



NAME – ABRAHAM PHILIP SOLOMON CHARLES THEODORE

STUDENT ID – 1973153

**UNIT TITLE AND CODE – CIS108-6 DATA MODELLING,
MANAGEMENT AND GOVERNANCE**

ASSIGNMENT NUMBER AND TITLE – ASSIGNMENT FINAL

DATA SYSTEM OF 'THE HEALTHY SPORT CLUB'

Executive Summary

The case study on the "HEALTHY" local sports club executive summary indicates a health and wellness business that offers a variety of physical activities and services. The club has facilities like gym, sport hall, yoga room for dancing and sauna rooms with a team of 20 staff to give personal training, group exercises and special classes. The minimum system functionalities needed are membership management with tiered privileges, staff information recording, and facility or space booking restrictions. The system design consists of tables for members, staff, facilities, classes, courses, and bookings, which generally assists effortless data storage and lookup. Database schema MySQL provides data integrity, with foreign key constraints, and sample data in aid of better illustration. The system is designed to have a capacity to book facilities for classes, record attendance for classes and track membership information. For future work, there are recommendations of improving the membership management, revising the booking rules, constructing a member application for accessing the facilities, implementing analytics for business insights, and building tools for instructors such as booking classes and collecting feedback. These changes made with the aim of enhancing the operations and member experience at the "HEALTHY" sports club are in quest.

Table of Contents

1. Introduction.....	4
2. Requirements Analysis	4
2.1 Use Case Diagram.....	5
2.2 Activity Diagram.....	6
3. System design	7
3.1 Data Modeling.....	8
3.2 Entity Relationship Diagram.....	9
4. System Implementation	10
4.1 Member	10
4.2 Staff	12
4.3 Facility.....	13
4.4 Class	15
4.5 Instructor	16
4.6 Booking	18
5. Testing and Deployment.....	21
5.1 Inserting a new member into the Member table.....	21
5.2 Book a Class for a Club Member	21
5.3 Inserting attendance records.....	22
6. Conclusion	23
6.1 Key findings and conclusions	23
6.2 The recommendations for future work on the system.....	23
List of the References	24

1. Introduction

"HEALTHY" sports club is the local thriving business that aim at spreading of health and wellbeing through the fitness programs and variety of classes offered. "HEALTHY" encompasses an exercise complex equipped with a fitness center and a sports arena, with yoga and dance studios, and a sauna and massage rooms in the background, providing a rich experience for its members. The club is a proud to have an experienced team of about 20 staff members that specializes in working one-on-one, group exercise and specialized classes. Thus, "HEALTHY" realizes that an efficient data system is crucial in order to manage with its operations and increase the membership satisfaction. It will be devised in such a way that it will cater for membership registration, employee management, scheduling of classes, and booking of facilities accurately and in the right timing. Participants will make the most of the booking procedures that have been simplified and will enjoy all the details about class timings and facilities. Staff members, including teachers, will have access to instruments that enable them to calculate and monitor class schedules and attendance without problems.

2. Requirements Analysis

The analysis of requirements for the "Healthy" Sports Club Management System involves such processes as exploration of users' needs and limiting business' constraints in order to develop a software solution that is effective (Yang *et al.* 2024).

- **Membership Management:** The system requires to implement several tier levels of membership (Gold, Silver, Bronze), which will be accompanied by different privileges and fees. Profiles of all members must contain their personal data including name, address, contact info, date of birth and any medical conditions while their current membership lasts.
- **Staff Management:** Workforce data, which entails trainers and instructors, must be put down with details including name, job, and contact. Teachers should have permission to put forward new classes, supervise class attendance and change their personal details.
- **Facility Booking:** The members should be allowed to reserve facilities such as sports hall, yoga rooms, as well as sauna rooms (Hu *et al.* 2024). The system must apply the following regulations: one week booking per user per facility and time slot management to avoid members from overlapping.
- **Class Management:** The system must make it possible for instructors to deliver and schedule classes, as well as to give a name, set a schedule, determine the limit of participants,

and to associate a facility. Members will be able to identify the current classes, place their booking, and organize their weekly booking.

- **Data Management:** A system of this nature requires an administrative component for managing data creation, querying, updating, and deleting, including instructors, members, classes, and facilities. Similarly, members and instructors should be able to use the platform to view and update the personal contact information.
- **Integration and Reporting:** The system should integrate all components amalgamated properly to provide real-time data on facilities using, class attendance, and members' development. Analysis should offer data-driven insights on business processes and outcomes.
- **User Interface:** The interface has to be designed in such a way that it can be easily understood by the administrators and members of the organization. The interface should be available both via web and mobile devices, through which easy navigation and interaction with the platform can be ensured.
- **Security and Compliance:** Security measures, including data privacy and security must be put in place to keep member information safe. The system should meet the legal criteria and also the relevant data handling and storage standards (Kim *et al.* 2024).

2.1 Use Case Diagram

The use case diagram for the "HEALTHY" sports club system shows the flow of information between different actors and the system's feature functionality. The next diagram describes the "Admin Staff," "Class Instructor" and "Club Member" actors, and their key actions. The Admin Staff are assigned the following tasks: managing professors, members, sessions, facilities and reports generated by the system (Mickelsson, 2024).

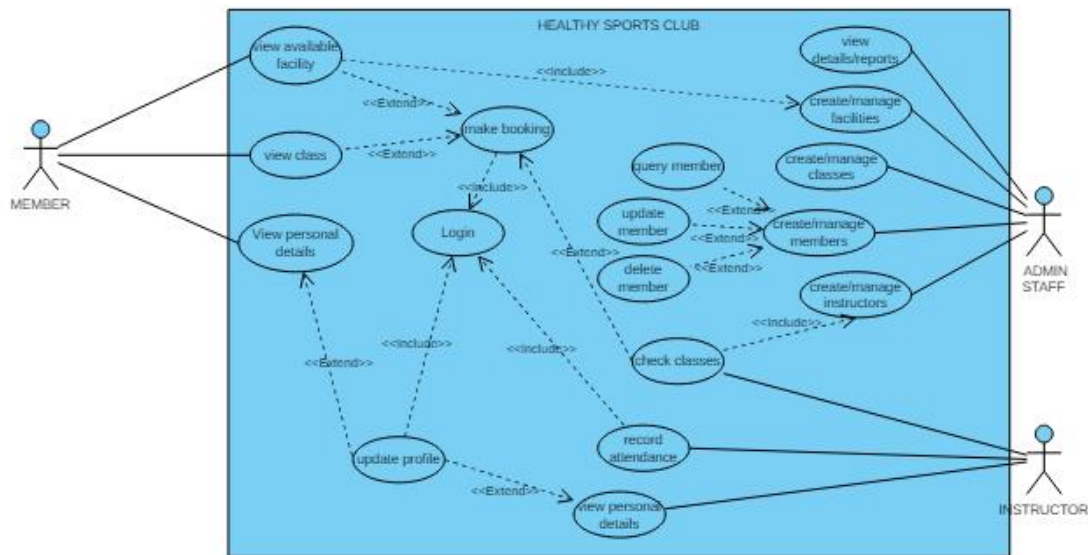


Figure 1: Use Case Diagram

(Source: Self-created)

The Class Instructors can submit requests for new classes through the system and also sign in their attendance of classes they teach. The club members can have access to all the available classes on the system, make reservations to a specific class and also check their personal accounts.

2.2 Activity Diagram

The operation diagram shows operational flows of the "HEALTHY" sports club, which is done by clients, instructors, and administrative personnel. Members use the club's service and then make their choice. They can either book facilities or attend classes they are interested in. Facility booking is done by members. Searching for availability is first and proceeding with booking if the found options are ideal and if not, alternative options are given.

(Source: Self-created)

3. System design

7

- **User Interface (UI):** It focuses on developing an interface that is intuitive and responsive which can be accessed through web and mobile devices. Facility access should be intuitive for members and staff. So, they can view schedules, book facilities, and manage their profiles. UI includes management console for admin to monitor activities, handle membership matters and bookings conveniently.
- **Database Schema:** The data schema is essential for data storage and arrangement in the form of members, staff, structures, classes, and bookings. Tables such as Member, Staff, Facility, Class, Instructor, and Booking will be introduced. Associations will be established accordingly. Keys will be established as primary and foreign ones in order to guarantee the data integrity and to decrease the time for querying.
- **Application Logic:** The algorithm will take care of the business rules, e.g., membership privileges, booking restrictions and time scheduling (Blomqvist Mickelsson, 2024). Scheduling algorithms will be implemented to plan availability of facilities, time slot management, and class attendance recording to optimize resources and automatize process.
- **Integration:** The Integration of different modules will make communication between membership management, facility booking, class scheduling, and staff administration smooth and easy. API can be applied to integrate an external payment system and make a communication easier with other services.
- **Security and Scalability:** The system will be designed with security features such as data encryption, role-based access control and regular backups in order to secure the information.

3.1 Data Modeling

Data modeling quite plays a crucial role in representation of data structure and its relationships within the system. The "Healthy" Sports Club Management System will model relational database modeling structure to store and process data in an organized way.

- **Entity-Relationship (ER) Diagram:** An ER diagram will not only visually display entities (e.g., Member, Staff, Facility) and their relationships (e.g., Member uses Facility, Instructor teaches Class) but also depict the key interactions and dependencies. Relationships will be identified using cardinality as they determine one-to-one, one-to-many, and many-to-many associations.
- **Tables and Attributes:** Each element of the ER diagram maps directly to an entity (e.g., Member, Staff) of the table with its targeted attributes (e.g., membership_number,

first_name). Original records of each table will be identified by the primary key, which is a foreign key that establishes linkages between tables.

- **Normalization:** Database normalization will be used to be rid of duplication issues and guarantee data consistency. The data management process here is all about organizing data into the tables that are related to each other while reducing the duplication and inconsistencies of updating.
- **Data Integrity and Constraints:** Integrity constraints such as NOT NULL, UNIQUE, and FOREIGN KEY will be applied to ensure data accountability and consistency. Restrictions will prevent wrong data entries and retain the referential data integrity (Kuo *et al.* 20

3.2 Entity Relationship Diagram

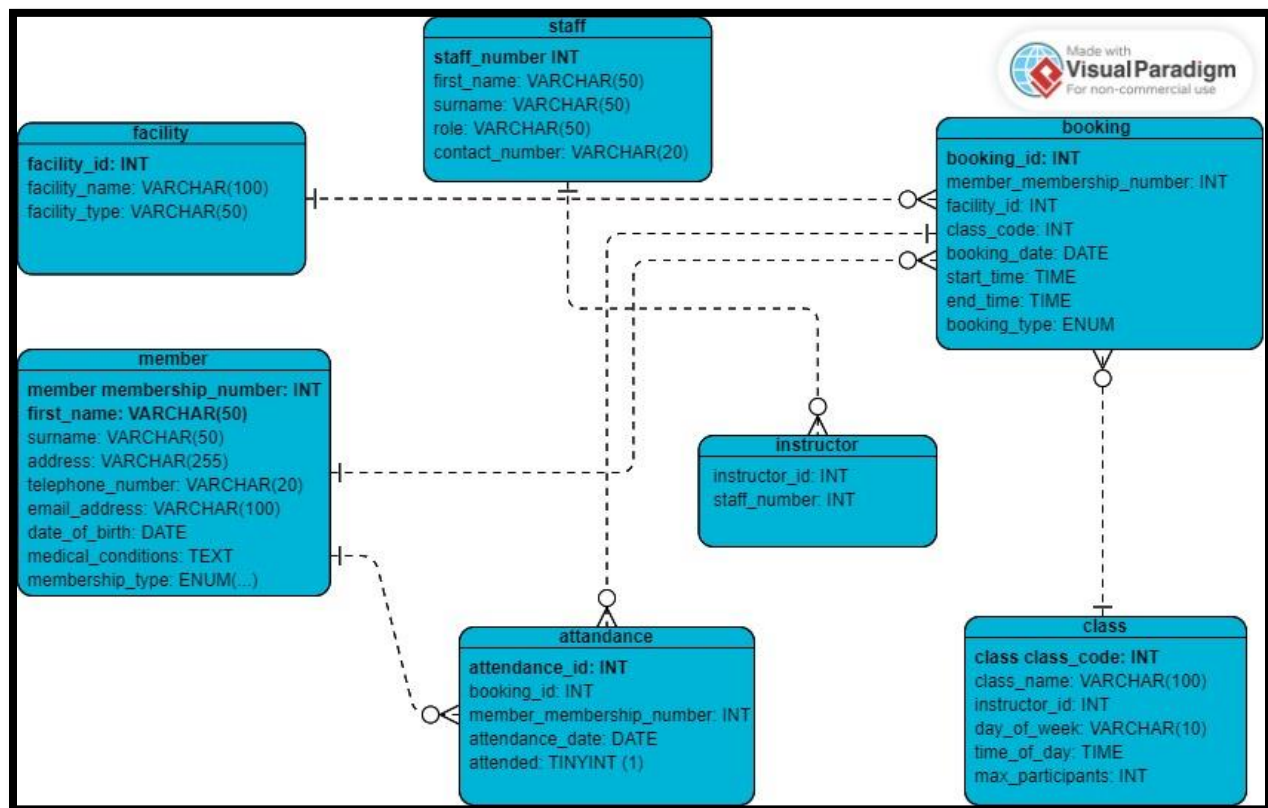


Figure 3: Entity Relationship Diagram

(Source: Self-created)

An Entity-Relationship Diagram (ERD) is the visual representation of the relationships between entities which the database schema of study "Healthy" sports club research is based on. The components of this model are Facility, Member, Staff, Attendance, Booking, and Class. The

corresponding figure represents each entity as a rectangle with its attributes inside, which. To show relationships entity lines are used that are designed in a way to visualize how they are working. For instance, Facility entity attributes include, for instance, facility_id, facility_name, and facility_type.

4. System Implementation

4.1 Member

```
1      -- Table: Member
2  ● ○ CREATE TABLE Member (
3      membership_number INT PRIMARY KEY,
4      first_name VARCHAR(50) NOT NULL,
5      surname VARCHAR(50) NOT NULL,
6      address VARCHAR(255),
7      telephone_number VARCHAR(20),
8      email_address VARCHAR(100),
9      date_of_birth DATE,
10     medical_conditions TEXT,
11     membership_type ENUM('Gold', 'Silver', 'Bronze') NOT NULL
12 ) ;
```

Figure 4: Create table: Member

(Source: Implemented by using MySQL Workbench)

The code contains a specify MySQL table called "Member" which is meant to store data about club members. For every member, there is given an individual identification number - membership_number. The table includes columns names like first_name, surname, address, telephone_number, email_address, date_of_birth, medical_conditions and membership_type. membership_type column is a datatype ENUM which want to be a membership level (Gold, Silver, Bronze). membership_type column validates against predefined values and provides always membership_type.

```
-- Inserting 10 sample values into the Member table
INSERT INTO Member (membership_number, first_name, surname, address, telephone_number, email_address, date_of_birth, medical_conditions, membership_type)
VALUES
(1, 'John', 'Doe', '123 Main St, City, Country', '123-456-7890', 'john.doe@example.com', '1990-05-15', 'None', 'Gold'),
(2, 'Jane', 'Smith', '456 Oak Ave, Town, Country', '987-654-3210', 'jane.smith@example.com', '1985-09-20', 'Allergies: peanuts', 'Silver'),
(3, 'David', 'Johnson', '789 Elm Rd, Village, Country', '555-123-4567', 'david.johnson@example.com', '1988-12-10', 'None', 'Bronze'),
(4, 'Emily', 'Brown', '321 Pine Blvd, City, Country', '222-333-4444', 'emily.brown@example.com', '1995-03-25', 'Asthma', 'Gold'),
(5, 'Michael', 'Wilson', '567 Cedar St, Town, Country', '444-555-6666', 'michael.wilson@example.com', '1992-07-08', 'None', 'Silver'),
(6, 'Sarah', 'Miller', '890 Birch Ave, Village, Country', '777-888-9999', 'sarah.miller@example.com', '1983-11-30', 'Diabetes', 'Bronze'),
(7, 'Alex', 'Lee', '234 Oakwood Dr, City, Country', '111-222-3333', 'alex.lee@example.com', '1998-09-12', 'None', 'Gold'),
(8, 'Jessica', 'Chang', '678 Maple Rd, Town, Country', '999-000-1111', 'jessica.chang@example.com', '1991-04-18', 'None', 'Silver'),
(9, 'Kevin', 'Garcia', '901 Pinecrest Ave, Village, Country', '333-666-9999', 'kevin.garcia@example.com', '1987-02-05', 'High blood pressure', 'Bronze'),
(10, 'Laura', 'Martinez', '345 Cedarwood Ln, City, Country', '555-777-8888', 'laura.martinez@example.com', '1994-06-22', 'None', 'Gold');
```

1 • SELECT * FROM healthy_sports_club.member;

	membership_number	first_name	surname	address	telephone_number	email_address	date_of_birth	medical_conditions	membership_type
1	1	John	Doe	123 Main St, City, Country	123-456-7890	john.doe@example.com	1990-05-15	None	Gold
2	2	Jane	Smith	456 Oak Ave, Town, Country	987-654-3210	jane.smith@example.com	1985-09-20	Allergies: peanuts	Silver
3	3	David	Johnson	789 Elm Rd, Village, Country	555-123-4567	david.johnson@example.com	1988-12-10	None	Bronze
4	4	Emily	Brown	321 Pine Blvd, City, Country	222-333-4444	emily.brown@example.com	1995-03-25	Asthma	Gold
5	5	Michael	Wilson	567 Cedar St, Town, Country	444-555-6666	michael.wilson@example.com	1992-07-08	None	Silver
6	6	Sarah	Miller	890 Birch Ave, Village, Country	777-888-9999	sarah.miller@example.com	1983-11-30	Diabetes	Bronze
7	7	Alex	Lee	234 Oakwood Dr, City, Country	111-222-3333	alex.lee@example.com	1998-09-12	None	Gold
8	8	Jessica	Chang	678 Maple Rd, Town, Country	999-000-1111	jessica.chang@example.com	1991-04-18	None	Silver
9	9	Kevin	Garcia	901 Pinecrest Ave, Village, Country	333-666-9999	kevin.garcia@example.com	1987-02-05	High blood pressure	Bronze
10	10	Laura	Martinez	345 Cedarwood Ln, City, Country	555-777-8888	laura.martinez@example.com	1994-06-22	None	Gold
*		NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 5: Table: Member

(Source: Implemented by using MySQL Workbench)

The provided code segment serves to the purpose of adding 10 rows of sample member records in the "Member" table. Each record is a club member that has specific features like the membership number, first names, surnames, addresses, telephone numbers, email addresses, date of birth, medical conditions, and membership type. For example, John Doe, Gold member at 123 Main St, City, has a contact number 123-456-7890, no existing medical conditions reported. The roles, membership types and medical conditions of the other members like Jane Smith, David Johnson, and Emily Brown also differ from the general population. This SQL code will use the "Member" table with the sample data which is different kind of member profiles to simulate club membership.

4.2 Staff

```
14  -- Table: Staff
15  ● ○ CREATE TABLE Staff (
16      staff_number INT PRIMARY KEY,
17      first_name VARCHAR(50) NOT NULL,
18      surname VARCHAR(50) NOT NULL,
19      role VARCHAR(50),
20      contact_number VARCHAR(20)
21  );
```

Figure 6: Create table: Staff

(Source: Implemented by using MySQL Workbench)

The given SQL code specifies a table named "Staff" having columns with data information. The table contains the primary key 'staff_number' for identification purposes. Others cover the basic staff attributes including the names of staff, surname, role, and contacts. The "NOT NULL" condition assures the presence of this data, which facilitates the fullness of the staff records. This structure allows to effectively and accurately manage employee data with the system.

```
78  -- Inserting sample data into the Staff table
79  ● INSERT INTO Staff (staff_number, first_name, surname, role, contact_number)
80  VALUES
81      (1, 'John', 'Doe', 'Fitness Trainer', '123-456-7890'),
82      (2, 'Jane', 'Smith', 'Yoga Instructor', '987-654-3210'),
83      (3, 'Mike', 'Johnson', 'Personal Trainer', '555-123-4567'),
84      (4, 'Emily', 'Wilson', 'Zumba Instructor', '777-888-9999'),
85      (5, 'David', 'Brown', 'Nutritionist', '333-444-5555'),
86      (6, 'Sarah', 'Taylor', 'Pilates Instructor', '111-222-3333'),
87      (7, 'Chris', 'Anderson', 'Group Exercise Instructor', '999-888-7777'),
88      (8, 'Amy', 'Clark', 'Massage Therapist', '666-555-4444'),
89      (9, 'Mark', 'Roberts', 'Front Desk Receptionist', '222-333-4444'),
90      (10, 'Laura', 'Davis', 'Fitness Trainer', '777-999-1111');
91
```

```
1 • SELECT * FROM healthy_sports_club.staff;
```

staff_number	first_name	surname	role	contact_number
1	John	Doe	Fitness Trainer	123-456-7890
2	Jane	Smith	Yoga Instructor	987-654-3210
3	Mike	Johnson	Personal Trainer	555-123-4567
4	Emily	Wilson	Zumba Instructor	777-888-9999
5	David	Brown	Nutritionist	333-444-5555
6	Sarah	Taylor	Pilates Instructor	111-222-3333
7	Chris	Anderson	Group Exercise Instructor	999-888-7777
8	Amy	Clark	Massage Therapist	666-555-4444
9	Mark	Roberts	Front Desk Receptionist	222-333-4444
10	Laura	Davis	Fitness Trainer	777-999-1111
NULL	NULL	NULL	NULL	NULL

Figure 7: Table: Staff

(Source: Implemented by using MySQL Workbench)

The implemented SQL query enters sample data into the Staff table that refers to the staff members of the "HEALTHY" sports club. Each staff member is initialized with a unique staff_number and is associated with the attributes such as first_name, surname, role, and contact_number. This code segment is designed to populate the Staff table with data about the various club positions, namely Fitness Trainer, Yoga Instructor, Zumba Instructor, and others, along with the relevant contact information.

4.3 Facility

```
23 -- Table: Facility
24 • CREATE TABLE Facility (
25     facility_id INT PRIMARY KEY,
26     facility_name VARCHAR(100) NOT NULL,
27     facility_type VARCHAR(50) NOT NULL
28 );
```

Figure 8: Create table: Facility

(Source: Implemented by using MySQL Workbench)

The provided SQL code defines a "Facility" table with three columns: facility_id (integer, column primary key), facility_name (required string 100 characters maximum), and facility_type

(required string 50 characters maximum). By means of this table would be storing the information about the different facilities that are available at the sports club, such as special identifiers, names, and type (example: sports, fitness, relaxation).

```

93      -- Insert 10 values into the Facility table
94 •    INSERT INTO Facility (facility_id, facility_name, facility_type) VALUES
95      (1, 'Main Hall', 'Sports'),
96      (2, 'Yoga Studio', 'Fitness'),
97      (3, 'Basketball Court', 'Sports'),
98      (4, 'Dance Studio', 'Fitness'),
99      (5, 'Volleyball Court', 'Sports'),
100     (6, 'Sauna Room', 'Relaxation'),
101     (7, 'Badminton Court', 'Sports'),
102     (8, 'Massage Room 1', 'Wellness'),
103     (9, 'Tennis Court', 'Sports'),
104     (10, 'Pilates Studio', 'Fitness');

```

```

1 •    SELECT * FROM healthy_sports_club.facility;

```

Result Grid			
Filter Rows:			
	facility_id	facility_name	facility_type
▶	1	Main Hall	Sports
	2	Yoga Studio	Fitness
	3	Basketball Court	Sports
	4	Dance Studio	Fitness
	5	Volleyball Court	Sports
	6	Sauna Room	Relaxation
	7	Badminton Court	Sports
	8	Massage Room 1	Wellness
	9	Tennis Court	Sports
	10	Pilates Studio	Fitness
✱	NULL	NULL	NULL

Figure 9: Table: Facility

(Source: Implemented by using MySQL Workbench)

SQL code provided inserts 10 rows of data to the Facility table. The table shown below reflects the club facilities of the "HEALTHY" sports club where the facilities are identified by their unique ID, facility name and type. Facilities will include sport-related areas that include

basketball and volleyball courts as well as massage and wellness amenities in a sauna room. The line editor, on the other side, fills in the Facility table with vital information concerning facility management and booking.

4.4 Class

```
30      -- Table: Class
31 • CREATE TABLE Class (
32         class_code INT PRIMARY KEY,
33         class_name VARCHAR(100) NOT NULL,
34         instructor_id INT,
35         day_of_week VARCHAR(10),
36         time_of_day TIME,
37         max_participants INT
38     );
```

Figure 10: Create table: Class

(Source: Implemented by using MySQL Workbench)

The SQL code given in this clause creates a "Class" table with columns to capture class data from the sports club. Each class has its own unique class_code which is assigned to it. The table contains the fields like class_name (class name), instructor_id (the instructor who teaches the class), day_of_week (the day of the week when the class is held), time_of_day (the time of the class), and max_participants (the maximum number of participants who are allowed in the class). This data structure on its part improves the efficiency of the management and organization of the class-related materials in the database.

```
107      -- Inserting sample values into the Class table
108 • INSERT INTO Class (class_code, class_name, instructor_id, day_of_week, time_of_day, max_participants)
109     VALUES
110     (1, 'Yoga Class', 101, 'Monday', '10:00:00', 20),
111     (2, 'Zumba Class', 102, 'Tuesday', '15:30:00', 25),
112     (3, 'Pilates Class', 103, 'Wednesday', '11:00:00', 18),
113     (4, 'Basketball Training', 104, 'Thursday', '17:00:00', 15),
114     (5, 'Badminton Practice', 105, 'Friday', '14:00:00', 12),
115     (6, 'Dance Workshop', 106, 'Saturday', '12:30:00', 30),
116     (7, 'Kickboxing Class', 107, 'Sunday', '16:00:00', 20),
117     (8, 'Aerobics Session', 108, 'Monday', '18:00:00', 25),
118     (9, 'Tai Chi Class', 109, 'Tuesday', '09:30:00', 15),
119     (10, 'Volleyball Practice', 110, 'Wednesday', '19:00:00', 10);
120
```

```
1 • SELECT * FROM healthy_sports_club.class;
```

	class_code	class_name	instructor_id	day_of_week	time_of_day	max_participants
▶ 1		Yoga Class	101	Monday	10:00:00	20
2		Zumba Class	102	Tuesday	15:30:00	25
3		Pilates Class	103	Wednesday	11:00:00	18
4		Basketball Training	104	Thursday	17:00:00	15
5		Badminton Practice	105	Friday	14:00:00	12
6		Dance Workshop	106	Saturday	12:30:00	30
7		Kickboxing Class	107	Sunday	16:00:00	20
8		Aerobics Session	108	Monday	18:00:00	25
9		Tai Chi Class	109	Tuesday	09:30:00	15
10		Volleyball Practice	110	Wednesday	19:00:00	10
*	NULL	NULL	NULL	NULL	NULL	NULL

Figure 11: Table: Class

(Source: Implemented by using MySQL Workbench)

The given code loads sample data in Class table into the database. Class titles, instructor IDs, course codes, day of the week, time of day, and maximum number of participants are all given in the rows that represent the individual classes. This information will be put in the Class table in order to facilitate class registration via the website.

4.5 Instructor

```
40      -- Table: Instructor
41 • CREATE TABLE Instructor (
42     instructor_id INT PRIMARY KEY,
43     staff_number INT UNIQUE,
44     FOREIGN KEY (staff_number) REFERENCES Staff(staff_number)
45 );
```

Figure 12: Create table: Instructor

(Source: Implemented by using MySQL Workbench)

The code configures the "Instructor" table with columns of `instructor_id` as the primary key and `staff_number` identical. The `staff_number` column is linked the `staff_number` column in the `Staff` table by a foreign key constraint, and this ensures data integrity. This table involving

instructors related to staff members is facilitating the identification and tracking of instructors amongst the system.

```

121  -- Inserting values into the Instructor table
122  • INSERT INTO Instructor (instructor_id, staff_number)
123  VALUES
124      (1, 1),
125      (2, 2),
126      (3, 3),
127      (4, 4),
128      (5, 5),
129      (6, 6),
130      (7, 7),
131      (8, 8),
132      (9, 9),
133      (10, 10);

```

```

1 • SELECT * FROM healthy_sports_club.instructor;

```

	instructor_id	staff_number
▶	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	10	10
•	NULL	NULL

Figure 13: Table: Instructor

(Source: Implemented by using MySQL Workbench)

The provided SQL command inserts values into the Instructor table and allocates (staff_numbr) unique instructor_id values to corresponding references. This, each team will be headed by a pair of instructors-staff, in which staff members will play the role of instructors in the sports club. The code in question assigns every employee their correct instructor role in order for the system to function properly.

4.6 Booking

```
47      -- Table: Booking
48 • CREATE TABLE Booking (
49     booking_id INT PRIMARY KEY,
50     member_membership_number INT,
51     facility_id INT,
52     class_code INT,
53     booking_date DATE,
54     start_time TIME,
55     end_time TIME,
56     booking_type ENUM('Facility', 'Class') NOT NULL,
57     FOREIGN KEY (member_membership_number) REFERENCES Member(membership_number),
58     FOREIGN KEY (facility_id) REFERENCES Facility(facility_id),
59     FOREIGN KEY (class_code) REFERENCES Class(class_code)
60 );
```

Figure 14: Create table: Booking

(Source: Implemented by using MySQL Workbench)

The proposed SQL code results in a table 'Booking' which stores the bookings made by members of the HEALTHY sports club at club. It includes features like booking_id, member_membership_number (shared with the members who're booking), facility_id or class_code (based on the type of booking), booking_date, start_time, end_time, and booking_type (for facility or class bookings). The foreign key relationships between the tables Member, Facility, and Class are created at this point to maintain data relationships and referential integrity in the database.

```
135      -- Inserting sample values into the Booking table
136 • INSERT INTO Booking (booking_id, member_membership_number, facility_id, class_code, booking_date, start_time, end_time, booking_type)
137     VALUES
138     (1, 1, 1, 2, '2024-05-01', '10:00:00', '12:00:00', 'Facility'),
139     (2, 2, 2, 3, '2024-05-02', '15:00:00', '17:00:00', 'Class'),
140     (3, 3, 3, 4, '2024-05-03', '09:00:00', '11:00:00', 'Facility'),
141     (4, 4, 4, 5, '2024-05-04', '14:00:00', '16:00:00', 'Class'),
142     (5, 5, 5, 6, '2024-05-05', '16:00:00', '18:00:00', 'Facility'),
143     (6, 6, 6, 7, '2024-05-06', '11:00:00', '13:00:00', 'Class'),
144     (7, 7, 7, 8, '2024-05-07', '12:00:00', '14:00:00', 'Facility'),
145     (8, 8, 8, 9, '2024-05-08', '13:00:00', '15:00:00', 'Class'),
146     (9, 9, 9, 10, '2024-05-09', '10:30:00', '12:30:00', 'Facility'),
147     (10, 10, 10, 1, '2024-05-10', '14:30:00', '16:30:00', 'Class');
148
```

```
1 • SELECT * FROM healthy_sports_club.booking;
```

	booking_id	member_membership_number	facility_id	class_code	booking_date	start_time	end_time	booking_type
▶	1	1	1	2	2024-05-01	10:00:00	12:00:00	Facility
	2	2	2	3	2024-05-02	15:00:00	17:00:00	Class
	3	3	3	4	2024-05-03	09:00:00	11:00:00	Facility
	4	4	4	5	2024-05-04	14:00:00	16:00:00	Class
	5	5	5	6	2024-05-05	16:00:00	18:00:00	Facility
	6	6	6	7	2024-05-06	11:00:00	13:00:00	Class
	7	7	7	8	2024-05-07	12:00:00	14:00:00	Facility
	8	8	8	9	2024-05-08	13:00:00	15:00:00	Class
	9	9	9	10	2024-05-09	10:30:00	12:30:00	Facility
	10	10	10	1	2024-05-10	14:30:00	16:30:00	Class
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 15: Table: Booking

(Source: Implemented by using MySQL Workbench)

The SQL statement given below executes the loading of example booking data into the Booking table. In each line data records a booking entry with its specific properties including booking_id, membership_number_members, facility_id or class_code (depending on the reservation type), booking_date, start_time, end_time, and booking_type (Facility or Class).

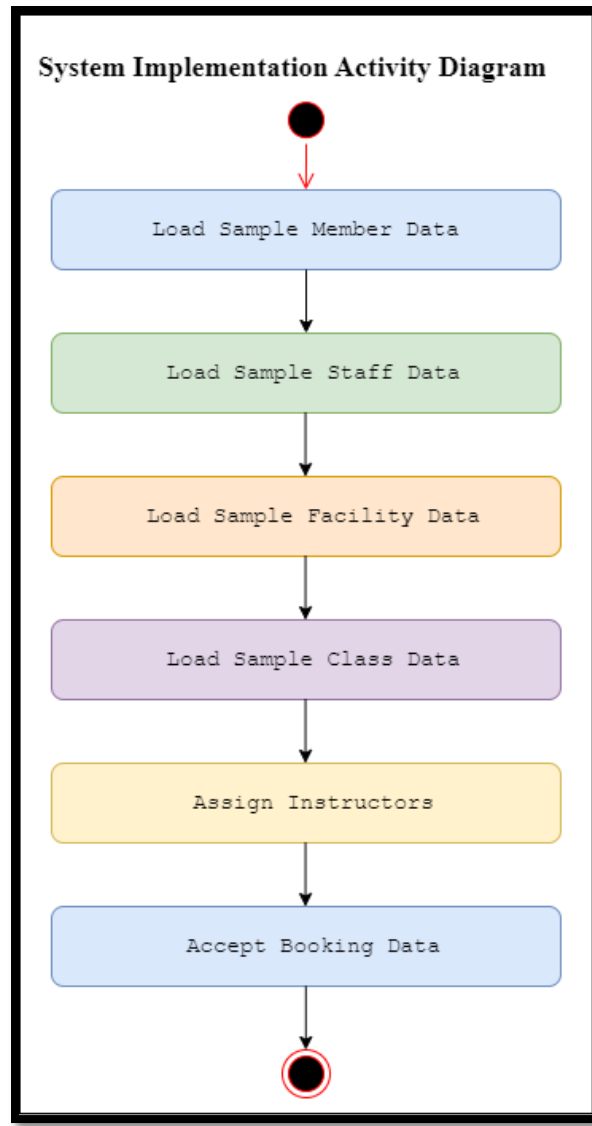


Figure 16: System Implementation Activity Diagram

(Source: Self-created)

Those entries act out like bookings that are supposed to be made by club members for different events within a specified period of time and on particular dates. The bookings range from the facility and class reservations though the entire session, showing different options members have.

5. Testing and Deployment

5.1 Inserting a new member into the Member table

```

151  -- Inserting a new member into the Member table
152  • INSERT INTO Member (membership_number, first_name, surname, address, telephone_number, email_address, date_of_birth, medical_conditions, member
153  VALUES
154  (11, 'Robert', 'Johnson', '789 Elm Rd, Village, Country', '555-999-8888', 'robert.johnson@example.com', '1993-08-15', 'None', 'Silver');
155

```

membership_number	first_name	surname	address	telephone_number	email_address	date_of_birth	medical_conditions	membership_type
1	John	Doe	123 Main St, City, Country	123-456-7890	john.doe@example.com	1990-05-15	None	Gold
2	Jane	Smith	456 Oak Ave, Town, Country	987-654-3210	jane.smith@example.com	1985-09-20	Allergies: peanuts	Silver
3	David	Johnson	789 Elm Rd, Village, Country	555-123-4567	david.johnson@example.com	1988-12-10	None	Bronze
4	Emily	Brown	321 Pine Blvd, City, Country	222-333-4444	emily.brown@example.com	1995-03-25	Asthma	Gold
5	Michael	Wilson	567 Cedar St, Town, Country	444-555-6666	michael.wilson@example.com	1992-07-08	None	Silver
6	Sarah	Miller	890 Birch Ave, Village, Country	777-888-9999	sarah.miller@example.com	1983-11-30	Diabetes	Bronze
7	Alex	Lee	234 Oakwood Dr, City, Country	111-222-3333	alex.lee@example.com	1998-09-12	None	Gold
8	Jessica	Chang	678 Maple Rd, Town, Country	999-000-1111	jessica.chang@example.com	1991-04-18	None	Silver
9	Kevin	Garcia	901 Pinecrest Ave, Village, Country	333-666-9999	kevin.garcia@example.com	1987-02-05	High blood pressure	Bronze
10	Laura	Martinez	345 Cedarwood Ln, City, Country	555-777-8888	laura.martinez@example.com	1994-06-22	None	Gold
11	Robert	Johnson	789 Elm Rd, Village, Country	555-999-8888	robert.johnson@example.com	1993-08-15	None	Silver
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 17: Inserting a new member into the Member table

(Source: Implemented by using MySQL Workbench)

The provided SQL code snippet establishes new position for the Member table in the HEALTHY the database of the sports club's database with Robert Johnson as the new member. The details of Robert may possibly be unique membership number (11), first name (Robert), surname (Johnson), address (789 Elm Rd, Village, Country), telephone number (555-999-8888), email address(robert.johnson@example.com), date of birth (1993-08-15), reported This information is inserted to membership database to ensure that have a full record of Robert's membership and can fulfil duties as the club's membership body.

5.2 Book a Class for a Club Member

```

180  -- Confirm the booking
181  • SELECT *
182  FROM Booking
183  WHERE member_membership_number = 1 -- Adjust with the actual member's membership_number
184  AND class_code = 2
185  AND booking_type = 'Class';
186

```

class_code	class_name	instructor_id	day_of_week	time_of_day	max_participants
2	Zumba Class	102	Tuesday	15:30:00	25
NULL	NULL	NULL	NULL	NULL	NULL

Figure 18: Book a Class for a Club Member

(Source: Implemented by using MySQL Workbench)




The example of the code exemplifies the actions of the club member indulging in class booking. This process starts with picking details of a particular class by using its class code through which the students get their admission confirmation at the institution. After this, a new record for the booking table is added which holds all the information of the member's membership number, class code, booking date, start time, end time, and booking type (as soon as 'Class' works in this case). This update then runs through the algorithm edit that counts current_participants and changes it to account for this new booking. Lastly, by using the specified member, the class code, and the type of booking ('Class') as parameters it will end by grabbing the confirmations of the booking and the booking details.

5.3 Inserting attendance records

```

220 FROM
221     Attendance A
222 JOIN
223     Member M ON A.member_membership_number = M.membership_number
224 JOIN
225     Booking B ON A.booking_id = B.booking_id
226 WHERE
227     B.class_code = 1; -- Replace '1' with the desired class_code to check attendance

```

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

	first_name	surname	attendance_date	attended
▶	Laura	Martinez	2024-05-10	1

Figure 19: Inserting attendance records for a Club Member

(Source: Implemented by using MySQL Workbench)

The record specifies the "Attendance" table with fields like attendance_id, booking_id, member_membership_number, attendance_date and attended, for that way. L'information prétente se trouve dans la colonne booking_id, qui se réfère au numéro de l'inscription du club avec membre_membership_number de la base de données et de la table, respectivement. Attending the sample records are included represented in either case member has been present at previous book (Guagliano *et al.* 2024). The question returns a specific class' attendance detail through a spatial join that commonly links the "Attendance" table with other tables at "Member"

and "Booking" to show members' names, participation dates, and the attendance status in the class_code.

6. Conclusion

6.1 Key findings and conclusions

Based on the comprehensive analysis of the case study and its associated results, main conclusion is the efficient management solution being integrated data system containing registering members, staff administration, facility booking, class schedules, and attendance tracking. System should handle efficiently membership levels, block hour periods, and people visits. In other words, the implementation of these changes will make club more effective in its operation, make more enjoyable experiences for members and just make the resources of the club to be used optimally at the "HEALTHY" sports club.

6.2 The recommendations for future work on the system

Based on the case study and the system analysis results, here are recommendations for future work on the system:

- **Enhanced Membership Management:** Having a more comprehensive membership management system as a feature with the ability to provide automated renewal reminders and type of privileges individualization according to each membership tier is what needed to be created in this feature.
- **Improved Booking Rules:** Upgrade booking controls to cover abstract conditions that may evolve such as merging time categories and average out the selection of facilities and classes in different periods.
- **Mobile Application Development:** Develop mobile application that will be dedicated for members with easy and simple access to manage booking, schedules, and notification.
- **Analytics and Reporting:** Embed analytics and reporting features that use members' engagement metrics, the number of popular classes, and facility's use patterns thus allowing to optimize their service accordingly.
- **Instructor Management Tools:** Develop teacher decentralized control schedules, attendance records, and performance appraisal tools to put in place.

List of the References

Blomqvist Mickelsson, T., 2024. Sports clubs' role in community capacity development—illustrations from the Swedish sports movement. *Community Development Journal*, p.bsad039.

Guagliano, J.M., Hliounakis, H., El Masri, A., Kolt, G.S. and George, E.S., 2024. Determining priority areas for health promotion amongst Australian professional sporting organisations. *Journal of Science and Medicine in Sport*, 27(1), pp.30-36.

Hu, J., Ye, X. and Gu, S., 2024. The impact of subjective consumer knowledge on consumer behavioral loyalty through psychological involvement and perceived service quality: sports clubs. *Asia Pacific Journal of Marketing and Logistics*.

Kim, M., Oja, B.D., Nite, C. and Anagnostopoulos, C., 2024. Expanding sport management toward management through sport: the promise of company sport in organizations. *Sport Management Review*, pp.1-22.

Kuo, N.I.H., Perez-Concha, O., Hanly, M., Mnatzaganian, E., Hao, B., Di Sipio, M., Yu, G., Vanjara, J., Valerie, I.C., de Oliveira Costa, J. and Churches, T., 2024. Enriching Data Science and Health Care Education: Application and Impact of Synthetic Data Sets Through the Health Gym Project. *JMIR Medical Education*, 10, p.e51388.

Liddelow, C., Schweickle, M.J., Sutcliffe, J.T., Swann, C., Keegan, R., Rice, S., Okely, A. and Vella, S.A., 2024. Defining the scope and content of mental health guidelines for community sport in Australia: A Delphi study. *Psychology of sport and exercise*, 70, p.102553.

Mickelsson, T.B., 2024. Facilitating migrant youths' inclusion into Swedish sport clubs in underserved areas. *Nordic Social Work Research*, 14(1), pp.121-135.

Yang, X., Lu, H., Cai, J. and Zhang, S., Constructing a social responsibility system for professional sports clubs based on the perspective of China. *Business Ethics, the Environment & Responsibility*.