Student id 2122698

CIS108-6 - Data Modelling, Management and Governance assessment

MEMBERSHIP RECORD SYSTEM FOR SPORTS CLUB

EXECUTIVE SUMMARY

The Healthy Sports Club Management System project has demonstrated its effectiveness in meeting the club's operational needs. This comprehensive endeavor covered data modeling, system design, and successful implementation using “Microsoft SQL Server Management Studio”. The project's core functions, including member registration, class booking, and attendance tracking, are meticulously developed and rigorously tested. The chosen deployment strategy aligns with budget constraints and scalability requirements. Also, a commitment to ongoing security measures and maintenance activities ensures the protection of member data and the reliability of the system. Looking ahead, potential improvements may include the integration of payment processing and the enhancement of reporting features, all with the aim of continuously meeting the evolving demands of the club.

Table of Contents

Chapter 1: Introduction

1.1 Background of the study

A sports club serves as a communal institution where diverse individuals congregate to partake in a wide array of physical activities and sports. Membership in such clubs is typically motivated by a variety of personal objectives. Some members join to maintain and enhance their physical fitness levels due to the recognized health benefits of regular exercise. Others enlist to acquire new skills, form social connections, or purely for enjoyment. Within the precincts of sports clubs, trained instructors or coaches are typically on hand to provide guidance, impart knowledge, and oversee the activities of participants. These mentors serve as experts who impart expertise in playing sports or improving proficiency. Membership options offered by sports clubs often encompass various privileges, such as access to diverse facilities or preferential rates for instructional sessions. Consequently, sports clubs represent multifaceted environments designed to facilitate physical activity, instruction, and social interaction.

1.2 The necessity of New Management System

The requirement for a new management system arises from a set of compelling factors and needs. These factors underscore the essentiality of implementing an efficient management system capable of effectively addressing the challenges and demands confronted by contemporary organizations. In the following exposition, the imperative for adopting such a system is explicated.

1. Enhancement of Efficiency: In the current dynamic and rapidly evolving business milieu, the emphasis on augmenting efficiency is of paramount importance. A new management system can rationalize operational procedures, automate repetitive tasks, and enable data-driven decision-making processes. This augmented efficiency is likely to result in significant reductions in operational costs, thereby leading to improved resource allocation.

2. Management of Data: In light of the exponential surge in data volumes within contemporary organizations, proficient data management assumes critical significance. A new system can provide the requisite tools and functionalities for the efficient capture, storage, processing, and

analysis of substantial data volumes. Such proficiency is particularly pivotal in a data-centric realm where insights derived from data can confer a competitive edge.

3. Integration and Compatibility: Many existing systems in operation currently function in isolation, giving rise to data silos and inefficient operational processes. A contemporary management system can facilitate seamless integration with other systems and technologies, fostering compatibility and data sharing. This connectivity can foster improved interdepartmental collaboration and heighten overall organizational cohesion.

4. Scalability: As organizations expand and evolve, their requisites evolve correspondingly. A novel management system ought to be scalable, adept at accommodating increasing data loads, users, and escalating business intricacy. Scalability guarantees the enduring relevance and efficacy of the system, capable of adapting to evolving demands.

5. Elevated Customer Experience: The provision of an exceptional customer experience assumes strategic significance. A new management system can offer tools for the enhanced comprehension of customer behaviors, preferences, and feedback. This comprehension empowers organizations to tailor products and services to customer needs, thereby heightening satisfaction and loyalty.

1.3 Scope and Objective of the new system

The scope of the new computer system intended for deployment at the Healthy Sports Club encompasses several vital aspects. These include the management of both club members and staff, along with the facilitation of facility and class bookings, instructor management, and the implementation of stringent data security protocols. The overarching objective is to simplify club operations, enhance member experiences, optimize the allocation of resources, foster scalability, and ensure adherence to regulatory and industry standards.

The objectives of the new computer system can be articulated as follows:

- To have club operations and administrative processes streamlined.
- To provide members with user-friendly tools to enhance their experience.

- To ensure that the efficient allocation of club resources is improved.
- To guarantee the security of sensitive data.
- To enable scalability and the accommodation of growth.
- To furnish data-driven insights for the purpose of informed decision-making.
- To ensure that compliance with industry regulations and club policies is supported.

1.4 Design Strategy and methods

The fundamental design approaches refer to the core principles guiding the structure and functionality of a system. In the context of designing a computer system for the Healthy Sports Club, several crucial design approaches come into play:

1. ***User-Centered Design:*** Prioritizing the needs and preferences of users, such as club members and staff, to ensure an interface that is easy to use and understand.
2. ***Modularity:*** Designing the system in separate, manageable components allows for flexibility, easier maintenance, and scalability as each part can be developed independently.
3. ***Scalability:*** The system should be capable of growing and adapting to the club's evolving needs, accommodating increased data and users without significant reconfiguration.
4. ***Integration:*** Emphasizing integration enables smooth data exchange with other technologies and systems, enhancing the overall operational efficiency and data accuracy.

Regarding the system utilization, the strategy and methods is focus on ensuring accessibility, efficiency, and effectiveness:

1. ***User Training:*** Developing a comprehensive training program to ensure that the club members and staff can effectively utilize the system's features.
2. ***Continuous Enhancement:*** Establishing a framework for ongoing system improvement based on user feedback and evolving requirements.

3. ***Routine Maintenance:*** Implementing regular maintenance protocols to address issues, enhance performance, and ensure data security.

4. ***User Support:*** Offer accessible and responsive support channels to help the users with system related concerns or questions.

Chapter 2: Requirement Analysis

2.1 System Requirements

The main goal of the system is to effectively and simply manage the Healthy Sports Club's operations. This includes making things easier for the club members and staff, ensuring data security, and complying with rules and regulations. Below mention the key things the system needs to do:

Managing Members: For creating, updating, and deleting the member profiles in the system. It contains their personal information, the type of membership they have, and their previous booking history.

Handling Staff: It can help to manage the staff information, their roles, and contact details.

Booking Facilities and Classes: Club members are able to book the club facilities and their classes using the system (Song *et al.* 2023). The system can also check if there are any restrictions or if the facility or class is available.

Instructor Management: The system can support managing class instructors, and allowing the club to add, change, or remove instructor details and assignments.

Security: There are strong security measures in place to protect sensitive information, like the member and staff data.

Scalability: The system is flexible enough to grow with the club, managing more data and users without needing important or major changes.

Integration: It can work well with other systems and technologies to make things data accurate and more efficient.

Final Statement of What the System Will Do: The system acts as a central hub for managing the Healthy Sports Club. It helps to handle the member profiles, staff data, facility bookings, class bookings, and everything is secure and follows the rules.

Identified Roles:

Club Member: The club members who are part of the Healthy Sports Club.

Class Instructor: These actors are the staff members who teach different classes.

Admin Staff: These are the administrative staff who can handle the overall system.

Use-case Diagram

The process of collecting the actual requirements for "Healthy Sports Club Management System'" involves understanding the needs and the expectations of various stakeholders, such as, club members, class instructors, and administrative staff (Namdeo and Suman, 2021).

2.2 The Process of Gathering Requirements

Conducting Stakeholder Interviews: Interviews are conducted with the club members, class instructors, and the administrative staff. This allows direct communication with the key stakeholders, enabling them to express their needs, preferences, and areas of concern regarding the system. For example, club members can stress the importance of an easy-to-use system for the booking classes.

Distributing Feedback Surveys: Surveys are distributed to the staff and members to gather structured feedback (Chen, 2022). For instance, a survey question may inquire about the preferred method of accessing the class schedules.

Observing Current Processes: The current club operations are observed, including how members book classes and how instructors manage classes (Alabi *et al.* 2021). This provides valuable insights into existing processes and areas that require improvement. For instance, long queues during class registration may suggest the need for an online booking system.

Reviewing Existing Documentation: Existing club policies, procedures, and documentation are examined to uncover requirements related to compliance and operational protocols. For example, documentation may specify that class instructors must maintain attendance records.

Creating Prototypes and Mockups: Prototypes or mockups of the system's user interface and functionalities are developed (Cheng *et al.* 2019). These visual representations help stakeholders visualize the proposed system, and their feedback on these visuals can clarify requirements and expectations.

Iterative Feedback Gathering: Collecting feedback from stakeholders is an ongoing and iterative process (Čerešňák and Kvet, 2019). It involves multiple rounds of discussions, demonstrations, and revisions to ensure that the system aligns with their requirements.

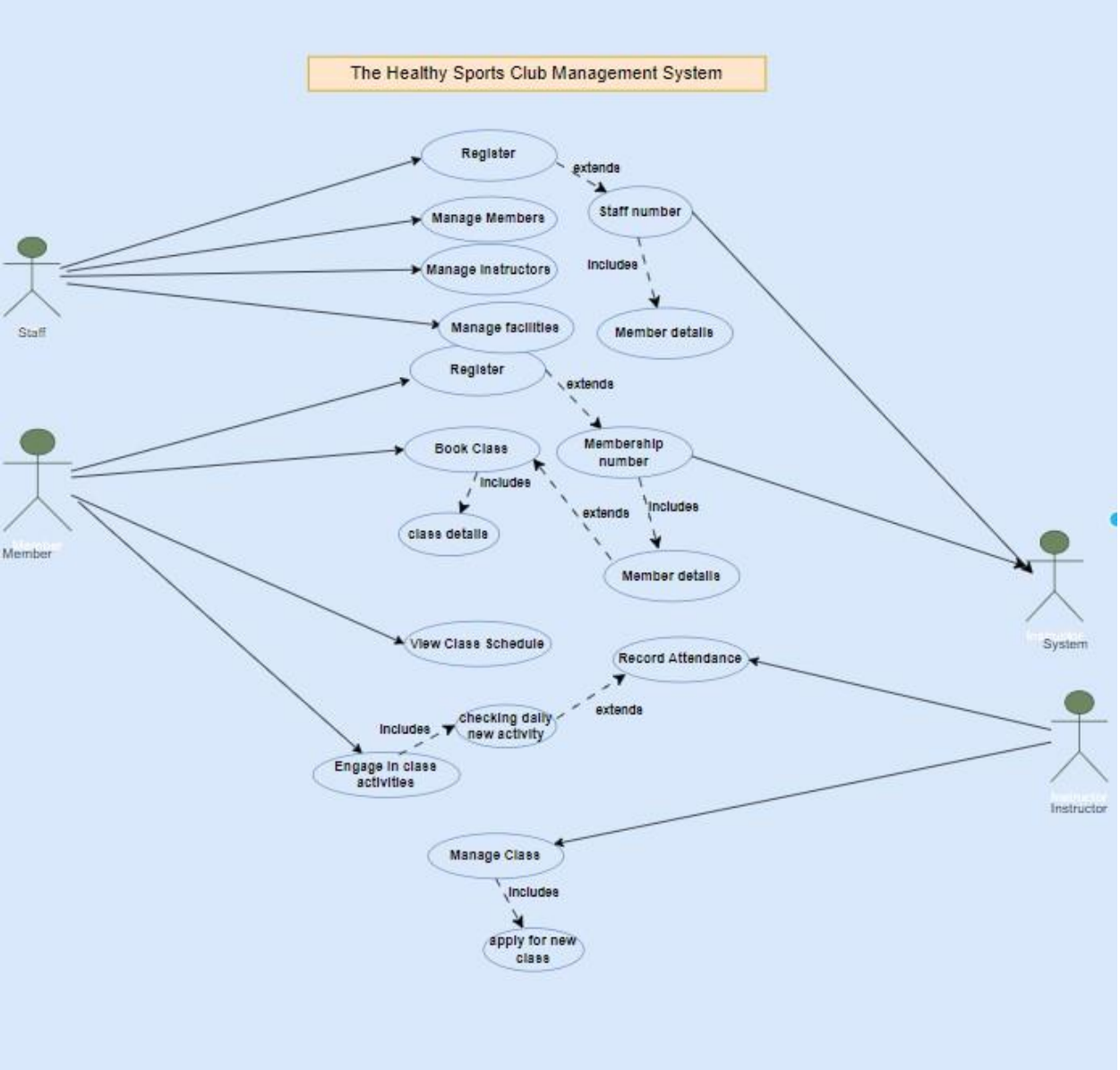


Figure 1: Use-case Diagram

(Source: Draw.io)

Key Actors: Club Member, Class Instructor, Admin Staff

Use-Cases:

1. ***Booking Classes:*** The club member can interact with this use-case to reserve a class.
2. ***Viewing Class Schedules:*** The club member can check the schedule of available classes.
3. ***Managing Classes:*** The class instructor can handle class details, including creating, updating, or deleting them.
4. ***Recording Attendance:*** Attendance for class members can be marked by the class instructor.
5. ***Managing Instructors:*** Instructor details, such as creating, updating, or deleting profiles, can be managed by admin staff.
6. ***Managing Members:*** Member details, including creating, updating, or deleting profiles, can be managed by admin staff.

This use-case diagram illustrates interactions between key actors and the system (Byun, 2021). It focuses on class bookings and management, a significant aspect of the Healthy Sports Club Management System, while acknowledging that additional use-cases and actors may exist in the complete system.

Activity Diagram

1. Initiation: The process commences when the "Book Class" use-case is triggered by a Club Member.

2. Member Authentication: The system checks and verifies the identity of the member to ensure they are a registered user.

- ***Positive Outcome:*** If the member's identity is successfully confirmed, the process proceeds.
- ***Negative Outcome:*** In case of authentication failure, the process concludes with an error message.

3. Class Availability: The Club Member selects a class from the list of available classes presented by the system.

- ***Availability Check:*** The system examines whether there are open slots in the selected class.
- ***Availability Confirmed:*** If slots are available, the process advances.
- ***No Availability:*** In case the class is fully booked, the process concludes, informing the user of unavailability.

4. Booking Action: The Club Member confirms their intention to book the class by triggering the "Book Class" action.

- ***Booking Success:*** If the booking process proceeds smoothly, the process advances.
- ***Booking Issue:*** If any problems arise during booking, the process concludes with an error message.

5. Confirmation Booking: The system validates the booking and records it in the database.

- ***Confirmation Success:*** If confirmation is successful, the process proceeds.
- ***Confirmation Issue:*** In case of confirmation problems, the process ends with an error message.

6. Receipt Generation: A booking receipt is generated by the system and displayed to the Club Member, serving as confirmation of their class reservation (Zhang, 2022).

8. End: The process concludes, and the Club Member has effectively booked a class.

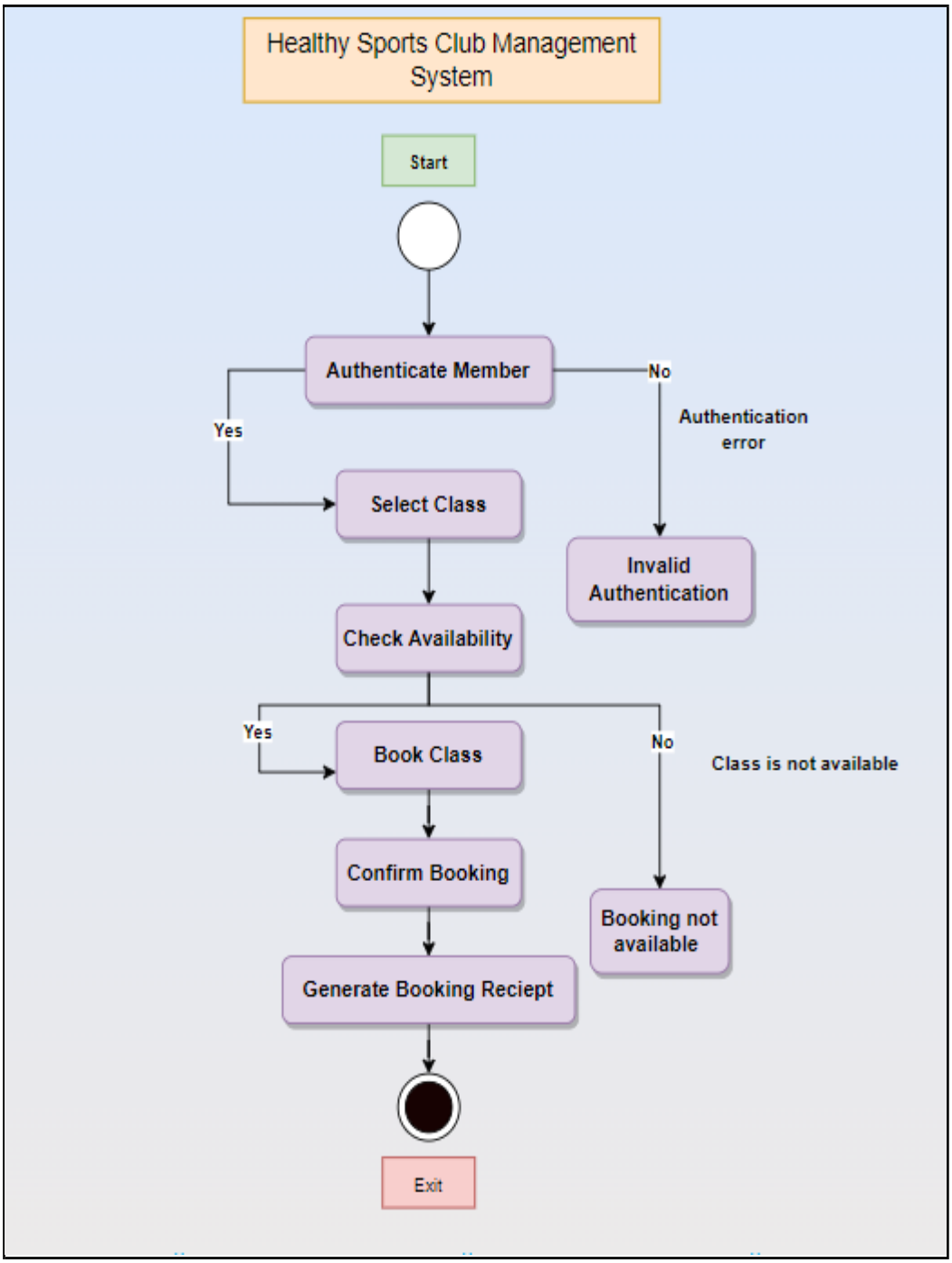


Figure 2: Activity Diagram

(Source: Draw.io)

Chapter 3: System Design

3.1 Data Modelling

1. Members Table:

Information regarding club members is stored in this table. It includes unique "MemberID", "MembershipType" (e.g., Gold, Silver, Bronze), "FirstName", "LastName", "Address", "PhoneNumber", "Email", "DateOfBirth", and "MedicalConditions".

2. Staff Table:

This Table contains the data related to club staff members such as, unique "StaffID", "FirstName", "LastName", "Role", and "ContactNumber" (Qin and Feng 2021). This table is essential to manage the staff related information.

3. Facilities Table:

The Facilities Table records of various club facilities that can be reserved. This table includes "FacilityID", "FacilityName" (e.g., Gym, Yoga Hall). This table helps to manage facility availability and allocation.

4. FacilityBookings Table:

This table records bookings made by the club members for specific facilities. This table stores "BookingID", "MemberID", "facilityID", "BookingDate", "StartTime", and "EndTime".

5. Classes Table:

The Classes Table contains the information about various classes offered by the club. It stores the unique values as "ClassID", "InstructorID", "StaffID", "ClassName", "DayOfWeek", "TimeOfDay", and "MaxParticipants".

6. ClassBookings Table:

This ClassBooking table tracks the registrations made by club members for their specific classes. It contains the "BookingID", "MemberID", "ClassID" of the chosen classes, and "BookingDate".

7. Instructors Table:

The Instructors Table stores the data regarding the class instructors employed by the club (CHAITHRA, 2021). This table stores "InstructorID", "FirstName", and "LastName". This table serves as a repository for instructor details, also in class management and the assignments. These tables collectively form the core of the data system for the Healthy Sports Club, facilitating the efficient management of members, staff, facilities, classes, and bookings, while also ensuring the data integrity and the accuracy.

3.2 Entity Relationship Diagram

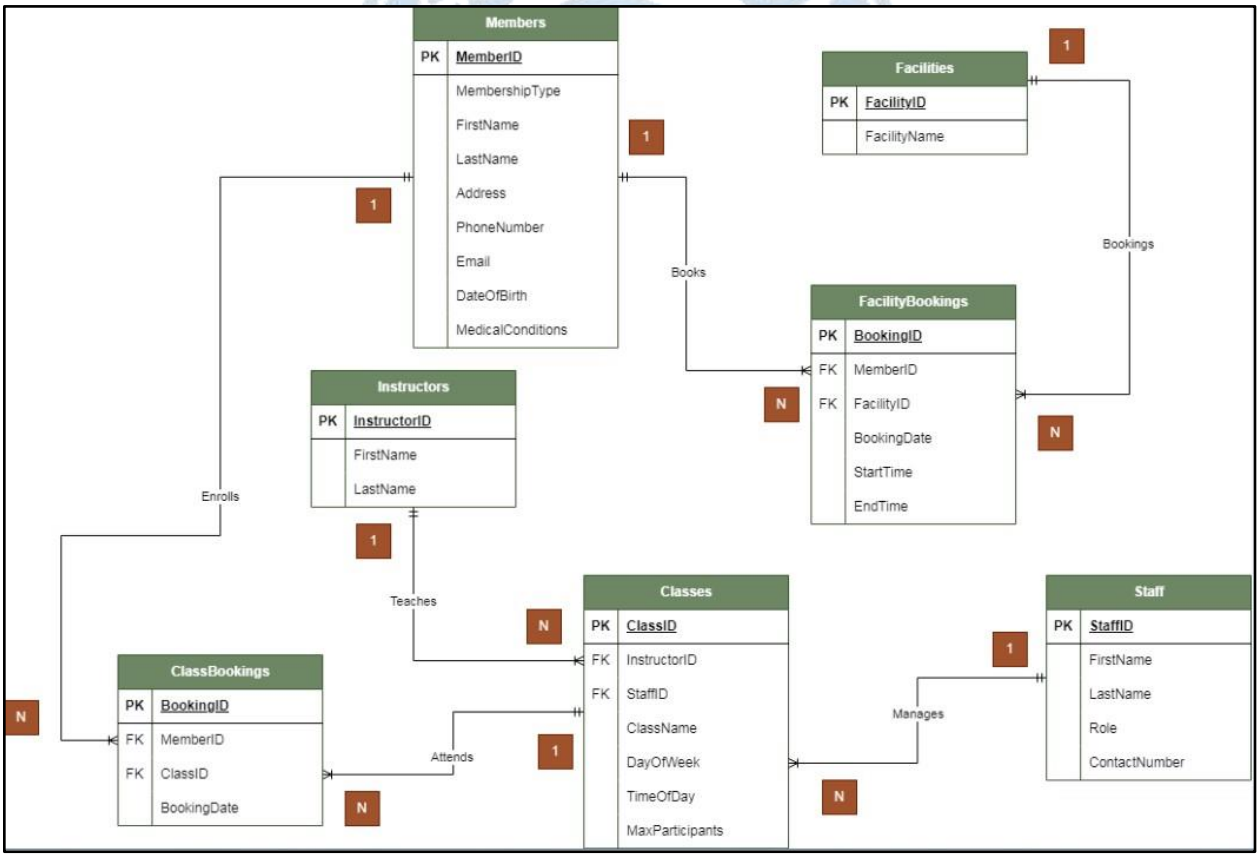


Figure 3: Entity Relationship Diagram (ERD)

(Source: Draw.io)

In the above image it represents the Entity Relationship Diagram of the Healthy Sports Club Management System (Tao *et al.* 2022). In the section below it displays the relationship and cardinality between these tables.

Entities Connection	Relationship	Cardinality
Members Table and FacilityBookings Table	Members have the capability to make bookings for club facilities	Many Members (1) can create Many Facility Bookings (N)
Members Table and ClassBookings Table	Members have the privilege to reserve spots in club classes	Many Members (1) can reserve Many Classes (N)
Classes Table and ClassBookings Table	Classes can be reserved by members	Many Classes (1) can be reserved by Many Members (N)
Facilities Table and FacilityBookings Table	Facilities can be booked by members	Many Facilities (1) can be booked by Many Members (N)
Instructors Table and Classes Table	Instructors are responsible for teaching classes	Many Instructors (1) instruct Many Classes (N)

Staff Table and Classes Table	Staff members are involved in managing classes	Many Staff (1) are engaged in managing Many Classes (N)

Chapter 4: System Implementation

In the process of bringing our system design to life, I selected MySQL, a Relational Database Management System (RDBMS) (Kuo *et al.* 2022). I'll outline the key steps in this implementation process, along with insights into why MySQL is the preferred option and the advantages and limitations associated with it.

1. Consideration of Other Database Options:

Alternative databases, such as MongoDB and PostgreSQL, are taken into account (Grasso *et al.* 2019). However, after careful evaluation, MySQL emerged as the best fit due to its reliability and suitability for our requirements.

2. Advantages and Limitations of MySQL:

Benefits:

- MySQL ensures robust data integrity and adheres to the principles of ACID, making it reliable for our data management needs.
- It excels in handling complex queries and transactions, which are vital for managing member data, reservations, and schedules.
- MySQL's open-source nature makes it cost-effective, a crucial factor for our local business, the Healthy Sports Club.

Constraints:

- While MySQL is highly suitable for numerous applications, it might face performance challenges with exceptionally large datasets or intricate relationships.
- In scenarios with high user concurrency, some optimization may be required for optimal performance.

3. Execution of Implementation:

The actual implementation involved the utilization of Microsoft SQL Management Studio to execute a sequence of SQL commands. These commands are responsible for creating the essential database tables and populating them with sample data for the purpose of testing (Epstein *et al.* 2021). The SQL statements covered the creation of tables for Members, Staff, Facilities, Facility Bookings, Classes, Class Bookings, and Instructors.

The implementation ensures that the database structure closely aligns with our system design, facilitating efficient data management and retrieval within the Healthy Sports Club Management System.

Table Creation

Members Table

```
CREATE TABLE Members (
    MemberID INT PRIMARY KEY,
    MembershipType VARCHAR(50),
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Address VARCHAR(100),
    PhoneNumber VARCHAR(20),
    Email VARCHAR(100),
    DateOfBirth DATE,
    MedicalConditions VARCHAR(255)
);
```

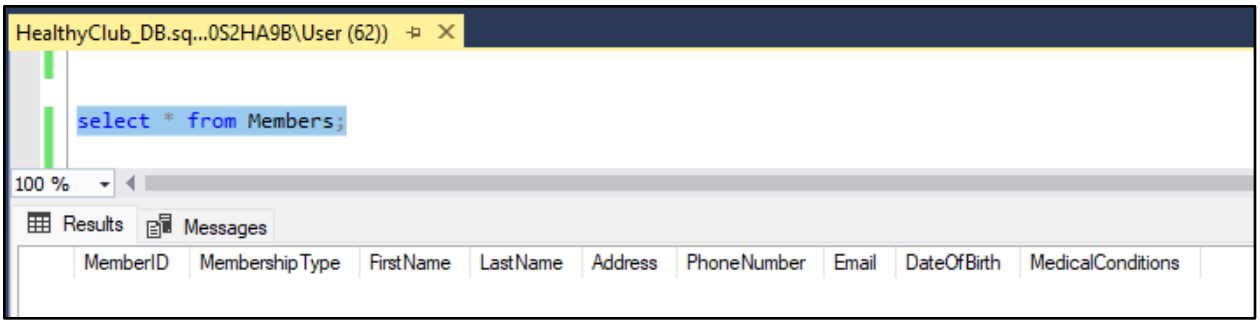


Figure 4: Create Table of Members

(Source: Microsoft SQL Server Management Studio)

Staff Table

```
CREATE TABLE Staff (
   StaffID INT PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Role VARCHAR(50),
   ContactNumber VARCHAR(20)
);
```

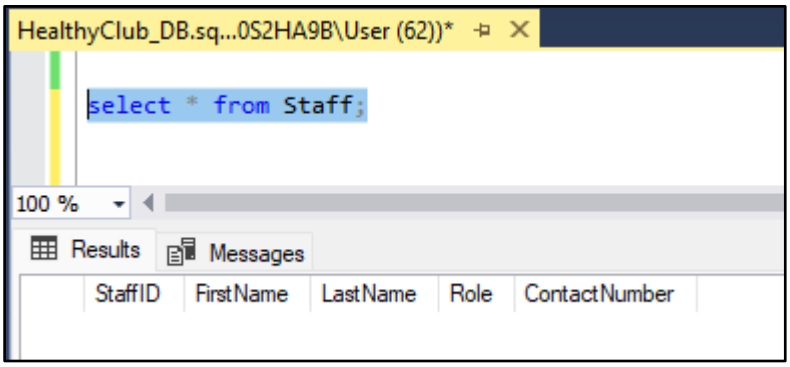


Figure 5: Create Table of Staff

(Source: Microsoft SQL Server Management Studio)

Facilities Table

```
CREATE TABLE Facilities (
    FacilityID INT PRIMARY KEY,
    FacilityName VARCHAR(50)
);
```

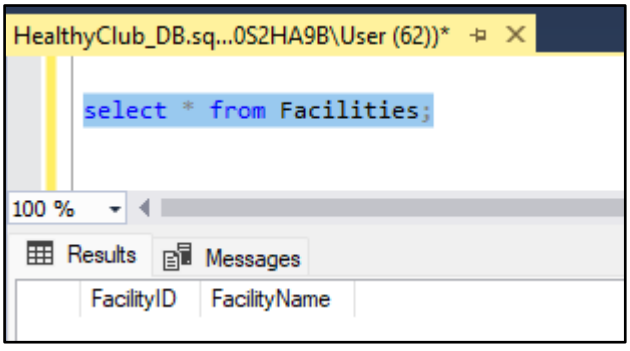


Figure 6: Create Table of Facilities

(Source: Microsoft SQL Server Management Studio)

FacilityBookings Table

```
CREATE TABLE FacilityBookings (
    BookingID INT PRIMARY KEY,
    MemberID INT,
    FacilityID INT,
    BookingDate DATE,
    StartTime TIME,
    EndTime TIME,
    CONSTRAINT FK_Member_FacilityBooking FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    CONSTRAINT FK_Facility_FacilityBooking FOREIGN KEY (FacilityID) REFERENCES Facilities(FacilityID)
);
```

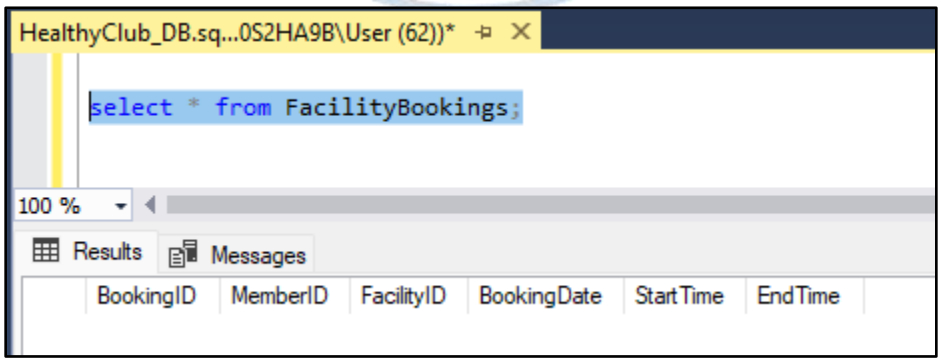


Figure 7: Create Table of FacilityBookings

(Source: Microsoft SQL Server Management Studio)

ClassBookings Table

```
CREATE TABLE ClassBookings (
    BookingID INT PRIMARY KEY,
    MemberID INT,
    ClassID INT,
    BookingDate DATE,
    CONSTRAINT FK_Member_ClassBooking FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    CONSTRAINT FK_Class_ClassBooking FOREIGN KEY (ClassID) REFERENCES Classes(ClassID)
);
```

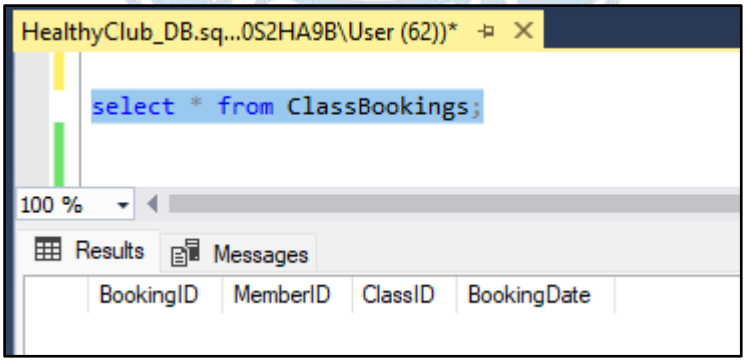


Figure 8: Create Table of ClassBookings

(Source: Microsoft SQL Server Management Studio)

Classes Table

```
CREATE TABLE Classes (
    ClassID INT PRIMARY KEY,
    ClassName VARCHAR(50),
    InstructorID INT,
    DayOfWeek VARCHAR(15),
    TimeOfDay TIME,
    MaxParticipants INT,
    CONSTRAINT FK_Instructor_Class FOREIGN KEY (InstructorID) REFERENCES Instructors(InstructorID)
);
```

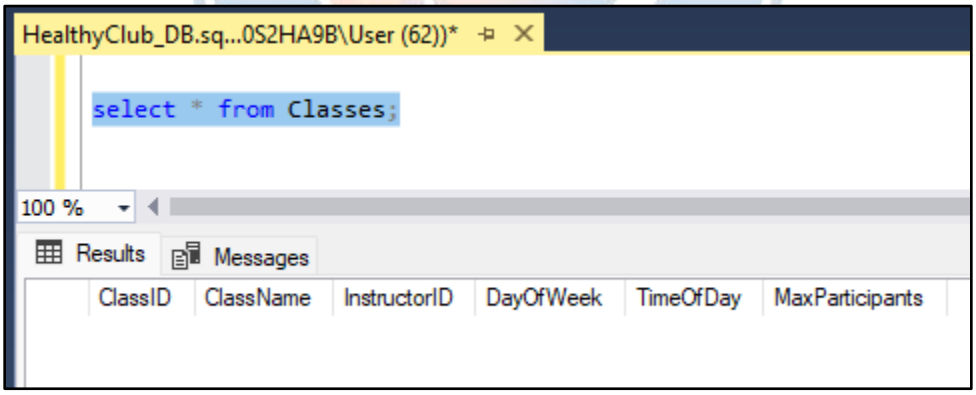


Figure 9: Create Table of Classes

(Source: Microsoft SQL Server Management Studio)

Instructors Table

```
CREATE TABLE Instructors (
    InstructorID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50)
);
```

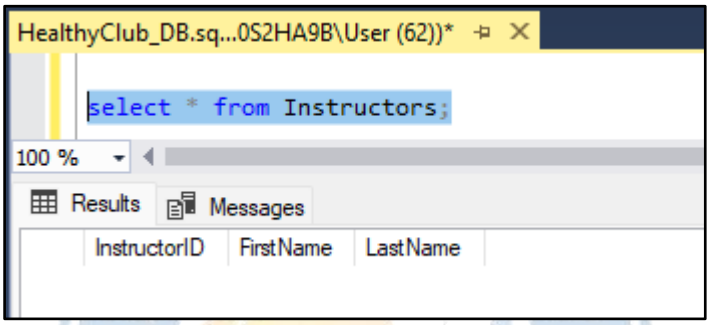


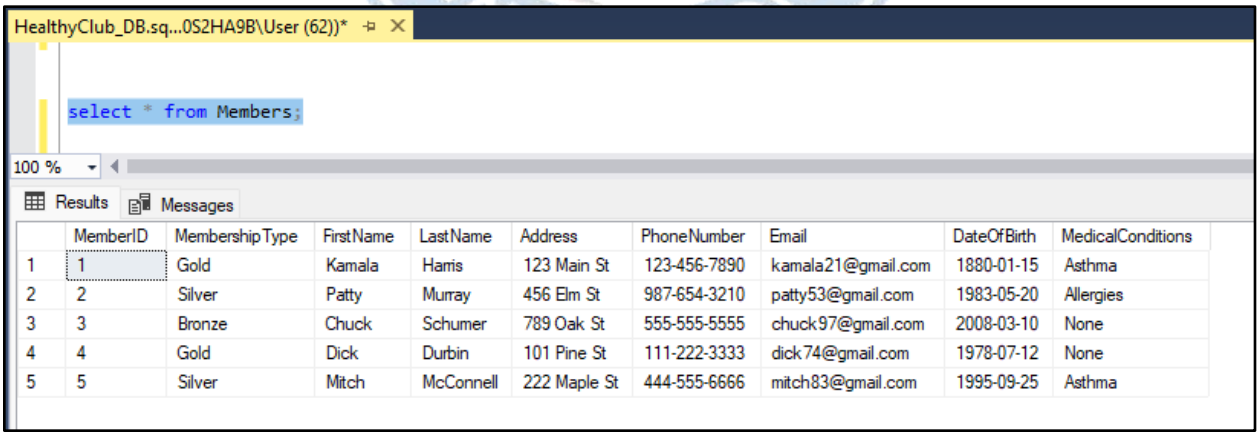
Figure 10: Create Table of Instructors

(Source: Microsoft SQL Server Management Studio)

Data Insertion

Members Table

```
INSERT INTO Members (MemberID, MembershipType, FirstName, LastName, Address,
PhoneNumber, Email, DateOfBirth, MedicalConditions)

VALUES

   (1, 'Gold', 'Kamala', 'Harris', '123 Main St', '123-456-7890', 'kamala21@gmail.com', '1880-
01-15', 'Asthma'),

   (2, 'Silver', 'Patty', 'Murray', '456 Elm St', '987-654-3210', 'patty53@gmail.com', '1983-05-
20', 'Allergies'),

   (3, 'Bronze', 'Chuck', 'Schumer', '789 Oak St', '555-555-5555', 'chuck97@gmail.com', '2008-
03-10', 'None'),

   (4, 'Gold', 'Dick', 'Durbin', '101 Pine St', '111-222-3333', 'dick74@gmail.com', '1978-07-12',
'None'),

   (5, 'Silver', 'Mitch', 'McConnell', '222 Maple St', '444-555-6666', 'mitch83@gmail.com',
'1995-09-25', 'Asthma');
```



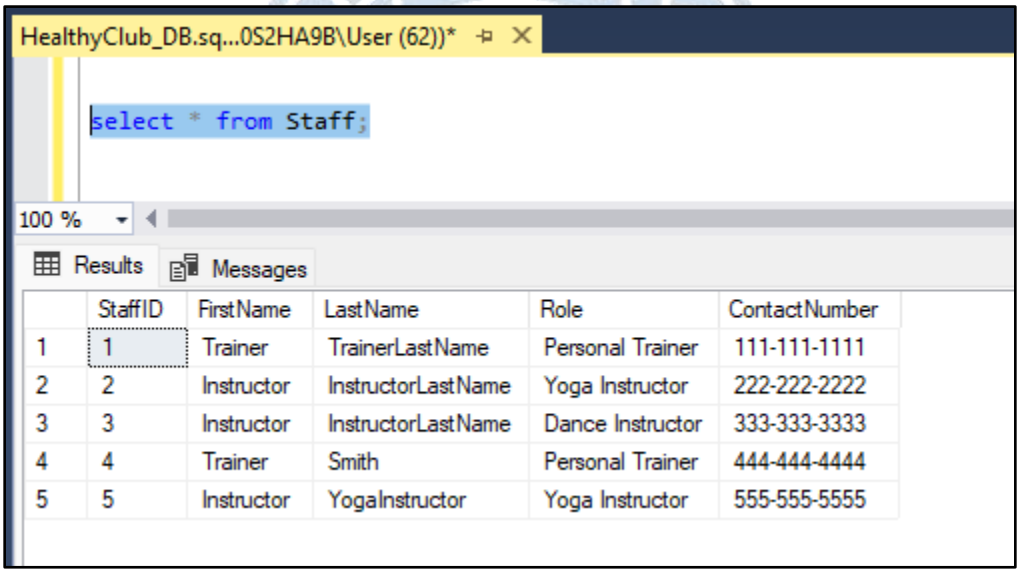
	MemberID	MembershipType	FirstName	LastName	Address	PhoneNumber	Email	DateOfBirth	MedicalConditions
1	1	Gold	Kamala	Harris	123 Main St	123-456-7890	kamala21@gmail.com	1880-01-15	Asthma
2	2	Silver	Patty	Murray	456 Elm St	987-654-3210	patty53@gmail.com	1983-05-20	Allergies
3	3	Bronze	Chuck	Schumer	789 Oak St	555-555-5555	chuck97@gmail.com	2008-03-10	None
4	4	Gold	Dick	Durbin	101 Pine St	111-222-3333	dick74@gmail.com	1978-07-12	None
5	5	Silver	Mitch	McConnell	222 Maple St	444-555-6666	mitch83@gmail.com	1995-09-25	Asthma

Figure 11: Insert data in Members Table

(Source: Microsoft SQL Server Management Studio)

Staff Table

```
INSERT INTO Staff (StaffID, FirstName, LastName, Role, ContactNumber)
VALUES
    (1, 'Trainer', 'TrainerLastName', 'Personal Trainer', '111-111-1111'),
    (2, 'Instructor', 'InstructorLastName', 'Yoga Instructor', '222-222-2222'),
    (3, 'Instructor', 'InstructorLastName', 'Dance Instructor', '333-333-3333'),
    (4, 'Trainer', 'Smith', 'Personal Trainer', '444-444-4444'),
    (5, 'Instructor', 'YogaInstructor', 'Yoga Instructor', '555-555-5555');
```



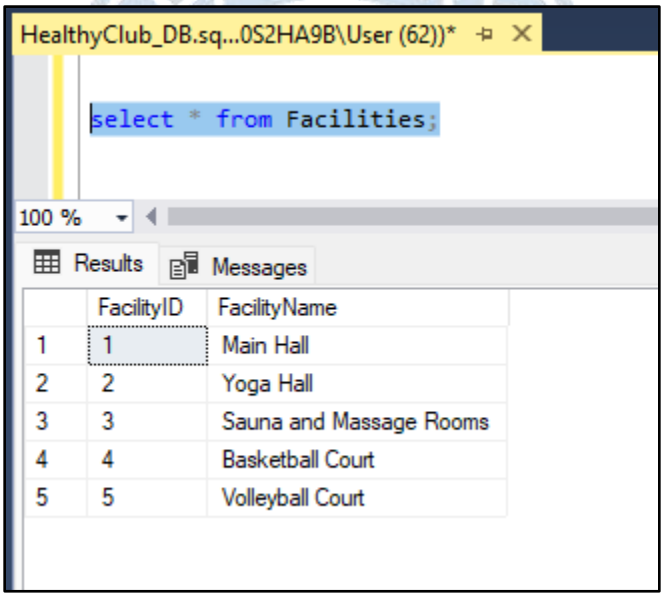
	StaffID	FirstName	LastName	Role	ContactNumber
1	1	Trainer	TrainerLastName	Personal Trainer	111-111-1111
2	2	Instructor	InstructorLastName	Yoga Instructor	222-222-2222
3	3	Instructor	InstructorLastName	Dance Instructor	333-333-3333
4	4	Trainer	Smith	Personal Trainer	444-444-4444
5	5	Instructor	YogaInstructor	Yoga Instructor	555-555-5555

Figure 12: Insert data in Staff Table

(Source: Microsoft SQL Server Management Studio)

Facilities Table

```
INSERT INTO Facilities (FacilityID, FacilityName)

VALUES

    (1, 'Main Hall'),

    (2, 'Yoga Hall'),

    (3, 'Sauna and Massage Rooms'),

    (4, 'Basketball Court'),

    (5, 'Volleyball Court');
```



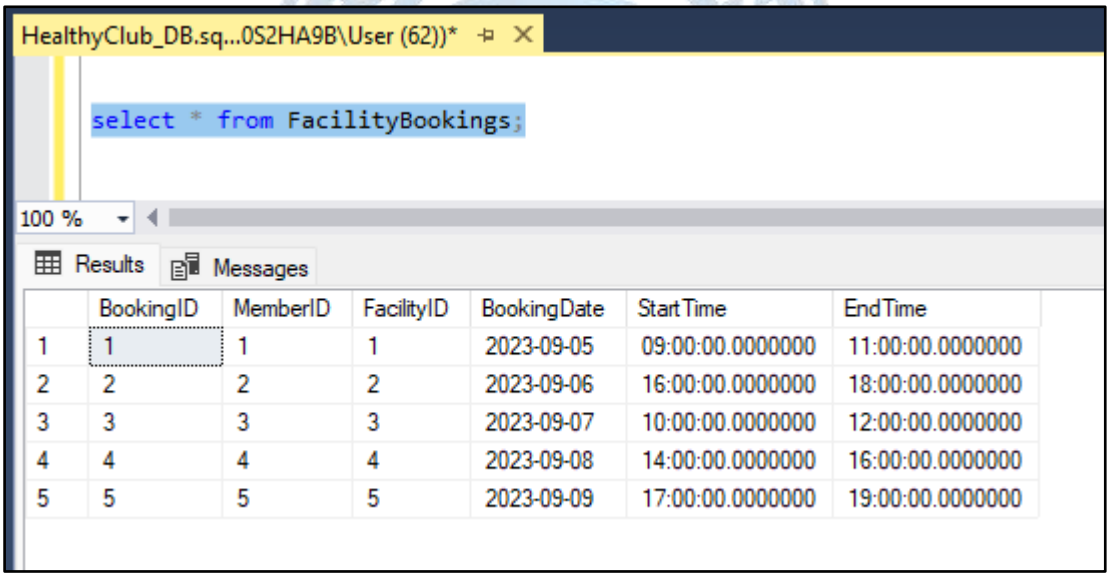
	FacilityID	FacilityName
1	1	Main Hall
2	2	Yoga Hall
3	3	Sauna and Massage Rooms
4	4	Basketball Court
5	5	Volleyball Court

Figure 13: Insert data in Facilities Table

(Source: Microsoft SQL Server Management Studio)

FacilityBookings Table

```
INSERT INTO FacilityBookings (BookingID, MemberID, FacilityID, BookingDate,
StartTime, EndTime)

VALUES

    (1, 1, 1, '2023-09-05', '09:00:00', '11:00:00'),

    (2, 2, 2, '2023-09-06', '16:00:00', '18:00:00'),

    (3, 3, 3, '2023-09-07', '10:00:00', '12:00:00'),

    (4, 4, 4, '2023-09-08', '14:00:00', '16:00:00'),

    (5, 5, 5, '2023-09-09', '17:00:00', '19:00:00');
```



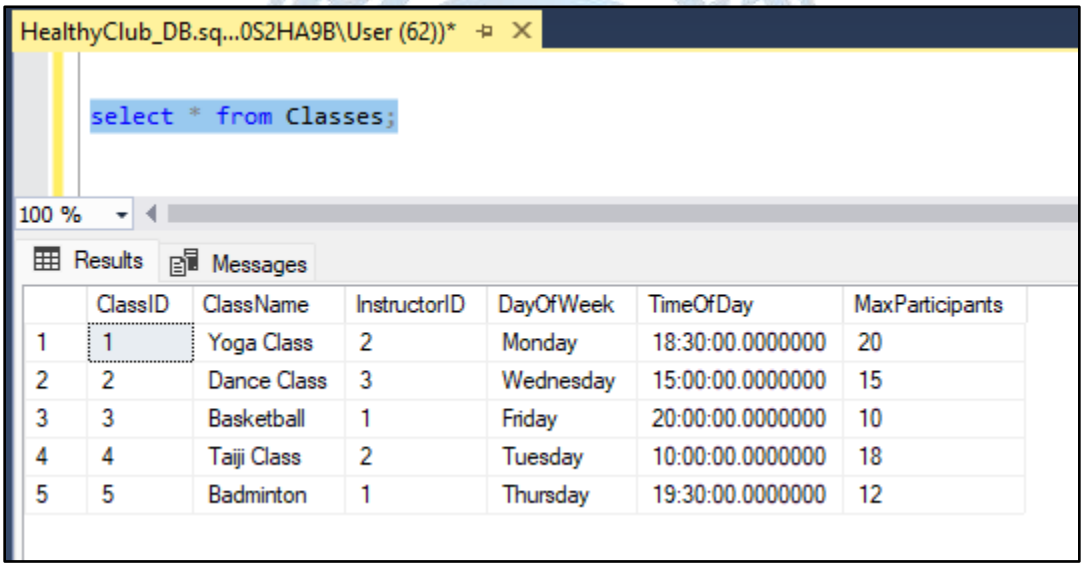
	BookingID	MemberID	FacilityID	BookingDate	StartTime	EndTime
1	1	1	1	2023-09-05	09:00:00.0000000	11:00:00.0000000
2	2	2	2	2023-09-06	16:00:00.0000000	18:00:00.0000000
3	3	3	3	2023-09-07	10:00:00.0000000	12:00:00.0000000
4	4	4	4	2023-09-08	14:00:00.0000000	16:00:00.0000000
5	5	5	5	2023-09-09	17:00:00.0000000	19:00:00.0000000

Figure 14: Insert data in FacilityBookings Table

(Source: Microsoft SQL Server Management Studio)

Classes Table

```
INSERT INTO Classes (ClassID, ClassName, InstructorID, DayOfWeek, TimeOfDay,
MaxParticipants)

VALUES

    (1, 'Yoga Class', 2, 'Monday', '18:30:00', 20),

    (2, 'Dance Class', 3, 'Wednesday', '15:00:00', 15),

    (3, 'Basketball', 1, 'Friday', '20:00:00', 10),

    (4, 'Taiji Class', 2, 'Tuesday', '10:00:00', 18),

    (5, 'Badminton', 1, 'Thursday', '19:30:00', 12);
```



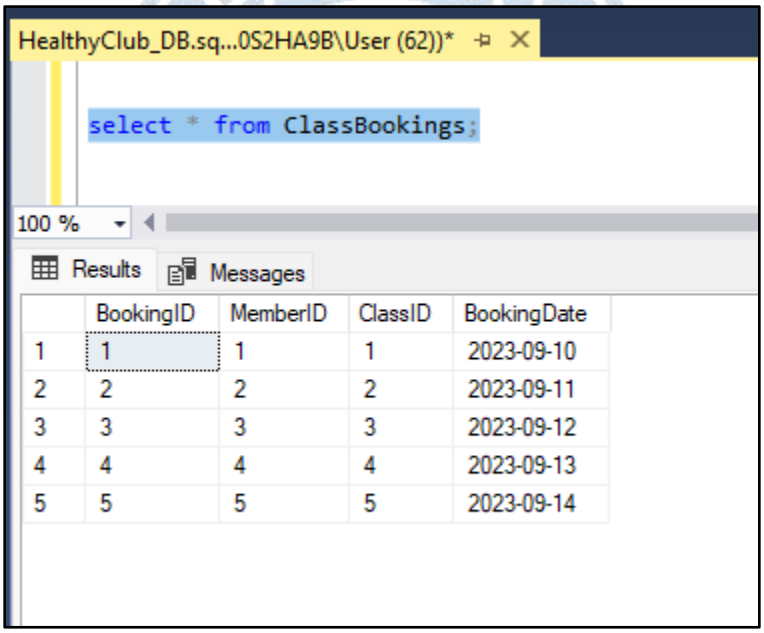
	ClassID	ClassName	InstructorID	DayOfWeek	TimeOfDay	MaxParticipants
1	1	Yoga Class	2	Monday	18:30:00.0000000	20
2	2	Dance Class	3	Wednesday	15:00:00.0000000	15
3	3	Basketball	1	Friday	20:00:00.0000000	10
4	4	Taiji Class	2	Tuesday	10:00:00.0000000	18
5	5	Badminton	1	Thursday	19:30:00.0000000	12

Figure 15: Insert data in Classes Table

(Source: Microsoft SQL Server Management Studio)

ClassBookings Table

```
INSERT INTO ClassBookings (BookingID, MemberID, ClassID, BookingDate)
VALUES
    (1, 1, 1, '2023-09-10'),
    (2, 2, 2, '2023-09-11'),
    (3, 3, 3, '2023-09-12'),
    (4, 4, 4, '2023-09-13'),
    (5, 5, 5, '2023-09-14');
```



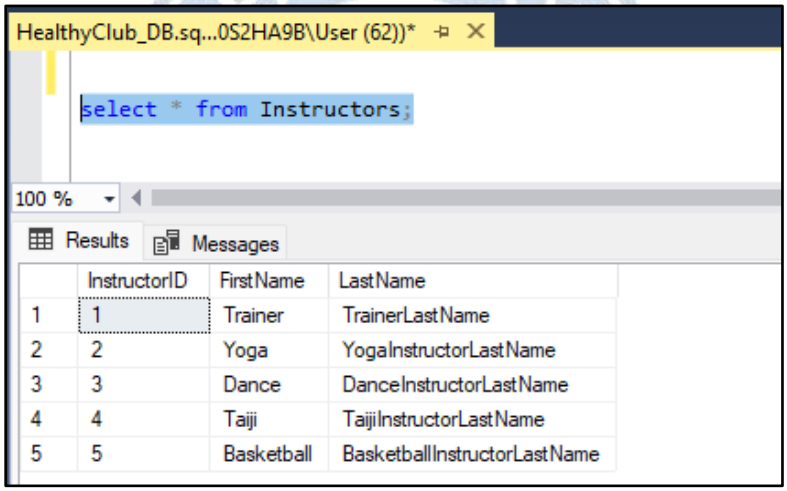
	BookingID	MemberID	ClassID	BookingDate
1	1	1	1	2023-09-10
2	2	2	2	2023-09-11
3	3	3	3	2023-09-12
4	4	4	4	2023-09-13
5	5	5	5	2023-09-14

Figure 16: Insert data in ClassBookings Table

(Source: Microsoft SQL Server Management Studio)

Instructors Table

```
INSERT INTO Instructors (InstructorID, FirstName, LastName)
VALUES
    (1, 'Trainer', 'TrainerLastName'),
    (2, 'Yoga', 'YogaInstructorLastName'),
    (3, 'Dance', 'DanceInstructorLastName'),
    (4, 'Taiji', 'TaijiInstructorLastName'),
    (5, 'Basketball', 'BasketballInstructorLastName');
```



	InstructorID	FirstName	LastName
1	1	Trainer	TrainerLastName
2	2	Yoga	YogaInstructorLastName
3	3	Dance	DanceInstructorLastName
4	4	Taiji	TaijiInstructorLastName
5	5	Basketball	BasketballInstructorLastName

Figure 17: Insert data in Instructors Table

(Source: Microsoft SQL Server Management Studio)

Chapter 5: Testing and Deployment

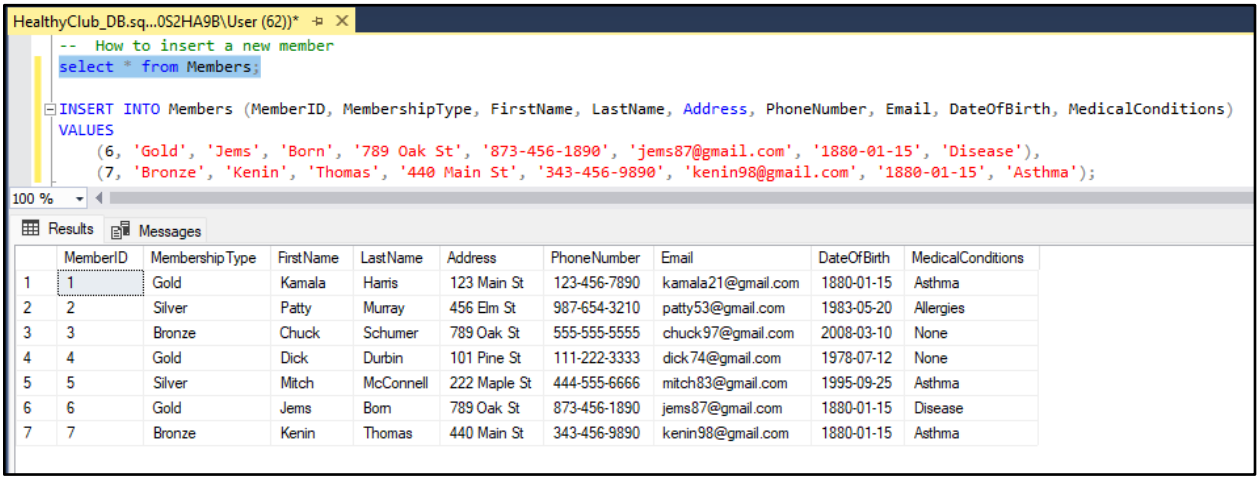
5.1 Testing Process

The testing process for the Healthy Sports Club Management System encompassed several critical functionalities: inserting a new member, booking a class, and checking attendance.

a. How to insert a new member

To validate the registration of new members, a test case is enacted. It involved executing SQL commands to insert sample member data into the "Members" table (Gamal *et al.* 2021). This step aimed to verify the system's ability to successfully incorporate new member records into the database.

```
-- How to insert a new member

select * from Members;


INSERT INTO Members (MemberID, MembershipType, FirstName, LastName, Address,
PhoneNumber, Email, DateOfBirth, MedicalConditions)

VALUES

   (6, 'Gold', 'Jems', 'Born', '789 Oak St', '873-456-1890', 'jems87@gmail.com', '1880-01-15',
'Disease'),

        (7, 'Bronze', 'Kenin', 'Thomas', '440 Main St', '343-456-9890', 'kenin98@gmail.com',
'1880-01-15', 'Asthma');
```



	MemberID	MembershipType	FirstName	LastName	Address	PhoneNumber	Email	DateOfBirth	MedicalConditions
1	1	Gold	Kamala	Harris	123 Main St	123-456-7890	kamala21@gmail.com	1880-01-15	Asthma
2	2	Silver	Patty	Murray	456 Elm St	987-654-3210	patty53@gmail.com	1983-05-20	Allergies
3	3	Bronze	Chuck	Schumer	789 Oak St	555-555-5555	chuck97@gmail.com	2008-03-10	None
4	4	Gold	Dick	Durbin	101 Pine St	111-222-3333	dick74@gmail.com	1978-07-12	None
5	5	Silver	Mitch	McConnell	222 Maple St	444-555-6666	mitch83@gmail.com	1995-09-25	Asthma
6	6	Gold	Jems	Born	789 Oak St	873-456-1890	jems87@gmail.com	1880-01-15	Disease
7	7	Bronze	Kenin	Thomas	440 Main St	343-456-9890	kenin98@gmail.com	1880-01-15	Asthma

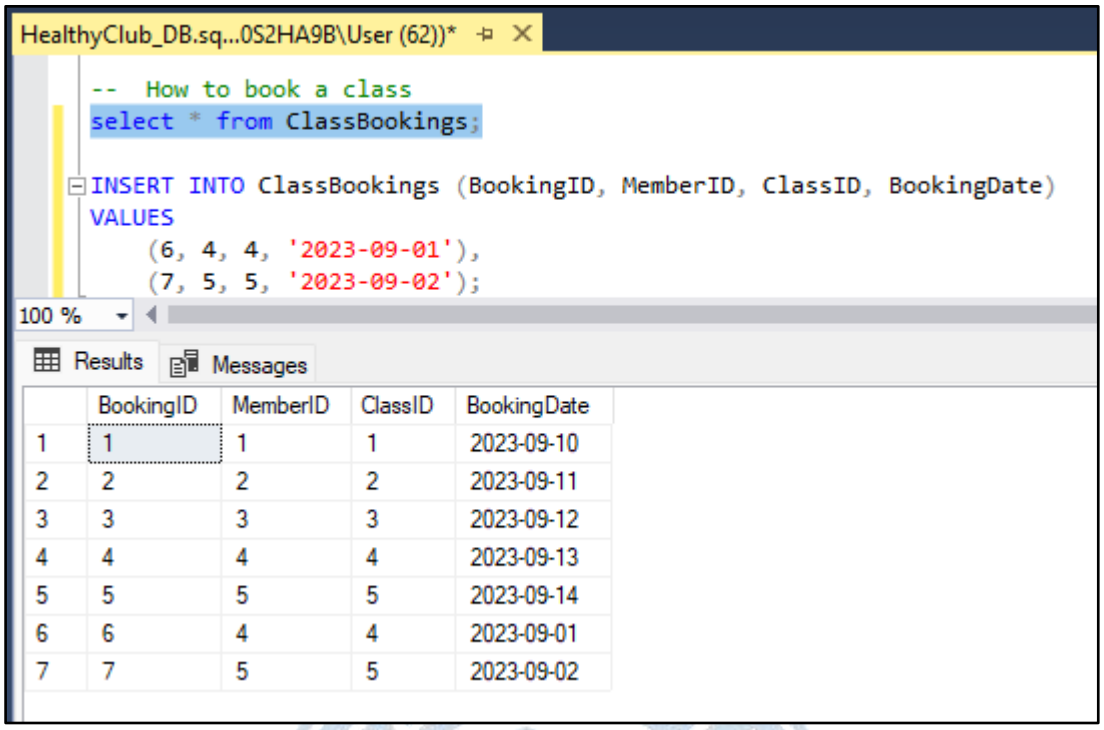
Figure 18: Insert new member

(Source: Microsoft SQL Server Management Studio)

b. How to book a class

Testing class booking functionality involved inserting booking records into the "ClassBookings" table. SQL statements are employed to create new booking entries for various members and classes on specific dates (Liu and Chen, 2022). The objective is to affirm that the system permitted members to effectively reserve classes.

```
-- How to book a class

select * from ClassBookings;



INSERT INTO ClassBookings (BookingID, MemberID, ClassID, BookingDate)

VALUES

   (6, 4, 4, '2023-09-01'),

   (7, 5, 5, '2023-09-02');
```



	BookingID	MemberID	ClassID	BookingDate
1	1	1	1	2023-09-10
2	2	2	2	2023-09-11
3	3	3	3	2023-09-12
4	4	4	4	2023-09-13
5	5	5	5	2023-09-14
6	6	4	4	2023-09-01
7	7	5	5	2023-09-02

Figure 19: Book class

(Source: Microsoft SQL Server Management Studio)

c. How to check my attendance

To authenticate the attendance-checking feature, a SQL query is devised to retrieve attendance details for a designated member (Biswas *et al.* 2020). The query extracted class names, booking dates, and attendance status (either "attended" or "not attended") based on the MemberID. This step substantiated that members could conveniently review their class attendance records.

```
-- How to check my attendance ?

DECLARE @MemberID INT = 3;

SELECT

    Classes.ClassName,

    ClassBookings.BookingDate,

    CASE

        WHEN ClassBookings.MemberID IS NOT NULL THEN 'Attended'

        ELSE 'Not Attended'

    END AS AttendanceStatus

FROM Classes

LEFT JOIN ClassBookings ON Classes.ClassID = ClassBookings.ClassID

WHERE ClassBookings.MemberID = @MemberID;
```

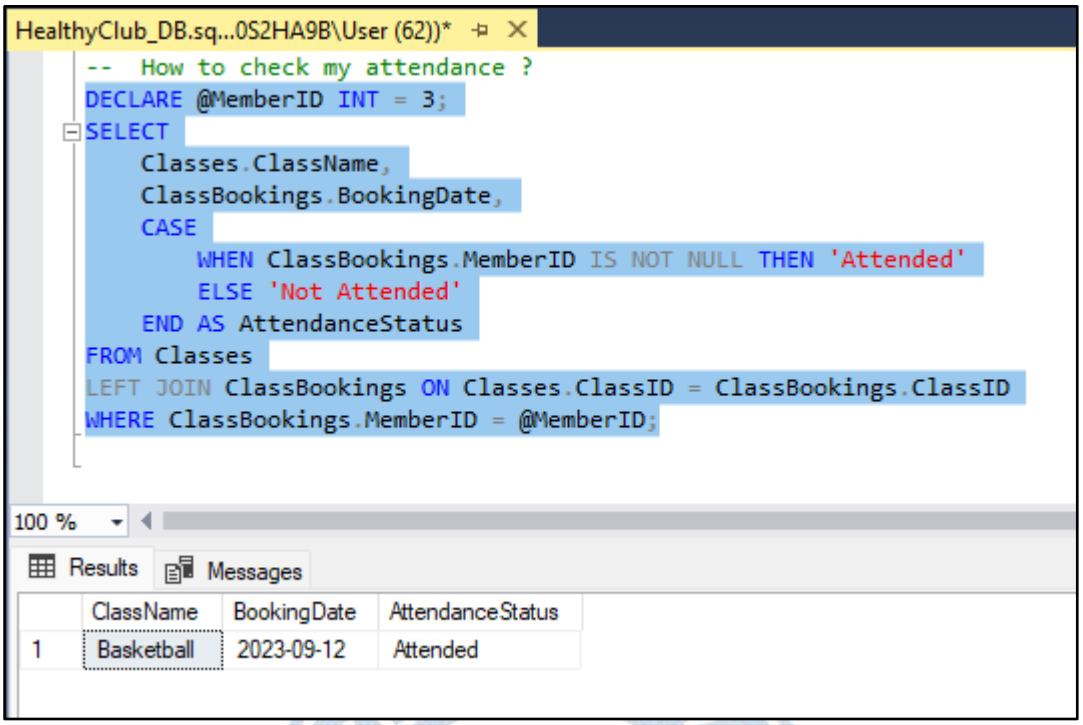


Figure 20: Check Attendance

(Source: Microsoft SQL Server Management Studio)

These testing procedures are instrumental in ensuring the dependability and precision of the Healthy Sports Club Management System, affirming that members can register, book classes, and monitor their attendance seamlessly within the system (Madubuike and Anumba, 2023).

d. Current week activities booked by a club member

```
#displays the current week activities booked by a club member.

SELECT c.ClassName, c.DayOfWeek, c.TimeOfDay, cb.BookingDate

FROM ClassBookings cb

JOIN Classes c ON cb.ClassID = c.ClassID

WHERE cb.MemberID = 3
```

```
AND YEARWEEK(cb.BookingDate, 1) = YEARWEEK(CURDATE(), 1);
```

```
193
194
195     #b.displays the current week activities booked by a club member.
196     SELECT c.ClassName, c.DayOfWeek, c.TimeOfDay, cb.BookingDate
197     FROM ClassBookings cb
198     JOIN Classes c ON cb.ClassID = c.ClassID
199     WHERE cb.MemberID = 3
200       AND YEARWEEK(cb.BookingDate, 1) = YEARWEEK(CURDATE(), 1);
201
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ClassName	DayOfWeek	TimeOfDay	BookingDate
Basketball	Friday	20:00:00	2023-09-08

Figure 21: Displaying week activities

(Source: Microsoft SQL Server Management Studio)

e. Search classed offered by an instructor in the current week

```
#c.search classed offered by an instructor in the current week

SELECT c.ClassName, c.DayOfWeek, c.TimeOfDay

FROM Classes c

JOIN Instructors i ON c.InstructorID = i.InstructorID

WHERE i.FirstName = 'Dance'

  AND i.LastName = 'DanceInstructorLastName';
```

```
205     #c.search classed offered by an instructor in the current week
206 •   SELECT c.ClassName, c.DayOfWeek, c.TimeOfDay
207     FROM Classes c
208     JOIN Instructors i ON c.InstructorID = i.InstructorID
209     WHERE i.FirstName = 'Dance'
210       AND i.LastName = 'DanceInstructorLastName';
211
212
213
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ClassName	DayOfWeek	TimeOfDay
Dance Class	Wednesday	15:00:00

Figure 22: Search classes offered by instructer

(Source: Microsoft SQL Server Management Studio)

f. Most active members monthly (most bookings and most attendance)

```
#d.show most active members monthly (most bookings and most attendance)
SELECT
    M.MemberID,
    M.FirstName,
    M.LastName,
    COUNT(FB.BookingID) AS TotalFacilityBookings,
    COUNT(CB.BookingID) AS TotalClassAttendances
FROM
```

```
    Members M

LEFT JOIN

    FacilityBookings FB ON M.MemberID = FB.MemberID

LEFT JOIN

    ClassBookings CB ON M.MemberID = CB.MemberID

WHERE

    DATE_FORMAT(IFNULL(FB.BookingDate, CB.BookingDate), '%Y-%m') = '2023-09'  -- Specify the desired month and year here

GROUP BY

    M.MemberID,

    M.FirstName,

    M.LastName

ORDER BY

    (COUNT(FB.BookingID) + COUNT(CB.BookingID)) DESC;
```

```
215
216     #d.show most active members monthly (most bookings and most attendance)
217 •   SELECT
218         M.MemberID,
219         M.FirstName,
220         M.LastName,
221         COUNT(FB.BookingID) AS TotalFacilityBookings,
222         COUNT(CB.BookingID) AS TotalClassAttendances
223     FROM
224         Members M
225     LEFT JOIN
226         FacilityBookings FB ON M.MemberID = FB.MemberID
227     LEFT JOIN
228         ClassBookings CB ON M.MemberID = CB.MemberID
229     WHERE
230         DATE_FORMAT(IFNULL(FB.BookingDate, CB.BookingDate), '%Y-%m') = '2023-09'
231     GROUP BY
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MemberID	FirstName	LastName	TotalFacilityBookings	TotalClassAttendances
1	Kamala	Harris	1	1
2	Patty	Murray	1	1
3	Chuck	Schumer	1	1
4	Dick	Durbin	1	1
5	Mitch	McConnell	1	1

Figure 23: Showing most active members

(Source: Microsoft SQL Server Management Studio)

Deployment Process

The deployment process of the Healthy Sports Club Management System involved critical aspects such as scaling, security, and operational regulations. Different deployment options are considered, including local hosting and cloud-based solutions (Nidhya *et al.* 2022). The choice is influenced by factors like budget and scalability needs, ultimately favouring an option that aligned with these requirements. Security and maintenance issues included the protection of member data, regular software updates, and database backups. Operational regulations encompassed compliance with data privacy laws and ensuring the system's availability and

reliability for members and staff. These considerations played a crucial role in the effective deployment and ongoing operation of the system.

Chapter 6: Conclusion

6.1 Key findings and Conclusion

This study has presented the Healthy Sports Club Management System, highlighting its successful design, implementation, and testing. The system effectively meets the requirements of the clubs, and offers features for member registration, booking class, and checking the attendance. It relies on MySQL as the database system and has proven its efficacy.

6.2 Recommendations for future work

There are opportunities to improve the system. Consideration can be given to integrating the payment processing for the membership fees, enhancing their reporting capabilities, and developing a user-friendly mobile app for the members, that they can use easily. Ongoing attention to security updates and data protection is needed and crucial. This system's adaptability to evolving the club needs is vital for its long-term success.

References

Alabi, I.O., Salau, R.B. and Sesugh, S.L., 2021. NutriMax: an Android Based Personalized Nutrition Management System.

Biswas, S., Sharif, K., Li, F., Latif, Z., Kanhere, S.S. and Mohanty, S.P., 2020. Interoperability and synchronization management of blockchain-based decentralized e-health systems. IEEE Transactions on Engineering Management, 67(4), pp.1363-1376.

Byun, J., 2021. A Study on Contributor to Sports Development Big Data Research Using Oral Records. Journal of Multimedia Information System, 8(4), pp.301-308.

Čerešňák, R. and Kvet, M., 2019. Comparison of query performance in relational a non-relation databases. Transportation Research Procedia, 40, pp.170-177.

CHAITHRA, B., 2021. SPORTS EVENT MANAGEMENT SYSTEM (Doctoral dissertation, KUVEMPU UNIVERSITY).

Chen, K., 2022, April. Database-Oriented Sports Competition Information Management Intelligent Retrieval System Based on Computer Multi-Heterogeneous Disaster Recovery Backup. In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1727-1730). IEEE.

Cheng, J.C., Chiu, C.Y. and Su, T.J., 2019. Training and evaluation of human cardiorespiratory endurance based on a fuzzy algorithm. International journal of environmental research and public health, 16(13), p.2390.

Epstein, R.H., Hofer, I.S., Salari, V. and Gabel, E., 2021. Successful implementation of a perioperative data warehouse using another hospital's published specification from epic's electronic health record system. Anesthesia & Analgesia, 132(2), pp.465-474.

Gamal, A., Barakat, S. and Rezk, A., 2021. Standardized electronic health record data modeling and persistence: A comparative review. Journal of biomedical informatics, 114, p.103670.

Grasso, C., Goldhammer, H., Funk, D., King, D., Reisner, S.L., Mayer, K.H. and Keuroghlian, A.S., 2019. Required sexual orientation and gender identity reporting by US health centers: first-year data. American Journal of Public Health, 109(8), pp.1111-1118.

Kuo, Y.W., Tsao, Y.C., Chien, W.C., Huang, Y.M. and Liao, L.D., 2022. Smart health monitoring and management system for organizations using radio-frequency identification (RFID) technology in hospitals or emergency applications. Emergency Medicine International, 2022.

Liu, H. and Chen, X., 2022. Construction and optimization of mental health education consultation management system based on decision tree association rule mining. Mathematical Problems in Engineering, 2022.

Madubuike, O.C. and Anumba, C.J., 2023. Digital twin–based health care facilities management. Journal of Computing in Civil Engineering, 37(2), p.04022057.

Namdeo, B. and Suman, U., 2021. A Middleware Model for SQL to NoSQL Query Translation.

Nidhya, R., Kumar, M., Maheswar, R. and Pavithra, D., 2022. Security and privacy issues in smart healthcare system using internet of things. IoT-Enabled Smart Healthcare Systems, Services and Applications, pp.63-85.

Qin, P. and Feng, W., 2021. Design of the Exercise Load Data Monitoring System for Exercise Training Based on the Neural Network. Journal of Healthcare Engineering, 2021, pp.1-6.

Song, J., Dou, W., Cui, Z., Dai, Q., Wang, W., Wei, J., Zhong, H. and Huang, T., 2023. Testing database systems via differential query execution. In Proceedings of IEEE/ACM International Conference on Software Engineering (ICSE).

Tao, J., Wang, W., Su, Y., Deng, H., Chen, H. and Tian, Y., 2022. Research on display and promotion platform of intelligent image processing technology achievements. American Journal of Applied Sciences (ISSN: 2321–089X), 13(1).

Zhang, H., 2022. Design and Implementation of Sports Industry Information Service Management System Based on Data Mining. Advances in Multimedia, 2022.


RESEARCHRYNO
INNOVATIONS
EST. 2018