

# **CS 5200 - Database Management Systems**

Project Report for

## **GYM MANAGEMENT SYSTEM**

Group Name: KulkarniNMagdumT

Group Members: Nikhil Kulkarni, Tanvi Magdum

### **1. INTRODUCTION**

Gold's Gym has a chain of gyms all around United States. A gym needs to keep track of its members, training staff, equipment, rooms, activities and the equipment cleaning staff. This gym will then be able to use the data in this portal to do a quick analysis on the data provided. The gym also needs to keep track of all the equipment and do the maintenance for the same. The database also contains the rooms every branch of the gym has and what activities take place in it.

The manager adds, updates or deletes the member, trainer and cleaning staff information. Moreover, the manager can add or remove the activities conducted in the gym or the list of equipment owned by the gym.

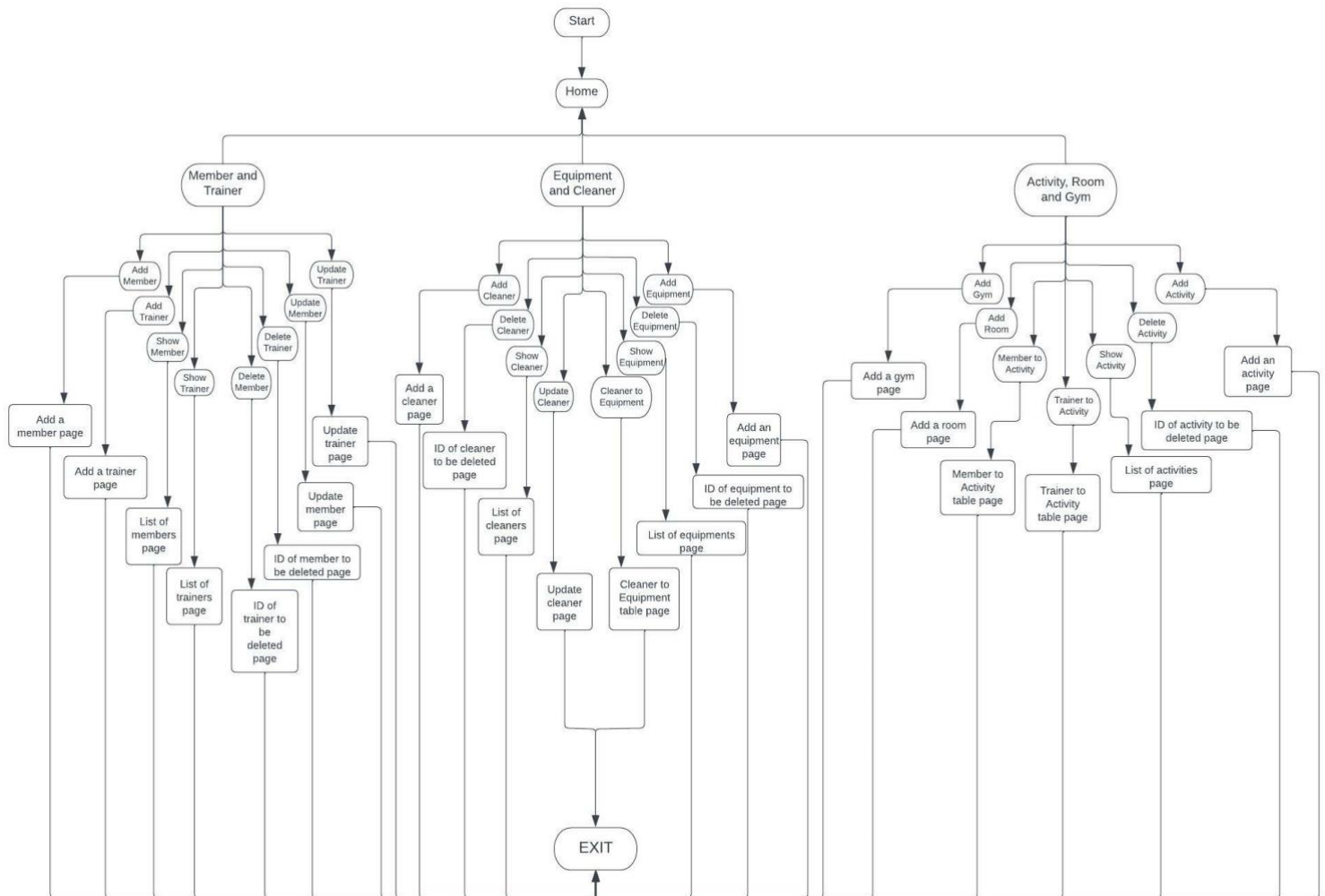
The manager can perform search operation on various entities by inputting various parameters.

### **2. RATIONALE**

Recently we faced a pandemic which has affected the healthy life of numerous people. In today's world where the data is produced at such an enormous rate, we can make use of such appropriate data, analyze and take steps to improve people's health. Hence, there is a need of Gym Management System for every gym through which, the data can be used to analyze what percent of people in a certain area attends gym to maintain their healthy lives and what all exercises they perform to stay fit.

### 3. ACTIVITY DIAGRAM

#### 3.1 DIAGRAM



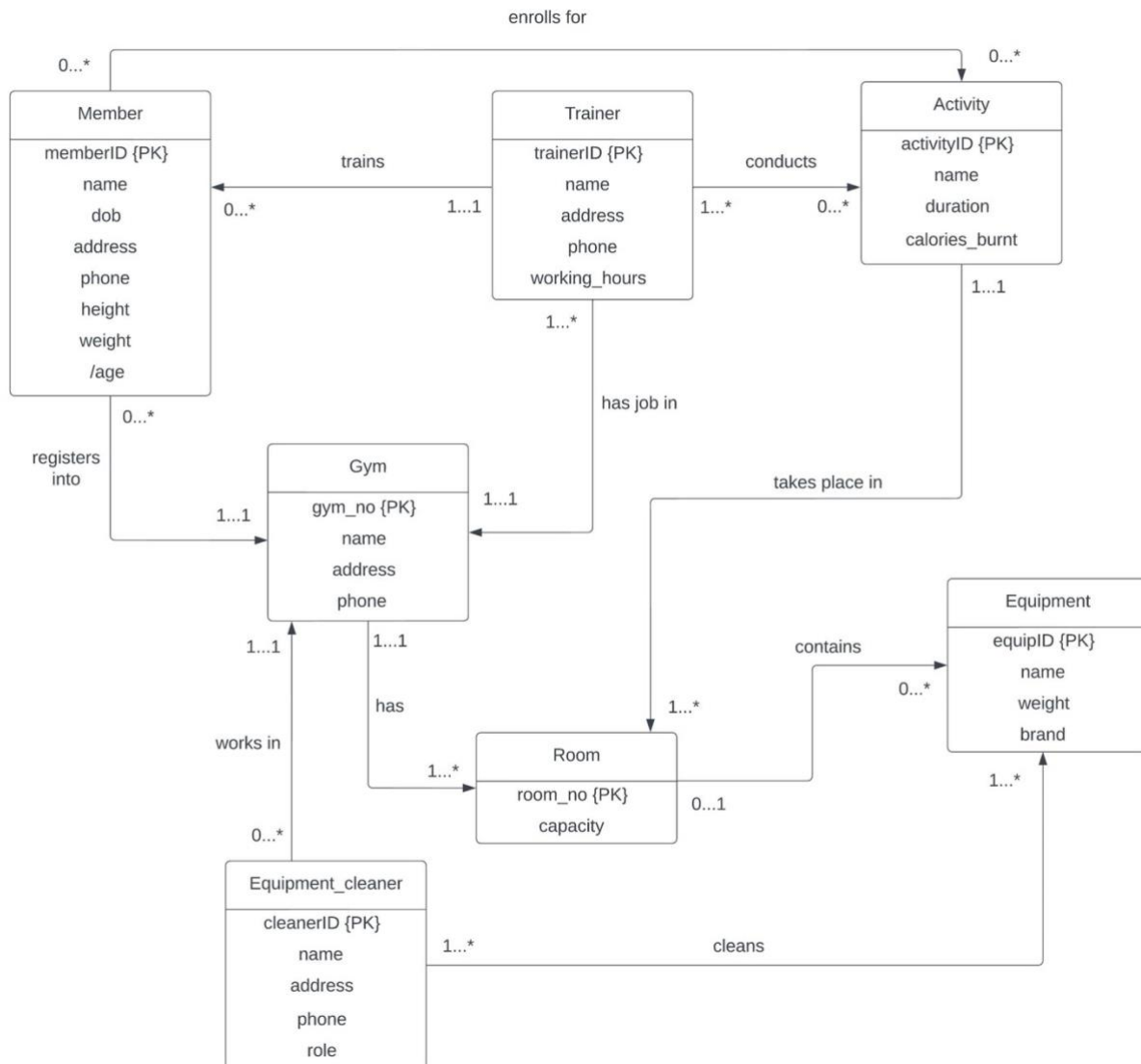
#### 3.2 DESCRIPTION

Every entity has a CREATE operation. End user can create records for some of the entities like member, trainer, activity, equipment, cleaner and gym, room. Similarly, few entities can be updated by manager, and others like equipment, room, activity, gym cannot be updated. There are various filters used to search member, trainer, equipment, cleaner and activity. Manager can delete member, trainer, equipment, cleaner and activity. Gym and room cannot be deleted, once created.

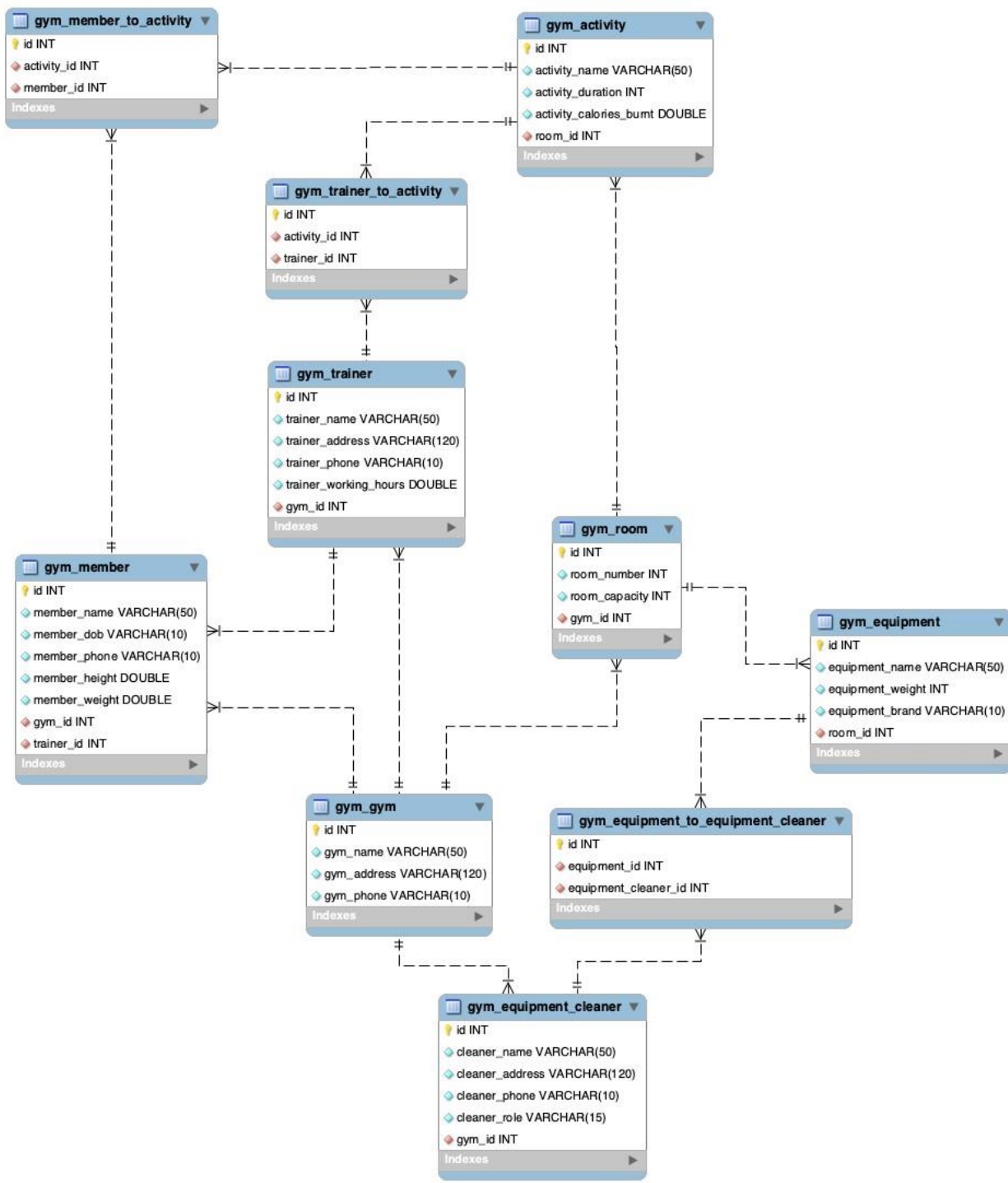
We have provided buttons for all the operations. There are 3 main buttons on Home, and sub-buttons when user clicks on any of the 3 buttons. Once user chooses the operation to perform, he'll be asked for the relevant information to input and a button to save that information (for create, update), delete record (for

delete) and show information (for viewing records). Every sub-page has a Home page. So user can navigate to Home anytime.

#### 4. UML DIAGRAM



5. LOGICAL DESIGN



## 6. TECHNICAL DESCRIPTION

### 6.1 README

#### 6.1.1 TO INSTALL:

Go to the directory of files of source code and install following things with given links.

1. Python version 3.8.8 - [Link](#) Confirm this by typing “python –version” in command prompt.
2. Pip version 20.2.3 - [Link](#) Confirm this by typing “pip –version” in command prompt.

If everything goes smoothly your command prompt will look like –

```
C:\Users\Nikhil\OneDrive\Desktop\DBMS-Mini-Project-main>python --version
Python 3.8.8

C:\Users\Nikhil\OneDrive\Desktop\DBMS-Mini-Project-main>pip --version
pip 20.2.3 from c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages\pip (python 3.8)

C:\Users\Nikhil\OneDrive\Desktop\DBMS-Mini-Project-main>
```

3. MySQL and workbench version – 8.0.30 [Link](#) (You may have already installed this.)
4. Go to the folder containing all files where you can see the requirements.txt file and hit the command “pip install -r requirements.txt” in command prompt.

If you run this command again and everything is installed smoothly your command prompt will look like –

```
C:\Users\Nikhil\OneDrive\Desktop\DBMS-Mini-Project-main>pip install -r requirements.txt
Requirement already satisfied: Django==4.1.3 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from -r requirements.txt (line 1)) (4.1.3)
Requirement already satisfied: pandas==1.4.2 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from -r requirements.txt (line 2)) (1.4.2)
Requirement already satisfied: plotly==5.11.0 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from -r requirements.txt (line 3)) (5.11.0)
Requirement already satisfied: tzdata; sys_platform == "win32" in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from Django==4.1.3->-r requirements.txt (line 1)) (2022.1)
Requirement already satisfied: asgiref<4,>=3.5.2 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from Django==4.1.3->-r requirements.txt (line 1)) (3.5.2)
Requirement already satisfied: backports.zoneinfo; python_version < "3.9" in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from Django==4.1.3->-r requirements.txt (line 1)) (0.2.1)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from Django==4.1.3->-r requirements.txt (line 1)) (0.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from pandas==1.4.2->-r requirements.txt (line 2)) (2.8.2)
Requirement already satisfied: numpy>=1.18.5; platform_machine != "aarch64" and platform_machine != "arm64" and python_version < "3.10" in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from pandas==1.4.2->-r requirements.txt (line 2)) (1.22.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from pandas==1.4.2->-r requirements.txt (line 2)) (2022.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from plotly==5.11.0->-r requirements.txt (line 3)) (8.1.0)
Requirement already satisfied: six>=1.5 in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (from python-dateutil>=2.8.1->pandas==1.4.2->-r requirements.txt (line 2)) (1.15.0)
```

5. Lastly, we need to install mysqlclient 2.1.1 for python hence, hit “pip install mysqlclient” command in the command prompt in the same directory.

If you run this command again your command prompt will look like -

```
C:\Users\Nikhil\OneDrive\Desktop\DBMS-Mini-Project-main>pip install mysqlclient
Requirement already satisfied: mysqlclient in c:\users\nikhil\appdata\local\programs\python\python38\lib\site-packages (2.1.1)
```

## 6.2.2 HOW TO RUN:

1. Open the workbench and import the given dump from self-contained file. (Make sure the port is 3306 and localhost). This will ensure all the tables and procedures are present in the schema. Make sure your server is always up. Use schema inspector to see following results:

Your schema will have tables like –

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length
auth_group	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
auth_group_permissions	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
auth_permission	InnoDB	10	Dynamic	64	256	16.0 KiB	0.0 bytes
auth_user	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
auth_user_groups	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
auth_user_user_permissions	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
django_admin_log	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
django_content_type	InnoDB	10	Dynamic	16	1024	16.0 KiB	0.0 bytes
django_migrations	InnoDB	10	Dynamic	22	744	16.0 KiB	0.0 bytes
django_session	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
gym_activity	InnoDB	10	Dynamic	6	2730	16.0 KiB	0.0 bytes
gym_equipment	InnoDB	10	Dynamic	17	963	16.0 KiB	0.0 bytes
gym_equipment_cleaner	InnoDB	10	Dynamic	9	1820	16.0 KiB	0.0 bytes
gym_equipment_to_equip...	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
gym_gym	InnoDB	10	Dynamic	5	3276	16.0 KiB	0.0 bytes
gym_member	InnoDB	10	Dynamic	12	1365	16.0 KiB	0.0 bytes
gym_member_to_activity	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes
gym_room	InnoDB	10	Dynamic	10	1638	16.0 KiB	0.0 bytes
gym_trainer	InnoDB	10	Dynamic	15	1092	16.0 KiB	0.0 bytes
gym_trainer_to_activity	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes

Your schema will have procedure like –

Name	Type	Definer	Modified	Created	Security Type	Client Charact
create_activity	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_equipment	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_equipment_cleaner	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_eq_to_eqcl	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_gym	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_member	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_mem_to_act	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_room	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_trainer	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
create_tr_to_act	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
delete_activity	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
delete_equipment	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
delete_equipment_cleaner	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
delete_gym	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
delete_member	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
delete_trainer	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_activity_by_gym	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_equipmentCleaners_by_role	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_equipment_by_gym	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_equipment_by_room	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_equipment_cleaner_by_name	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_gym_by_name	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_member_by_gym	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_member_by_name	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_member_by_trainer	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_trainer_by_gym	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
read_trainer_by_name	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4
update_activity	PROCEDURE	root@localhost	2022-12-09 10:2...	2022-12-09 10:2...	DEFINER	utf8mb4

2. Run the create\_user.sql file given in directory using mysql workbench. This will create DB user and password which is saved in the application.

Run the following script in sql to confirm if user name “dbadmin” has been created.

Script – “use mysql;

SELECT user from user;”

Result –

	user
▶	dbadmin
	mysql.infoschema
	mysql.session
	mysql.sys
	root

3. After this just hit the command in command prompt “python manage.py runserver” and you can see the website on <http://127.0.0.1:8000/>.

If everything goes smoothly your command prompt should look like –

```
System check identified 10 issues (0 silenced).
December 09, 2022 - 10:49:27
Django version 4.1.3, using settings 'DBMS_Mini_Project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

### 6.2.3. NOTES:

Please create the records in following order –

1. Gym records
2. Trainers
3. Rooms
4. Members
5. Other entity records

Unless and until there are records for gym, room and trainer, user won't be able to create other entities which are dependent on them.

On member creation page, please select Gym first, then only relevant Trainers would appear. Similar for Trainer to Activity creation and Cleaner to Equipment creation.

## **6.2 SOFTWARES USED**

### **6.2.1 MySQL Database**

We have used MySQL as our database as it is fast and reliable. MySQL could be considered as good option for small scale businesses that don't have a large, sophisticated data team, like in our case. It has simple and low-cost setup process and a widespread support of huge community.

### **6.2.2 Python and Django Framework**

#### **6.2.2.1 Python**

We have Python as medium to interact with the data as a user. Python is a user-friendly language with readable syntax and that makes it well-loved by both seasoned developers and experimental students. The simplicity of Python means that developers can focus on building the correct database rather than spending all their time understanding just the technical nuances of the language. It has hundreds of different libraries and frameworks that can be used by developers.

#### **6.2.2.2 Django Framework**

The Django framework is used to achieve our requirements. Django is a high-level Python web framework which ensure rapid development with clean and practical design. It has many inbuilt features, and it is very versatile, compatible with majority of databases. It also makes easier to build webpages using Python by its built-in HTML templating, URL routing, object-relational mapping, and session management, helping developers avoid the vexing search for third-party tools. Also, Django makes it possible to migrate from one database to another and perform common operations without having to write much additional code. In our application, user interacts with the web-interface (GUI) and MySQL runs the queries and generate data.

### **6.2.3 HTML5, CSS**

The GUI is built using HTML and CSS. CSS describes how HTML elements are to be displayed on screen, paper or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. We have added and modified the HTML templates given by Django with the relevant data that user needs access to.



### **6.3 SCHEMA DESCRIPTION**

The above schema design describes the whole structure of the Gym Management System. There are mainly 7 entities – gym\_gym, gym\_member, gym\_trainer, gym\_activity, gym\_room, gym\_equipment, gym\_equipment\_cleaner. There are 3 intermediary entities – gym\_member\_to\_activity, gym\_trainer\_to\_activity, gym\_equipment\_to\_equipment\_cleaner. These 3 entities are created since there is a many-to-many relationship between member to activity, trainer to activity and equipment to equipment cleaner. All the tables are in 3NF. Every entity has a PRIMARY KEY as entity ID and foreign keys depending on their relationships with other entities. Also, there are integrity constraints such as action to be performed for foreign keys ON DELETE, ON UPDATE clauses within the foreign keys in the tables. Mostly, all the primary key fields are IDs and they are auto-incremented.

### **6.4 PROCEDURES**

For every operation, there is a procedure built in MySQL. Each procedure contains error handling. If any of the input fields is null, the procedure throws an error. Also, if there is an SQLException, there is an exit handler which jumps out of the procedure when this exception is thrown.

The procedures for CREATE, UPDATE, DELETE and READ are defined for various entities. We have used TABLE JOINS and SUB-QUERIES to extract the desired data. During creation, we have used INSERT operation to insert tuples in the table, UPDATE for updating and DELETE for deletion of tuples. The tests for PROCEDURE CALLS are included in the SQL script.

The stored procedures execute only within the database but can be invoked from both inside the database (as part of a trigger, for example) and outside the database from connected software, like a web or desktop application. In our application, we invoke the procedures from Python script.

### **6.5 EXTRA WORK DONE**

1. We have built web application as GUI using Django framework.
2. We have also built complex queries using table joins and sub-queries for data extraction.
3. We have built create operation for total 9 entities, update for 3 entities, read for 5 entities and delete operation for 5 entities.

## **8. LESSONS LEARNT**

### **8.1 TECHNICAL EXPERTISE GAINED**

As we have used MySQL as the database, Python as programming language, Django framework and HTML/CSS to build GUI, we can say we have a hold over these skills now. In MySQL, we implemented creation, update, delete and various queries for data extraction. We have used the Stored Procedures and Error handlers and integrated these procedures with Python to interact with the user.

Right from creating the UML, to implementing the actual schema, there are numerous 'what-ifs' and 'what-if not's' to be considered. As our application – 'Gym Management System' is a real-world application, before developing we had to think that as a user what all functionalities, I would expect in such an application, what all wrong inputs I might enter, and what response I expect from the system in such cases.

We had acquired skills like MySQL on a scale of 4.5/5 and Python with Django on a scale of 4/5. We have gained an experience of how it is to work on a real-world application, starting from scratch and building it in a specific timeline along with coordinating with the team.

### **8.2 INSIGHTS**

Time management is the key factor in making a project successful. We understand that if we do not complete the requirements within the project deadline there would be a penalty for us, but when we go out there in corporate industry, not fulfilling the client expectations within the timelines would cost a great loss for the company. Though as developers, we can be realistic and propose a filtered list of the things to client which can be achieved in the given timeframe, but we cannot turn back in the middle of the development (or project release). Through this academic project, we tried to cover as much as possible that we had originally proposed, but due to time and resource constraints we couldn't achieve it to the fullest. This makes us realize that it is crucial to analyze the time required for all the functionalities to implement before starting the development and have a back-up plan ready, if something goes wrong unexpectedly.

### **8.3 COMTEMPLATED APPROACHES**

Initially, we proposed a design which included the User Login. But when we started building the project, we realized that basically there is no need for separate user logins, since the manager/gym owner is the only one who would be accessing the gym data. We don't give access to members, trainers or cleaners to this management system. We did not find a strong rationale to ask for user logins in our case, so we disregarded that approach.

## **9. FUTURE SCOPE**

### **9.1 PLANNED USE OF DATABASE**

In future, we plan to optimize the performance of our database to give the exact information required by the user. We plan to add more data extraction procedures, make these procedures return the exact numeric data, for ex. How many activities are linked to a specific room or how many members are trained by a particular trainer. In this way we can reduce the quantity of data being sent to the user, unless user requests a detailed information. We are also securing the information by allowing the end user to access a generalized data, instead of detailed information of every person at the gym. This type of numeric data would be beneficial for Data Management companies (in reference to our rationale of implementing this project). Many such small-scale businesses could make money by selling this data to the Data companies.

### **9.2 POTENTIAL ADDITION OF FUNCTIONALITIES**

1. Addition of fields to the member table to track daily attendance, time of check-in and check-out, duration of total work-out in the gym.
2. Storing an image of each of the members, trainers and cleaners admitted in the gym by their consent.
3. Providing a save information functionality to the manager. This functionality would save the information of an entity in a pdf form on the desktop. For ex. Information of a specific equipment must be sent to the original equipment manufacturing company since the equipment is damaged. So, the manager would go on the website, search the equipment, click on download information as pdf and get the information ready to send it to the manufacturer.