

# fitFlex

"Unlock your potential with our exclusive gym membership – where fitness meets community!

DONE BY

Arpita Roy 220101065  
Suhana Safir 220101089  
Sithy Ramzia 220101039

# About fitFlex



To provide an overview of the new membership feature in the gym application. This feature enables users to register new members, store their information, and manage their membership status. The report covers the functionalities, data management, and user interactions within the new membership module.

# Market Overview

The new membership feature in the gym application is designed to simplify the process of registering new members and managing their information. With a user-friendly interface, robust data management, and enhanced security measures, this feature ensures efficient and secure handling of member data. It enhances operational efficiency for gym administrators and contributes to a positive experience for gym members.





# Key Features

Welcome to the heart of fitFlex, a robust solution designed to streamline and enhance the management of gym memberships, class enrollments, scheduling, and payments. This report provides an in-depth look at the app's features, benefits, and overall impact on gym operations and member experience.



## User Interface for Registration

The new membership page includes a user-friendly interface that allows gym administrators to input member details efficiently.



## Data Management

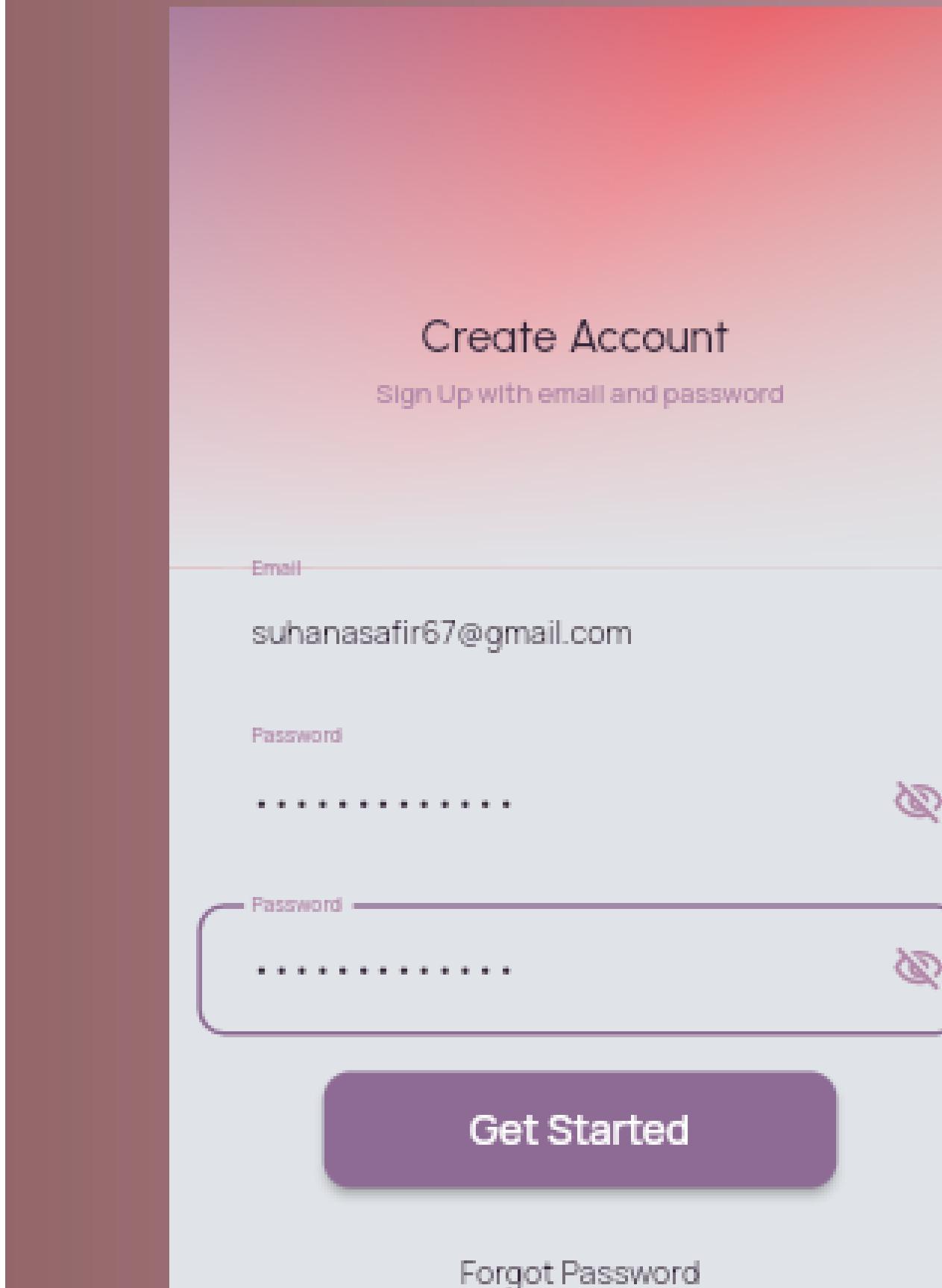
- The system ensures that all fields are validated before submission to prevent errors and incomplete data entries.



## Security and Privacy

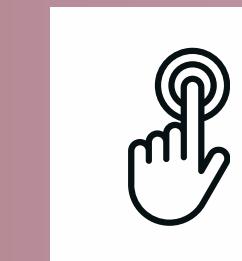
The application incorporates security measures such as data encryption and secure communication protocols to protect member information.

# Getting into the App



# User Benefits

As we delve deeper into the user benefits of Brocelle, you'll discover that our commitment goes beyond providing a mere payment solution.



## Convenience

Brocelle allows you to pay with a single tap, providing unparalleled convenience anytime, anywhere.



## Speed

Enjoy the efficiency of instant transfers, eliminating unnecessary delays in your financial transactions.



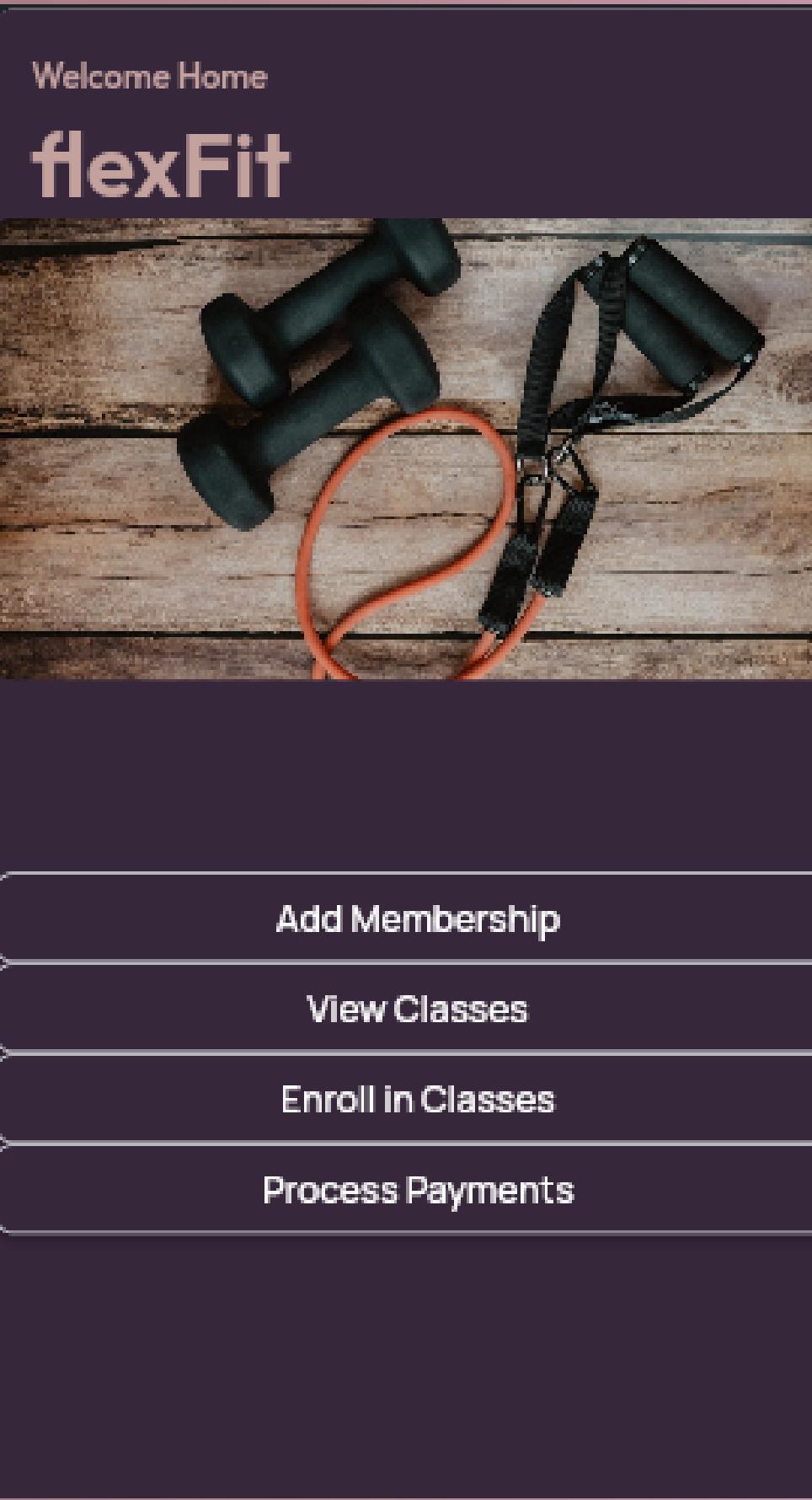
## Security

Trust in our robust security measures that ensure your financial data remains safeguarded.



# Getting into the App

Step 1: Login Page

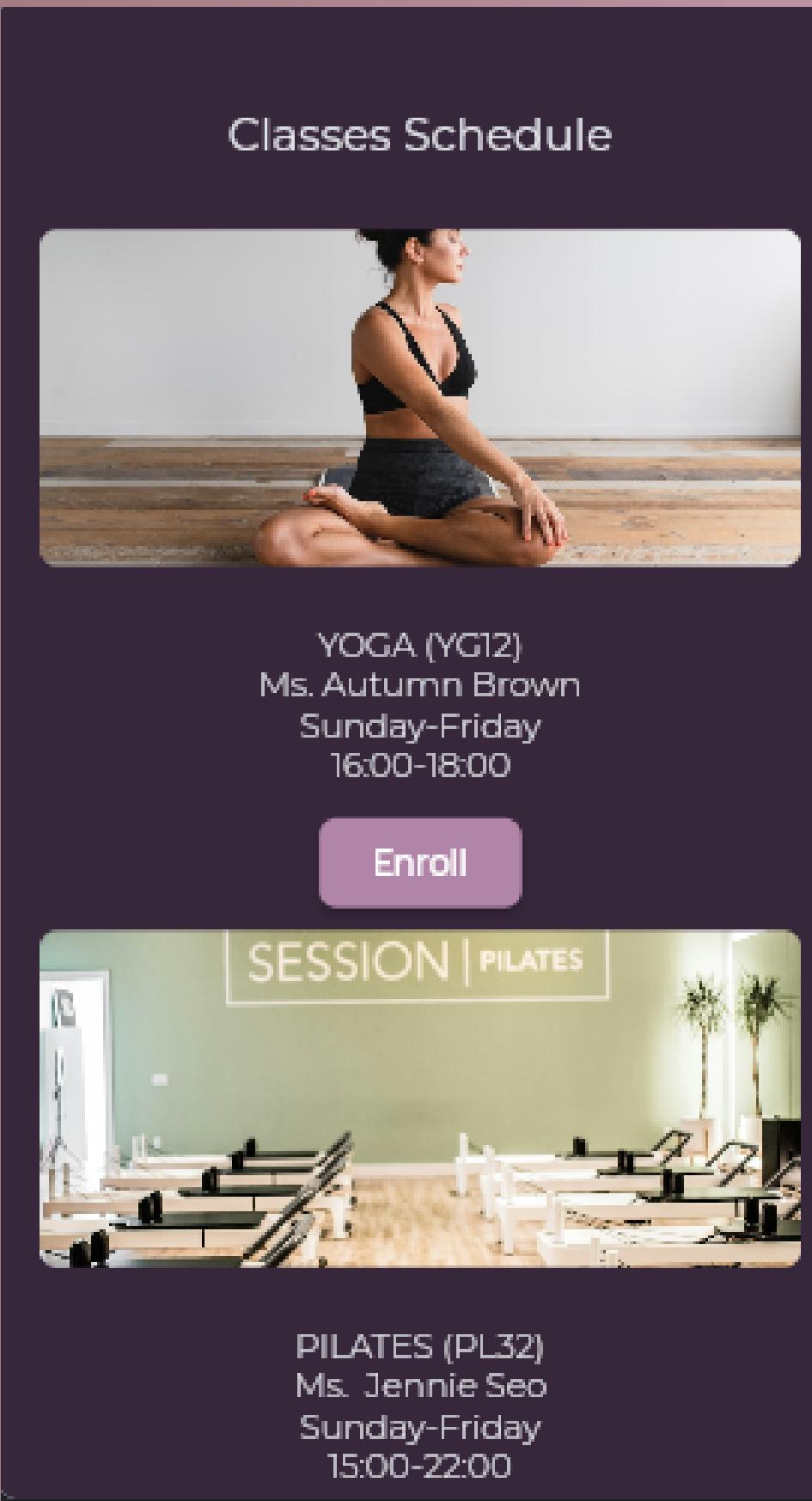


Step 2: Adding member details

The image shows the 'Add New Member Details' screen. It has five input fields with labels: 'Name', 'Mobile Number', 'Member ID', 'Membership Type', and 'Status'. Each field has a light purple placeholder text. Below the 'Membership Type' and 'Status' fields are dropdown menus with 'Please select...' as the default option. At the bottom right is a large, rounded rectangular button labeled 'Register'.

# Getting into the App

## Step 3: Classes Schedule



## Step 4: Class enrollment

The screenshot shows a dark-themed mobile application interface for enrolling in a class. At the top, the title "Enroll Class" is displayed in white. Below it are four input fields, each with a label and a corresponding text input box: "Enrollment ID", "Member ID", "Enrollment Date", and "CLASS ID". Under the "CLASS ID" field is a dropdown menu with the placeholder text "Please select...". At the bottom of the screen is a large, rounded rectangular button labeled "Register".

# Getting into the App

## Step 5 : Payment Process

PAYMENT PROCESS

Payment ID

Member ID

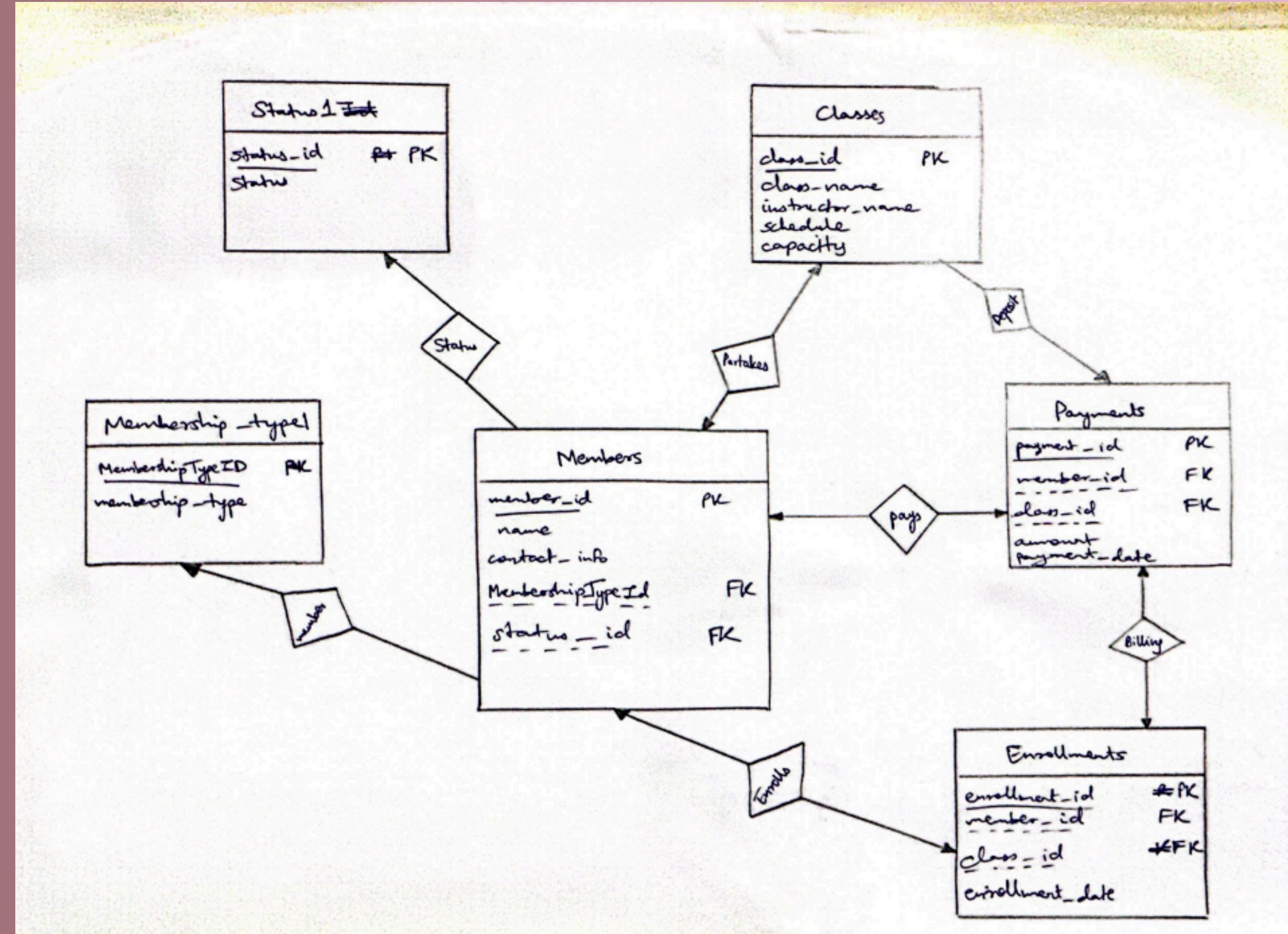
Amount

Credit Card Number

Payment Date

Payment Status

# ER Diagram



# ER Diagram Details

## ER Diagram Entities:

1. **Membership\_type** (**MembershipTypeId (PK)**, **membership\_type**)
2. **Status** (**status\_id (PK)**, **status**)
3. **Class** (**class\_id (PK)**, **class\_name**, **instructor\_name**, **schedule**, **capacity**)
4. **Members** (**member\_id (PK)**, **name**, **contact\_info**, **MembershipTypeId (FK)**, **status\_id (FK)**)
5. **Enrollments** (**enrollment\_id (PK)**, **member\_id (FK)**, **class\_id (FK)**, **enrollment\_date**)
6. **Payments** (**payment\_id (PK)**, **member\_id (FK)**, **class\_id (FK)**, **amount**, **payment\_date**)

## Relationships:

**One-to-Many:** **Membership\_type** has a **one-to-many relationship with Members** (a member can have one membership type, but a membership type can have many members).

**One-to-Many:** **Status** has a **one-to-many relationship with Members** (a member can have one status, but a status can have many members).

**Many-to-Many:** **Members** and **Class** have a **many-to-many relationship through Enrollments** (a member can enroll in many classes, and a class can have many members enrolled).

**One-to-Many:** **Members** has a **one-to-many relationship with Payments** (a member can have many payments, but a payment belongs to one member).

**One-to-Many:** **Class** has a **one-to-many relationship with Payments** (a class can have many payments, but a payment belongs to one class).

# Tables Created

```
CREATE TABLE Membership_type1 (
    MembershipTypeld INT PRIMARY KEY,
    membership_type VARCHAR(100)
);
```

```
CREATE TABLE Status1 (
    status_id INT PRIMARY KEY,
    status VARCHAR(100)
);
```

```
CREATE TABLE Class_id1(
    sno INT PRIMARY KEY,
    class_id VARCHAR(20) NOT NULL
);
```

```
CREATE TABLE Members (
    member_id INT PRIMARY KEY,
    name VARCHAR(100),
    contact_info VARCHAR(100),
    MembershipTypeld INT,
    status_id INT,
    FOREIGN KEY (MembershipTypeld) REFERENCES Membership_type1(MembershipTypeld),
    FOREIGN KEY (status_id) REFERENCES Status1(status_id)
);
```

# Tables Created

```
CREATE TABLE Classes (
    class_id VARCHAR(20) PRIMARY KEY,
    class_name VARCHAR(100),
    instructor_name VARCHAR(100),
    schedule VARCHAR(1000),
    capacity INT
);
```

```
CREATE TABLE Enrollments (
    enrollment_id INT PRIMARY KEY,
    member_id INT,
    class_id INT,
    enrollment_date DATE,
    FOREIGN KEY (member_id) REFERENCES Members(member_id),
    FOREIGN KEY (class_id) REFERENCES Class_id1(sno)
);
```

```
CREATE TABLE Payments (
    payment_id INT PRIMARY KEY,
    member_id INT,
    class_id INT,
    amount DECIMAL(10, 2),
    payment_date DATE,
    payment_status VARCHAR(20),
    FOREIGN KEY (member_id) REFERENCES Members(member_id),
    FOREIGN KEY (class_id) REFERENCES Class_id1(sno)
);
```