



Software Engineering

FINAL PROJECT SPRING 2023

SUPERVISED BY

Prof.Dr / Youssry Taha

TA/ Hashim Hossam

‘System for gym management’

Presented By

NAME	ID
Zeyad Waheed	6797

Introduction

The Gym Management System is a comprehensive software application designed to streamline the management and administration of a gym or fitness center. The system is specifically tailored to cater to the needs of gym administrators, providing them with powerful tools and functionalities to efficiently manage various aspects of the gym operations.

With a web-based architecture and a three-tier design, the Gym Management System offers a user-friendly interface accessible through web browsers. It is important to note that the system is restricted to only authorized administrators and is not accessible to clients. This restriction ensures that the sensitive information and administrative functionalities of the gym remain secure and limited to designated personnel.

The Gym Management System empowers administrators with a range of features to effectively manage clients, trainers, packages and billing. It provides a centralized platform for administrators to handle critical tasks such as client enrollment, trainer assignments, package creation and management, appointment scheduling, and financial transactions.

By utilizing the three-tier architecture, the system achieves a clear separation of concerns. The presentation layer delivers a seamless and intuitive user interface, while the business logic layer handles the processing of client requests, enforces business rules, and facilitates secure access to the system. The data access layer ensures efficient interaction with the underlying database, storing and retrieving essential information related to clients, trainers, packages, and other relevant entities.

The Gym Management System prioritizes scalability, allowing administrators to handle varying levels of gym activity without compromising system performance. It is designed to adapt to peak hours or increased workload by independently scaling each layer of the architecture as needed.

In conclusion, the Gym Management System offers a powerful and secure solution for gym administrators to effectively manage and administer all aspects of their operations. By leveraging a web-based architecture and a three-tier design, the system provides a user-friendly interface, scalability, and robust security measures. Through the integration of comprehensive features and functionalities, the Gym Management System aims to optimize the efficiency and productivity of gym management while ensuring the confidentiality and integrity of the gym's data.

Glossary of Terms

- **System for Gym Management:** A software system designed to handle the management and administration of a gym or fitness center, including tasks such as client management, trainer management, package management and offers of payments .
- **Member:** A person who has registered or enrolled in the gym or fitness center to avail its services. The client may have various attributes such as name(first name-last name), email, membership ID, and Trainer.
- **Trainer:** A fitness professional who provides training and guidance to clients. Trainers may have attributes such as name, Trainer ID, Phone Number.
- **Package:** A pre-defined set of services or facilities offered by the gym, typically categorized into different tiers or levels. Packages may include details such as Package ID, Package Name and Amount.
- **User Interface:** The visual and interactive components of the system that allow Admin to interact with the Gym Management System. This can include web pages, forms, menus, and other graphical elements.
- **Business Logic:** The layer of the system that handles the processing and manipulation of data, as well as the implementation of business rules and logic. This layer performs tasks such as calculating Payments ,Displaying Package Details, Member Details and Trainer Details.
- **Data Access Layer:** The layer of the system responsible for interacting with the underlying database or data storage system. It handles tasks such as retrieving and storing client and trainer information, managing packages, and maintaining data integrity.
- **Database:** A structured collection of data that stores information related to clients, trainers, packages, appointments, and other relevant entities. The database is accessed and manipulated through queries and operations performed by the system.
- **Authentication:** The process of verifying the identity of a user or entity accessing the Gym Management System. This ensures that only authorized individuals can access and perform actions within the system.
- **Authorization:** The process of granting or denying access to specific features or data within the Gym Management System based on the roles and permissions assigned to individual users. It ensures that users can only access the functionalities they are allowed to use.
- **Payment Processing:** The functionality within the system that handles financial transactions related to client payments, such as membership fees, package purchases, or additional services. It may involve integration with payment gateways or third-party payment processors.

Requirements

User requirements

the process of identifying and documenting the needs and expectations of users for a particular software system or product. This process involves understanding the perspective of the users, their goals, tasks, and preferences.

It is divided into two categories functional requirements and non functional requirements.

Functional requirements

- User shall login into the system as an admin and confirm his identity before accessing any features or functionalities.
- User shall be able to add new trainers.
- User shall be able to add new members.
- User shall be able to check trainer details.
- User shall be able to check member details.
- User shall be able to search for a specific member.
- User shall be able to choose a specific trainer for each member.
- User shall be able to add each new payment to the system with a specific id.
- User shall be able to monitor the member membership status.

Non functional requirements

- Security
- High efficiency
- Easy to use
- Reliability
- Usability
- Dependability

System requirements

System requirements are important to consider during the design and development phase of a system, as they define the boundaries and limitations of what the system can and cannot do.

System requirements are often documented in a requirements specification or similar document, which outlines the functional and non-functional requirements of the system.

Functional requirements

- The system should have a login feature that verifies the manager's identity by requesting data from the database.
- Users can add new members by filling out a form includes member's name ,member's email and member's phone number and the system will add this information as a tuple to the database.
- Users can add new trainers by filling out a form includes trainer's name ,trainer's ID and trainer's phone number, and the system will add this information as a tuple to the database.
- Users can search for members by member ID or name and select the piece of information they want.
- Users can create a new membership by filling out a form with information specified in the database schema.
- Users can view packages corresponding to the duration of each membership.
- User should be able to logout from the system.

Non-functional requirements:

- The system should be user-friendly, and it should take little time for the user to learn the system and all its features.
- The system must have a fast and efficient restart time in the event of a failure.
- The system must be reliable and stable.
- The system must have a fast response time, with minimal lag or delays, to ensure that users can access and update data quickly and efficiently.
- The system must be available to users at all times, with minimal downtime or maintenance windows, to ensure uninterrupted access to gym services and data.
- The system must comply with relevant laws and regulations governing data privacy, financial reporting, and other aspects of gym management.

Software Processes

The recommended software development process was an incremental-agile approach, wherein the specification, development, and validation processes are interlaced. This approach was selected for several reasons. Firstly, time was a critical factor, and the team aimed to release a functional version of the product as quickly as possible. Secondly, the team consisted of only three members, making an agile methodology the most practical choice.

The chosen approach ensures that each successive version of the product is usable and adds more functionality to the system. The development process involved four primary phases: requirements, design, implementation, and testing.

Due to the lack of a real customer and the project's simulated nature, the requirements document underwent minimal modifications during the development process. As a result, the team created the requirements document at the project's onset, and it remained relatively unchanged thereafter.

Additionally, testing was conducted on a subset of features rather than the entire system. This approach allowed the team to focus on the critical functionalities while minimizing the time and effort required for testing.

The team faced several assumptions and constraints throughout the project's lifecycle. Due to the simulated nature of the project, they made some assumptions that guided their implementation. Additionally, they considered several constraints, including budget, time, and team size, to ensure that the project adhered to the available resources.

Overall, the incremental-agile approach allowed the team to develop a working version of the GYM management system that met the project's requirements while working within the available resources. The approach's flexibility enabled the team to adjust to changes and modifications as required while ensuring the project's timely completion.

Architecture Design

One suggested system architecture for a Gym Management System would be a three-tier architecture, consisting of a presentation layer, a business logic layer, and a data access layer.

The **presentation layer** would provide the interface for the clients to interact with the system, such as a web or mobile application. This layer would handle the user input and display output to the user.

The **business logic layer** would handle the processing of the client requests and responses. It would handle the authentication and authorization of users, as well as perform any necessary business rules, such as calculating pricing based on packages or scheduling appointments with trainers.

The **data access layer** would be responsible for interacting with the database of trainers and client packages. It would handle the retrieval and storage of data in the database and ensure data integrity.

Reasons of Using This Architecture

- **Separation of concerns:** A Gym Management System typically involves a user interface for clients to interact with, business logic for handling client requests and responses, and data storage for managing clients, trainers, and packages. A three-tier architecture can provide a clear separation of concerns between these different layers, making the system easier to maintain and modify in the future.
- **Scalability:** A Gym Management System may experience spikes in usage during certain times of the day or year. With a three-tier architecture, each layer can be scaled independently, allowing the system to handle increased traffic or load by scaling up or down each layer as needed, making it easier to maintain and modify in the future. It also allows for scalability, as each layer can be scaled independently
- **Maintainability:** With a clear separation of concerns, each tier can be modified or updated without affecting the other tiers, making it easier to maintain and improve the system over time.
- **Security:** Separating the data storage tier from the user interface and application logic tiers can provide additional security by limiting direct access to the data and controlling access through the application logic layer.
- **Flexibility:** A three-tier architecture allows for flexibility in choosing the technologies and tools used for each tier, making it easier to adapt to changes in requirements or technology.
- **Modularity:** By breaking down the system into smaller, more manageable components, a three-tier architecture allows for greater modularity and code reusability.
- the three-tier architecture is widely used and has proven to be reliable and efficient for **web applications**. So it Would be our Suggested **Application Architecture**.

Overall, a three-tier architecture can provide several benefits for a Gym Management System, including scalability, maintainability, security, flexibility, and modularity.

Suggested Application Architecture

the suggested application architecture for the Gym Management System would be a **web-based architecture**.

The presentation layer would consist of a web application that would be accessible to the clients through a web browser or a mobile device (**on our case would be WEB BROWSER**). The web application would be responsible for displaying the user interface and handling user input and provide the required service to give the efficient output.

The business logic layer would be implemented using a web server, that would handle the processing of client requests and responses. The web server would interact with the data access layer to retrieve and store data in the database.

The data access layer would be implemented using a database management system, such as **MySQL** on our Project, that would handle the storage and retrieval of data. The data access layer would be responsible for ensuring data integrity and providing a secure and efficient interface to the web server.

Overall, this web-based application architecture would provide a scalable, reliable, and efficient solution for the Gym Management System, enabling clients to interact with the system from anywhere and at any time, while providing a secure and reliable platform for the business logic and data storage.

Reasons of using Web based – Application on our project

- **Accessibility:** A web-based system (**our Gym Management System Website**) can be accessed from anywhere with an internet connection, allowing clients and trainers to easily access the system from their computers or mobile devices.
- **User interface:** A web-based system allows for the development of a user-friendly and responsive user interface that can be easily accessed and used by clients and trainers
- **Platform independence:** A web-based system can be accessed from any platform or operating system, allowing for greater flexibility and accessibility.
- **Scalability:** A web-based system can be easily scaled up or down to handle changes in traffic or usage, allowing for greater scalability and performance
- **Easy maintenance:** A web-based system can be easily maintained and updated from a central location, reducing the need for manual updates on individual client or trainer devices.
- **Security:** A web-based system can be secured through the use of secure connections (such as HTTPS) and other security measures to protect sensitive data and prevent unauthorized access.

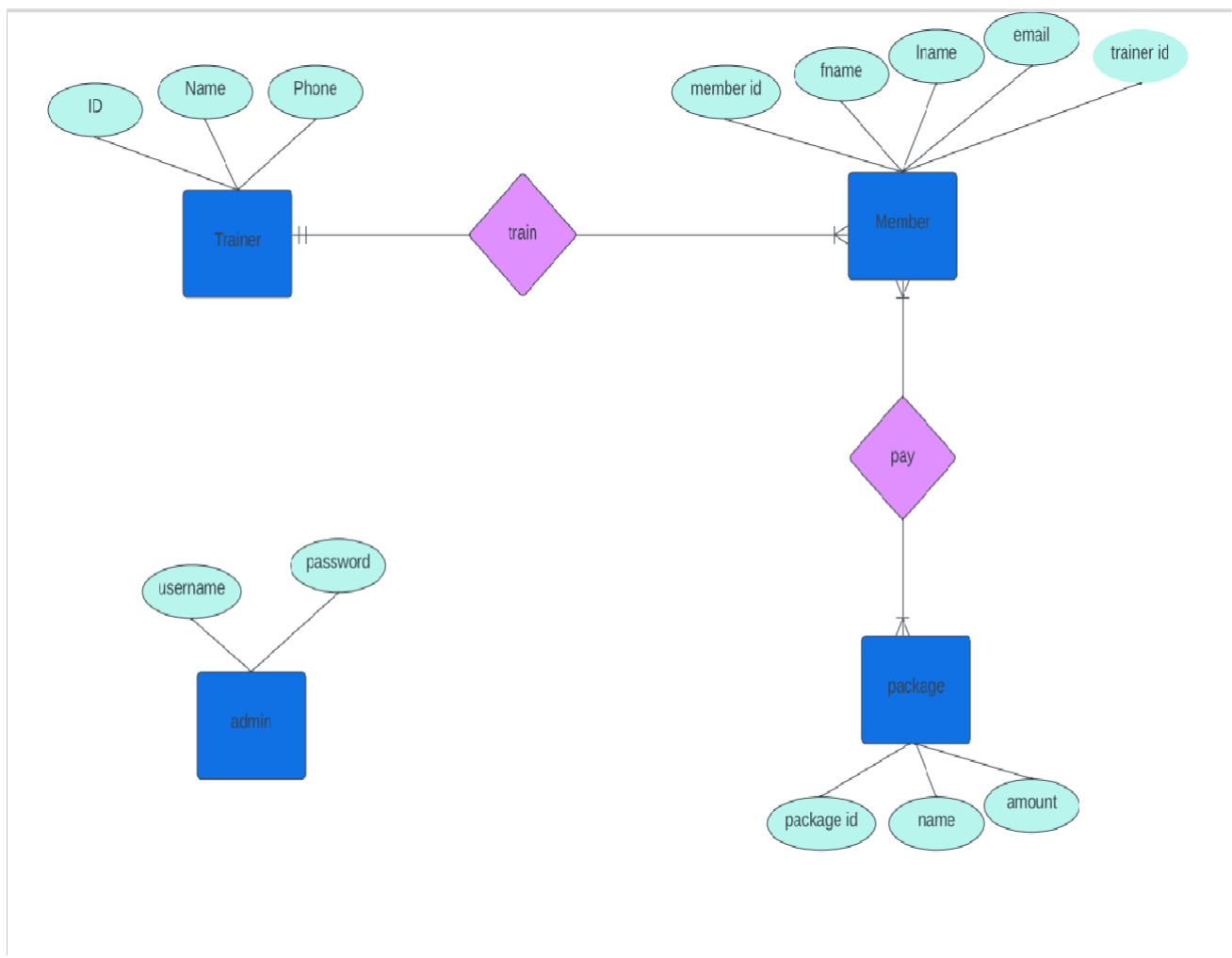
Diagrams

Entity Diagram

The entity diagram represents the various entities or objects within the Gym Management System, such as clients, trainers, packages, and appointments.

It illustrates the relationships and attributes of these entities, helping to define the structure and organization of the system's data.

The entity diagram is essential for database design and ensures efficient storage and retrieval of information.

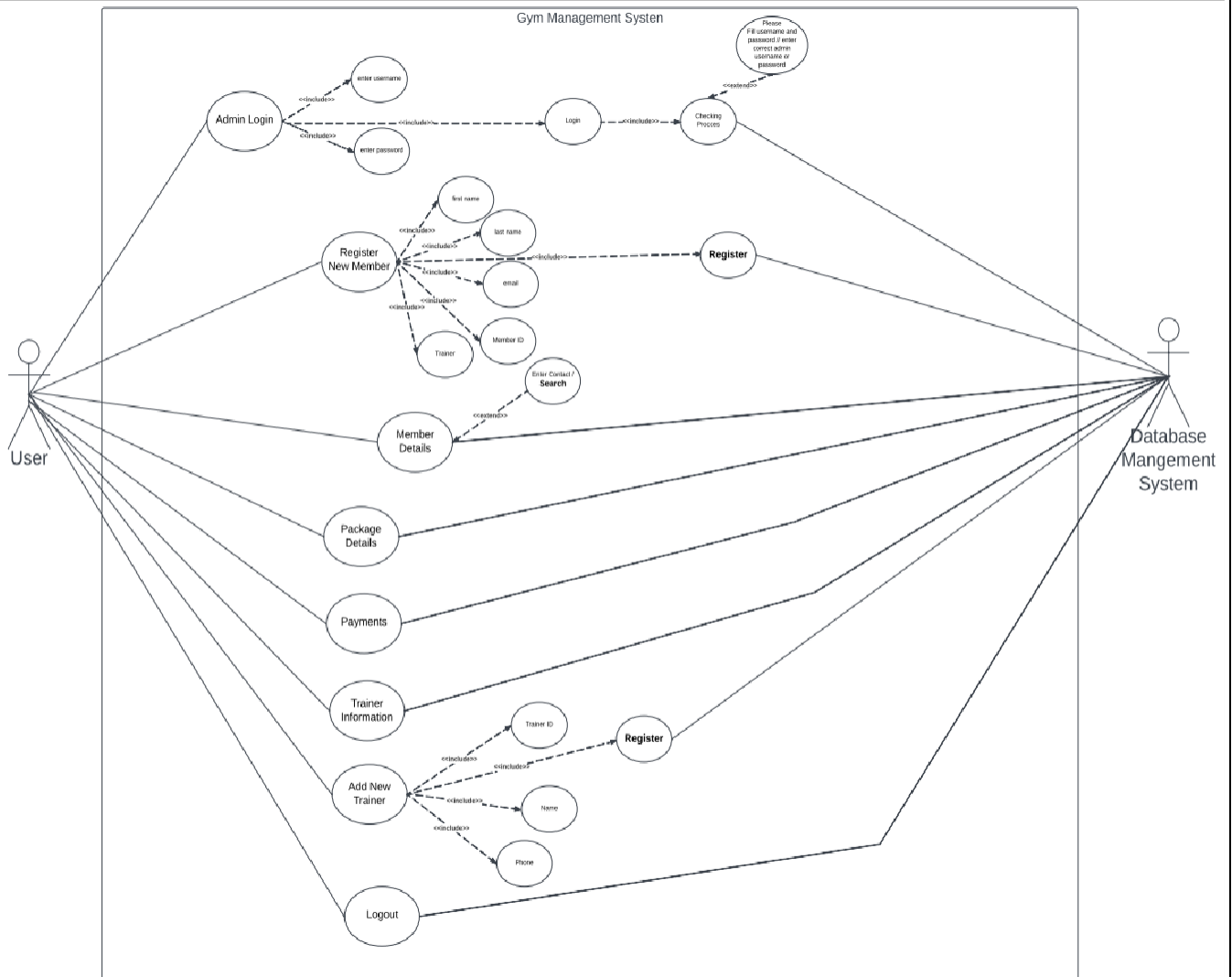


Use Case Diagram

The use case diagram depicts the different actions or functionalities that users can perform within the Gym Management System

It identifies the actors (such as administrators, trainers, and clients) and their interactions with the system.

The use case diagram helps to capture the system's requirements, identify user roles and responsibilities, and provide a high-level view of system functionality.



Class Diagram

The class diagram represents the various classes, their attributes, and their relationships within the Gym Management System.

It illustrates the structure of the system's objects, their properties, and the methods they can perform.

The class diagram serves as a blueprint for implementing the system's business logic and aids in understanding the system's architecture and design.

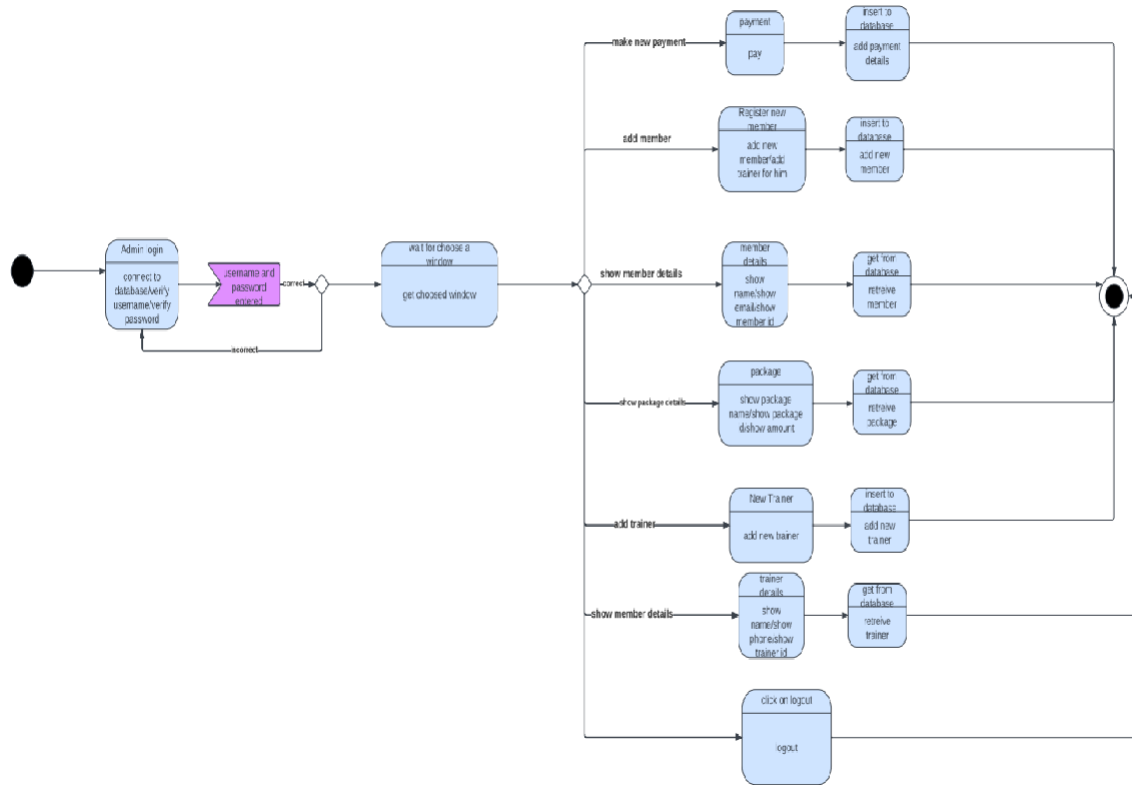


State Machine Diagram

The state machine diagram models the different states and transitions of a particular entity or object within the Gym Management System

It shows how an object's behavior changes in response to events and conditions

The state machine diagram is particularly useful for modeling complex workflows, such as the lifecycle of a client membership or the progress of a trainer certification process

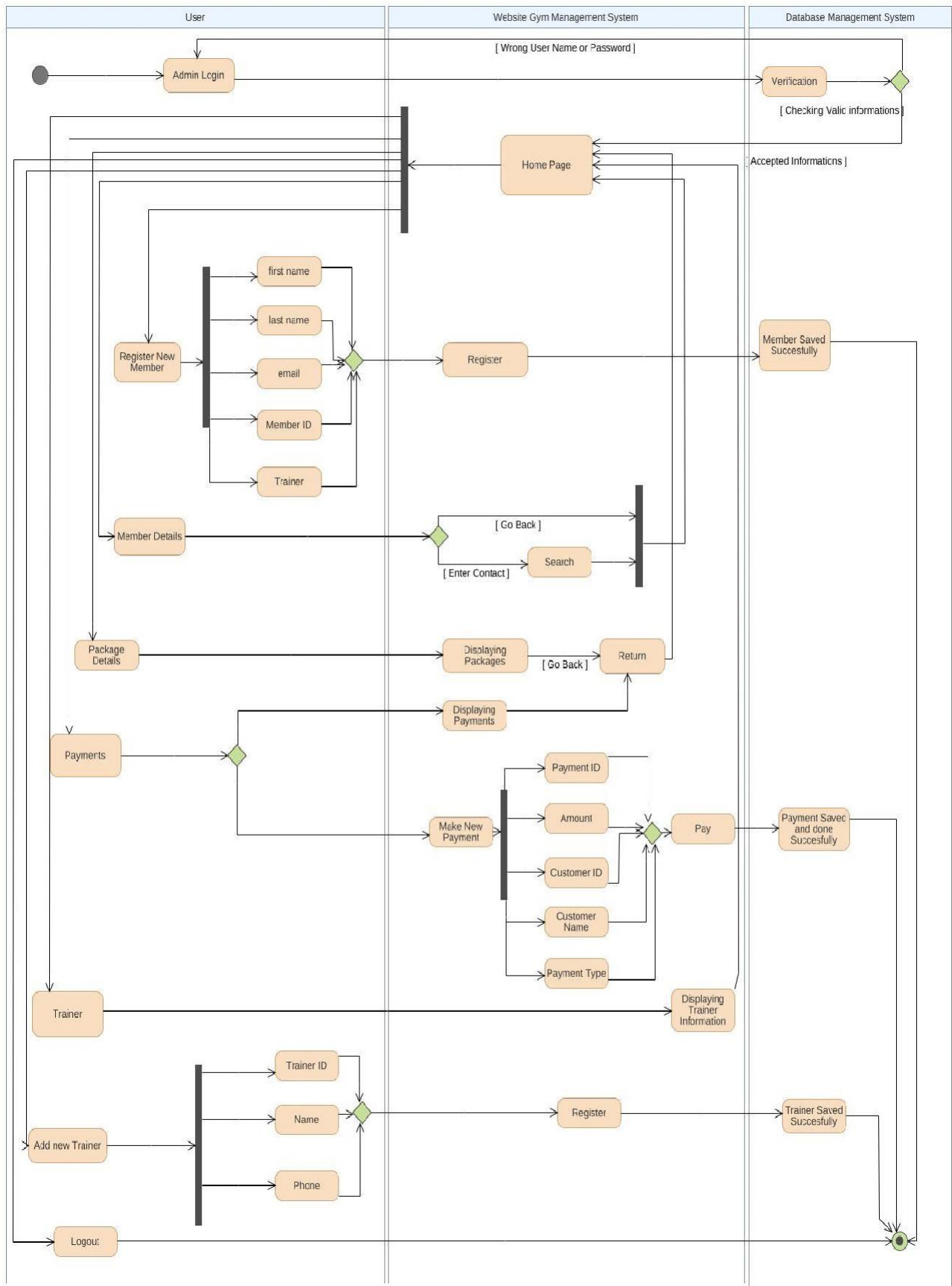


Activity Diagram

The activity diagram represents the flow of activities or steps involved in a particular process or scenario within the Gym Management System.

It shows the sequential and parallel activities, decision points, and the order in which actions occur.

The activity diagram helps to visualize and understand the logic and control flow of processes, such as client registration or appointment scheduling. (Described on Next Page According its much bigger to fit on this half of page)

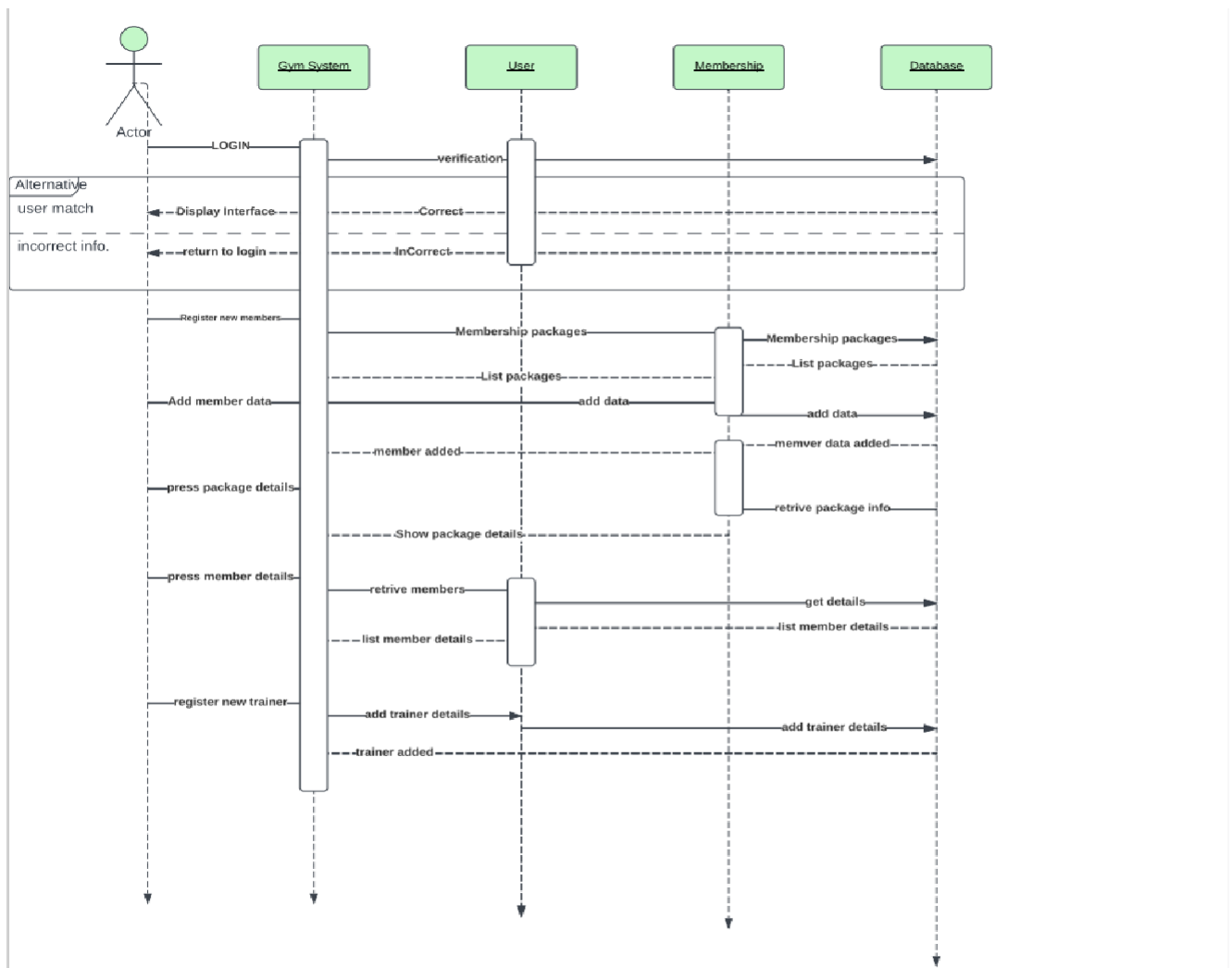


Sequence Diagram

The sequence diagram represents the interactions and messages exchanged between objects or components in a specific scenario or use case within the Gym Management System.

It illustrates the chronological order of method calls, along with the objects involved and the flow of control between them.

The sequence diagram helps to visualize the dynamic behavior of the system, showing how objects collaborate to accomplish a particular task or achieve a specific outcome.



System Evolution

Introduction

The Gym Management System is a **web-based application** designed to streamline the management and administration of a gym or fitness center. This document outlines the system's evolution, taking into consideration the suggested system application as a web-based architecture and the chosen architecture design as a three-tier architecture.

Initial Development

The initial development phase of the Gym Management System focused on establishing the foundation for the system's functionality and structure. The web-based application architecture was chosen to provide accessibility by the admin from admin login webpage and platform independence for clients and trainers.

The three-tier architecture was implemented, consisting of the presentation layer, business logic layer, and data access layer. The presentation layer included a user-friendly web interface accessible through browsers or mobile devices as we provided on our website is a **web based application**. The business logic layer handled client requests, authentication, business rules with all services provided, package calculations, and registering new trainers and members. The data access layer interacted with the database, managing client information, trainer details, and package data and available packages .

Enhancements and Iterations

Following the initial development, the Gym Management System underwent several enhancements and iterations to improve its capabilities and address specific requirements. Some notable enhancements include:

User Interface Refinement: The user interface was refined based on feedback of team members and usability, ensuring an intuitive and visually appealing design. Changes were made to optimize navigation, provide informative dashboards, and streamline common tasks.

Additional Functionality: New features were introduced to expand the system's capabilities. This included the addition of add an new members and trainers, searching on members by their contact, providing payment details and surely add new payments.

Scalability Improvements: The system's scalability was improved by implementing load balancing techniques and optimizing database queries to handle increased traffic during peak gym hours or busy periods this case happens when more than allowed admin are trying a new members or trainers or add payments or try to add a new packages.

Security Enhancements: To ensure the security of client data and system access, various security measures were implemented. (only limited access are allowed to access the website)

Future Development Opportunities (Optional Added as Future Development)

Looking ahead, several potential areas for future development and expansion of the Gym Management System could be explored:

Mobile Applications: Developing dedicated mobile applications for both clients and trainers can enhance accessibility and provide additional convenience features such as workout tracking or class bookings.

Integration with Wearable Devices: Integration with wearable fitness devices can allow for real-time monitoring of client activities, providing personalized recommendations and performance tracking.

Advanced Analytics: Implementing advanced analytics and machine learning techniques can offer deeper insights into client behavior, allowing for targeted marketing campaigns, personalized recommendations, and performance optimization.

Social Engagement: Integrating social media platforms or creating a community within the system can encourage client engagement, fostering a sense of belonging and promoting a healthy lifestyle.

Conclusion

The Gym Management System, built on a web-based architecture using a three-tier design, has evolved to become a comprehensive and efficient solution for gym management. Through iterative enhancements, scalability improvements, security enhancements, and ongoing maintenance, the system has adapted to meet the evolving needs of the gym and its users.

The system's architecture and application design choices have proven to be effective in providing accessibility, scalability, maintainability, and security. With a strong foundation and potential for future development, the Gym Management System is well-equipped to support the gym's operations and contribute to its success in

Design Description (How To Use our Web-based-Application)

The design of the Gym Management System revolves around a web-based application, ensuring ease of use and accessibility for users. The system follows a seamless workflow with the following steps:

Login Process:

Users access the system through a secure web-based login screen.

They enter their credentials, such as username and password, to authenticate their identity.

Upon successful authentication, the system transfers the user to the main menu screen.

Main Menu Screen:

The main menu screen serves as the central navigation hub within the web-based application.

Users are presented with several buttons, each accompanied by a clear description of its functionality.

By selecting the appropriate button, users can easily navigate to different sections or perform specific actions within the Gym Management System.

Through the web-based application design, the Gym Management System ensures that users can access the system conveniently from various devices with internet connectivity. The login process verifies user credentials, granting secure access to authorized personnel. The main menu screen provides an intuitive interface, offering clear descriptions of the available functions, enabling users to efficiently navigate through the system and perform their desired tasks.

Implementation & Used software:

- Database Management System → MySQL
- Website Implementation →
 - HTML
 - CSS (Design)
 - JavaScript
 - PHP

IDE's used → Visual Studio, PyCharm

Tools For running Code → XAMPP (MySQL – Apache)

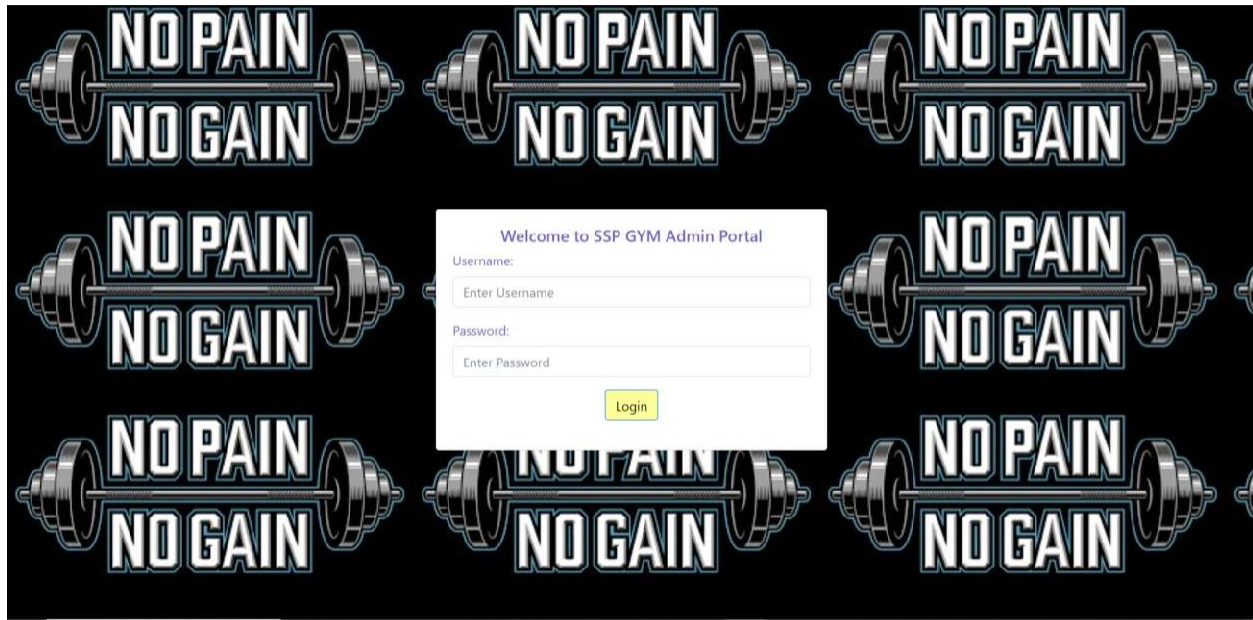
Uml Diagrams, Entity Relationship Graph → LucidChart – GenMyModel

Test case

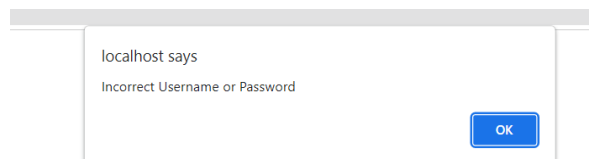
1-Login page

The admin must enter a correct username and correct password as it given in the database

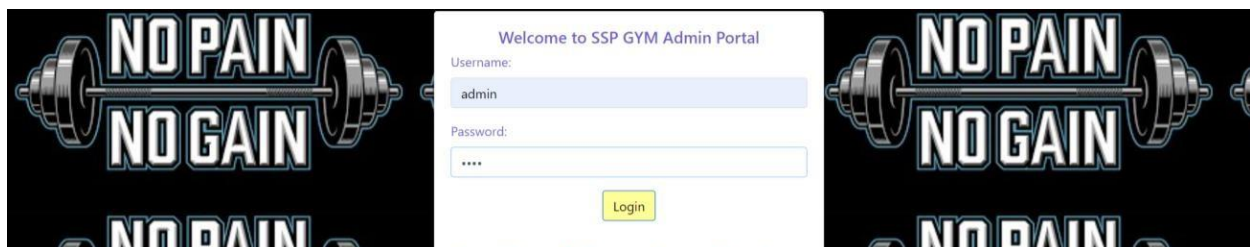
If not, an alert is shown.



If the admin entered incorrect username or password , an alert is shown as the next figure



Our admin username is “admin” and password is “pass” as it given in the database



2-New member registration

Admin can add new member by filling a form include the new member's name,id,email and Choosing a specific trainer for him.

Members options

- Check Member details
- View Package details
- Payments

Trainer details/Add new Trainer

New members registration

First Name: ayman

Last Name: mohamed

ID: 144

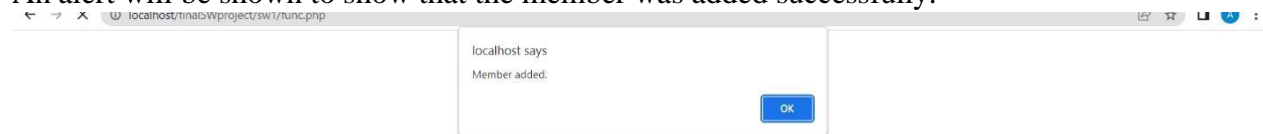
Email: ayman@gmail.com

Choose a trainer: saved

Register

Logout

An alert will be shown to show that the member was added successfully.




The member will be added to the database too.

fname	lname	email	contact	trainer
ayman	mohamed	ayman@gmail.com	144	101
sayed	abdelfatah	sayed@gmail.com	201	101
mostafa	kassas	mustafa121@gmail.com	202	102
surya	tarek	surya@gmail.com	203	101
Rawan	sayed	rawan@gmail.com	204	103
ziad	wahed	ziad@gmail.com	205	103
eyad	hamdy	momo@gmail	3456	101
kasso	kasso	kasso@gmail	6666	101


3-Member details

This page shows all the details of each member .



Go Back		Members Details			<input type="text" value="Enter ID or Name"/>		<input type="button" value="Search"/>
First Name	Last Name	Email	ID	Trainer ID			
ayman	mohamed	ayman@gmail.com	144	101			
sayed	abdelfatah	sayed@gmail.com	201	101			
mostafa	kassas	mustafa121@gmail.com	202	102			
surya	tarek	surya@gmail.com	203	101			
Rawan	sayed	rawan@gmail.com	204	103			
ziad	wahed	ziad@gmail.com	205	103			
eyad	hamdy	momo@gmail	3456	101			
kasso	kasso	kasoo@gmail	6666	101			

Admin is also able to search for any member by it's id or by it's name.



Go Back		Members Details			<input type="text" value="201"/>		<input type="button" value="Search"/>
First Name	Last Name	Email	ID	Trainer ID			
ayman	mohamed	ayman@gmail.com	144	101			
sayed	abdelfatah	sayed@gmail.com	201	101			
mostafa	kassas	mustafa121@gmail.com	202	102			
surya	tarek	surya@gmail.com	203	101			
Rawan	sayed	rawan@gmail.com	204	103			
ziad	wahed	ziad@gmail.com	205	103			
eyad	hamdy	momo@gmail	3456	101			
kasso	kasso	kasoo@gmail	6666	101			

Go Back				
First name	Last name	Email id	Contact	Trainer ID
sayed	abdelfatah	sayed@gmail.com	201	101

4-Package details

This page shows all the details of each package.



Go Back Package Details		
Package ID	Package Name	Amounts
121	platinum	800
122	vip	1500
123	gold	1000

5-Payments details

This page shows the details of the payment of each member.



Go Back Payment Details				
Payment ID	Amount	Payment Type	Customer ID	Customer Name
301	1500	cash	201	sayed
302	800	card	202	mostafa
303	1000	cheque	203	surya
304	1500	cash	204	rawan
305	1000	cash	11	osama
306	1000	cash	11	osama

6-Make a new payment

Admin is also able to make a new payment for any new customer by filling a form includes payment id, amount, customer id, customer name and payment type.

Make new Payment

Payment ID

Amount

Customer ID

Customer Name

Payment Type

An alert will be shown to show that the payment was added successfully.



The payment will be added to the database too.

A screenshot of the phpMyAdmin interface. The left sidebar shows a tree view of databases, with 'loginsystem' selected. The main area displays the 'payment' table. At the top, there are tabs for 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', 'Tracking', and 'Triggers'. Below these is a SQL query editor with the text 'SELECT * FROM `payment`'. Underneath the editor are options for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (a search box), and 'Sort by key' (set to None). Below these options is a table with 7 columns: 'Payment_id', 'Amount', 'customer_id', 'customer_name', and 'payment_type'. The table contains 7 rows of data. Each row has a checkbox and three icons (Edit, Copy, Delete) to its left. At the bottom of the table, there are links for 'Check all', 'With selected:', 'Edit', 'Copy', 'Delete', and 'Export'.

	Payment_id	Amount	customer_id	customer_name	payment_type
<input type="checkbox"/>	301	1500	201	sayed	cash
<input type="checkbox"/>	302	800	202	mostafa	card
<input type="checkbox"/>	303	1000	203	surya	cheque
<input type="checkbox"/>	304	1500	204	rawan	cash
<input type="checkbox"/>	305	1000	11	osama	cash
<input type="checkbox"/>	306	1000	11	osama	cash
<input type="checkbox"/>	307	1000	144	ayman	cash

7-Trainer details

This page shows all details of each trainer .



Go Back Trainer Information		
Trainer ID	Name	Phone
2	kasso	123456789
3	ahmed samir	5556211
101	sayed	12365489
102	aymen	21365789
103	ali	123564789
104	Sameer	12536987

8-Register new trainer

Admin is also able to register new trainer by filling form includes trainer's id, name and phone number.

Register new Trainer

Trainer ID

105

Name

omar

Phone

55626111

Register

An alert will be shown to show that the trainer was added successfully.

localhost says
Trainer added.

OK

The Trainer will be added to the database too.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'loginsystem'. The 'trainer' table is selected, and its structure and data are displayed. The table has three columns: 'Trainer_Id', 'Name', and 'phone'. There are six rows of data. The interface includes a sidebar with a tree view of databases and tables, a top menu bar with options like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', 'Tracking', and 'Triggers', and a bottom status bar with 'Show all', 'Number of rows', 'Filter rows', and 'Sort by key'.

Trainer_Id	Name	phone
2	kasso	123456789
3	ahmed samir	5556211
101	sayed	12365489
102	ayman	21365789
103	ali	123564789
104	Sameer	12536987
105	omar	55626111