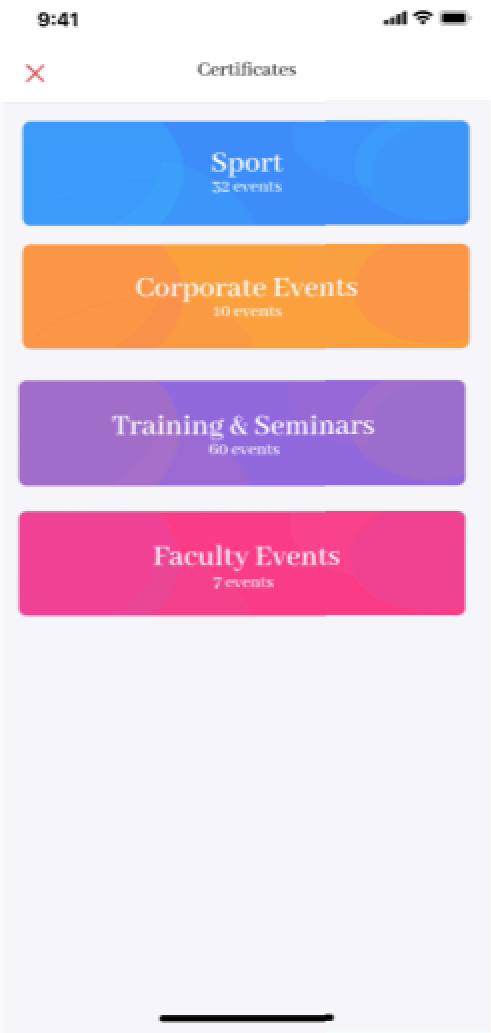
9:41

disapproved
Summer RUN 2024!

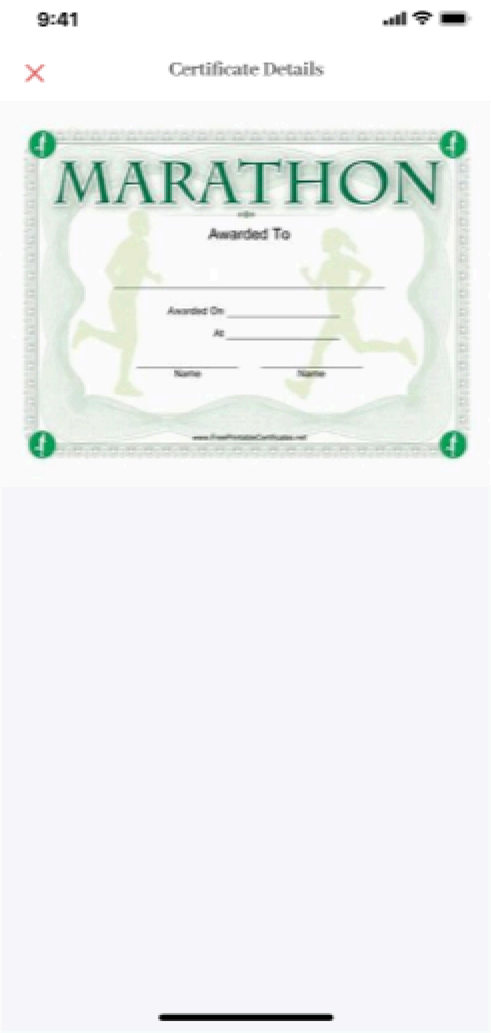
Back to main

7.19 View Participation Certificates list

Certificate List



Certificate Details



7.20 User Log Out

Log out

# 8. Summary

During this stage we Carefully Transformed the Conceptual ERD we made previously to Logical ERD. This process involved modifying the design to optimize it for a more efficient Implementation of ClubHub system. We removed non-relational elements and improved relationships to ensure the database fully complied with relational database principles.

After we Completed the Logical ERD we Modified the data dictionary to reflect the changes we applied, And now we have a complete picture about all the entities relationships and Attributes in our system, Thus we performed the Normalization from the first normal form until the Boyce Codd Normal form, This process aimed to eliminate redundancies and dependencies and to ensure the integrity of our data, The Normalization resulted in an Organized relations.

After Normalization we Developed  Relational database Schema to create a well-structured and optimized database design. This schema defines the logical organization of the Data by clearly outlining Tables, Attributes required in our project.

We Implemented Coding SQL DDL and DML statements based on Relational database Schema to create the tables and to validate the logical ERDs against the system's transactional requirements. After that we developed a User Interface design to depict the main functionalities of our system like Events Filtering by Student, Events scheduling by Organizer, And Event Approving by Administrator.

In Conclusion, this phase taught us how to transform conceptual ideas into structured and normalized database designs. By applying the course objectives to our Proposed System ClubHub which aims to Enhance Student Extracurricular Experiences in UTM and improve Clubs activities .