

Darshan University

A Project Report on

**“Gym Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Harmik Rathod | Enrollment No.: 24010101680 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2301CS405)** for entitled **“Gym Management System”** submitted in partial fulfilment for the Semester-4 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Harmik Rathod

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Gym Management System” has** been satisfactorily prepared by **Harmik Rathod** (**24010101680**) under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliyaand all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Harmik Rathod**

**ABSTRACT**

A Gym Management System is designed to simplify and digitalize the daily operations of a gym. This system aims to provide a computerized platform for managing memberships, tracking attendance, scheduling classes, and monitoring gym resources. It acts as a modern tool to replace traditional manual processes, ensuring efficient and streamlined operations.

The system includes features like member and manager login, enabling personalized dashboards for users. Members can register, view their profiles, check attendance, and access their workout schedules. It also allows them to make payments online and view payment history. Additionally, members can request personalized training sessions or diet plans by submitting a request form.

For manager, the system offers advanced functionalities such as adding, updating, and deleting member details, managing trainers and tracking equipment usage

The primary goal of the Gym Management System is to reduce manual workload, enhance user experience, and optimize gym efficiency while maintaining accurate records.

**Table of Contents**

[List of Figures I](#_Toc185614192)

[List of Tables II](#_Toc185614193)

[1 Introduction 1](#_Toc185614194)

[1.1 Product perspective 1](#_Toc185614195)

[1.2 Product features 1](#_Toc185614196)

[1.2.1 There are three different users who will be using this product: 1](#_Toc185614197)

[1.2.2 The features that are required for the Manager are: 1](#_Toc185614198)

[1.2.3 The features that are required for the Trainer are: 1](#_Toc185614199)

[1.2.4 The features that are required for the Member are: 1](#_Toc185614200)

[1.3 Functional Requirement 1](#_Toc185614201)

[1.3.1 Manager 1](#_Toc185614202)

[1.3.2 Trainer 2](#_Toc185614203)

[1.3.3 Member 2](#_Toc185614204)

[1.3.4 Cleaning Staff 2](#_Toc185614205)

[1.4 Non-Functional Requirement 2](#_Toc185614206)

[1.4.1 Performance: 2](#_Toc185614207)

[1.4.2 Scalability: 2](#_Toc185614208)

[1.4.3 Security: 2](#_Toc185614209)

[1.4.4 Availability: 2](#_Toc185614210)

[1.4.5 Accessibility: 2](#_Toc185614211)

[1.4.6 Maintainability : 2](#_Toc185614212)

[1.4.7 Data Backup: 2](#_Toc185614213)

[2 Design and Implementation Constraints 3](#_Toc185614214)

[2.1 Use case diagram 3](#_Toc185614215)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc185614216)

[2.3 Sequence diagram 5](#_Toc185614217)

[2.4 State diagram 6](#_Toc185614218)

[2.5 Class diagram 7](#_Toc185614219)

[2.6 Data flow diagram 8](#_Toc185614220)

[2.6.1 Context diagram (level-0) 8](#_Toc185614221)

[2.6.2 DFD Level-1 8](#_Toc185614222)

[2.6.3 DFD Level-2 9](#_Toc185614223)

[3 External interface requirement (Screens) 10](#_Toc185614224)

[3.1 Screen-1: Registration Form 10](#_Toc185614225)

[3.2 Screen-2: Login Form 11](#_Toc185614226)

[3.3 Screen-3: Add borrower book 12](#_Toc185614227)

[4 Database design 13](#_Toc185614228)

[4.1 List of Tables 13](#_Toc185614229)

[5 Stories and Scenario 14](#_Toc185614230)

[5.1 Story-1: Add New Book in Library Catalogue 14](#_Toc185614231)

[5.1.1 Scenario# S1.1 14](#_Toc185614232)

[5.1.2 Scenario# S1.2 14](#_Toc185614233)

[5.1.3 Scenario# S1.3 14](#_Toc185614234)

[5.2 Story-2: Search Book 15](#_Toc185614235)

[5.3 Story-3: Manage due date for borrowed book 15](#_Toc185614236)

[5.4 Story-4: Renew book 15](#_Toc185614237)

[5.5 Story-5: Generate a report on book usage and availability 15](#_Toc185614238)

[6 Test cases 16](#_Toc185614239)

[7 References 20](#_Toc185614240)

# List of Figures

[Figure 2.1‑1 Use case diagram for gym management system 3](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for fill attendance 4](#_Toc146706849)

[Figure 2.2‑2 Activity diagram for trainer selection 4](#_Toc146706849)

[Figure 2.2‑3 Swimlane diagram for fill attendance](#_Toc146706850) 5

[Figure 2.3‑1 Sequence diagram for fill attendance 5](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 6](#_Toc146706852)

[Figure 2.4‑2 State diagram for Librarian 6](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Library management system 7](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Library management system 8](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Library management system 8](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for Issue book 9](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Registration Form 10](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Login Form 11](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Add borrower book 12](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 10](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 11](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 12](#_Toc146706863)

[Table 4.1‑1 Table: Book 13](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 13](#_Toc146706865)

[Table 4.1‑3 Table: Staff 13](#_Toc146706866)

[Table 4.1‑4 Table: Student 13](#_Toc146706867)

# Introduction

## Product perspective

This project is focused on transforming the manual gym management system into an internet-based application to enhance user experience and streamline operations. The system allows users to manage their accounts, check the availability of equipment, and view schedules for workout sessions and classes. It is a multi-user platform capable of handling all the essential functions of a gym, such as Membership Management, Equipment Tracking, Scheduling, and Billing. This system is designed to meet the fundamental requirements of a small to medium-sized gym, ensuring efficient operations and improved member satisfaction

## Product features

### There are three different users who will be using this product:

* Manager manage gym details and history of member and trainer.
* Member can be view his details view diet plan.
* Trainer can assign exercise to member.

### The features that are required for the Manager are:

* Manage the gym's schedule, equipment usage, and avoid conflicts in booking.
* Search for specific members in the system based on their name, membership ID, etc.
* Manage and track the availability of equipment and maintenance schedules.
* Register new members and manage their profiles.
* Edit the information of existing members.
* View reports of active memberships, member attendance.
* Manage subscription plans and their renewal.
* Can check the report of the issued Books.
* Can access all the accounts of the students.

### The features that are required for the Trainer are:

* View assigned members and their fitness goals.
* Create and update personalized workout plans for members.
* Mark attendance for assigned members during sessions.
* Track members’ progress and update fitness metrics.

### The features that are required for the Member are:

* View available gym equipment and facilities.
* Access fitness reports shared by trainers.
* View payment history and renew memberships.
* Request a specific fitness program or session with a trainer.

## Functional Requirement

### Manager

* Add Trainer: Add Training Staff in database
* Add Member: Add Member in database
* Add Cleaning Staff: Add Cleaning Staff in database
* Remove Trainer: Remove Training Staff to database
* Remove Member: Remove Member to database
* Remove Cleaning Staff: Remove Cleaning Staff to database
* Assign Trainer: Assign Trainer to Member
* Pay Salary: Pay Salary to trainer and Cleaning Staff
* Collect Fees: Manager Collect fees from member

### Trainer

* View Member: Trainer view his member
* View Salary: View his Salary History
* Request for Leave: Trainer need to leave that request to online
* Assign Task: Assign task to member
* Add content: Add new exercise in the database
* Attendance: Mark the member attendance
* Request to Equipment: To request for new equipment

### Member

* Registration: Register to new membership
* Pay Fees: to Pay fees
* View Diet Plan: TO check his diet plan
* View Exercise: TO check exercise to give trainer
* Gym Details: can view Gym details

### Cleaning Staff

* View Salary: View his Salary History
* Request for Leave: Trainer need to leave that request to online
* Request to Overtime: To request to overtime
* Task Completion Update: Staff should mark task as completion

## Non-Functional Requirement

### Performance:

* The system will deliver high performance, supporting multiple user access simultaneously without significant delays or downtime.

### Scalability:

* The project will be designed to accommodate scalability, allowing for future enhancements and feature additions post-deployment without extensive redesign.

### Security:

* The system will ensure secure transactions for payments and protect user data through encryption and robust authentication mechanisms.

### Availability:

* The software will be accessible anytime and anywhere, ensuring uninterrupted access for users when needed.

### Accessibility:

* The system will feature a user-friendly interface, making it easy to use and navigate for both members and administrators.

### Maintainability :

* The software will be designed for easy maintenance, ensuring that updates, bug fixes, and modifications can be performed efficiently after deployment.

### Data Backup:

* The database will be securely stored in the cloud or on a reliable server to ensure regular backups and quick recovery in case of data loss.

# Design and Implementation Constraints

## Use case diagram

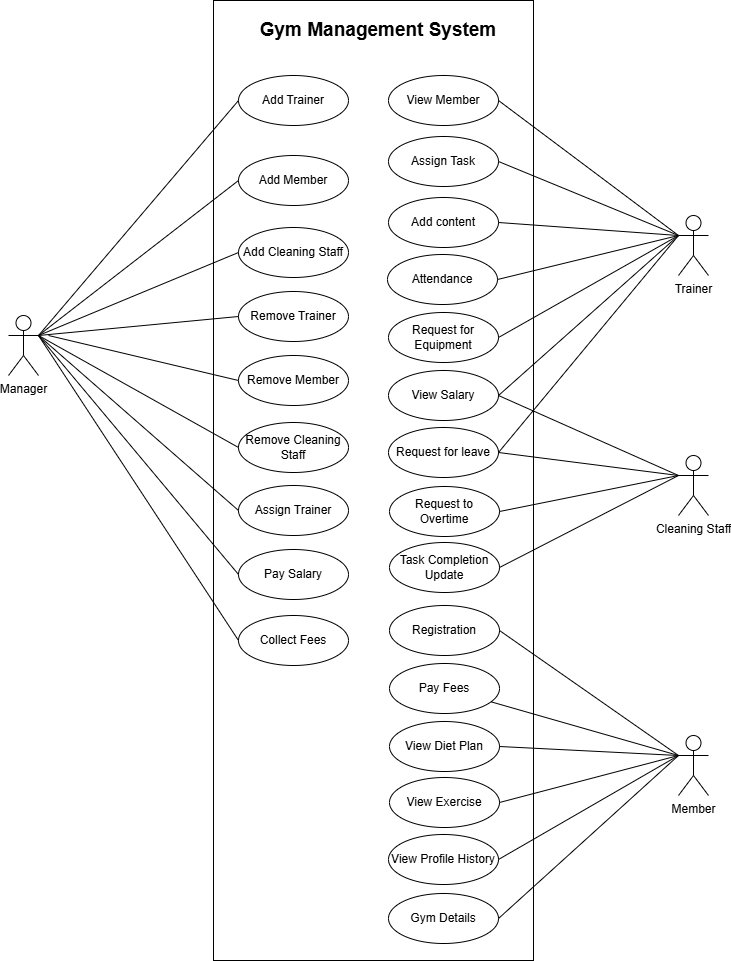


Figure 2.1‑1 Use case diagram for gym management system

## Activity diagram and Swimlane diagram

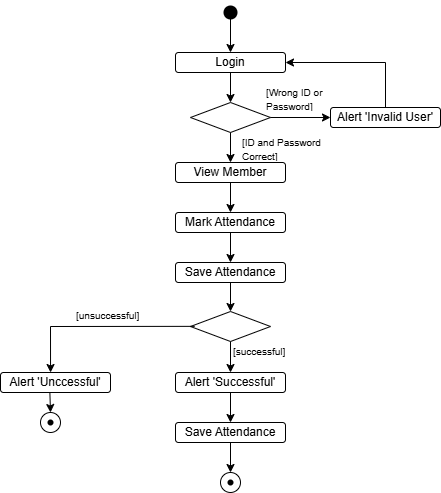


Figure 2.2‑1 Activity diagram for fill attendance

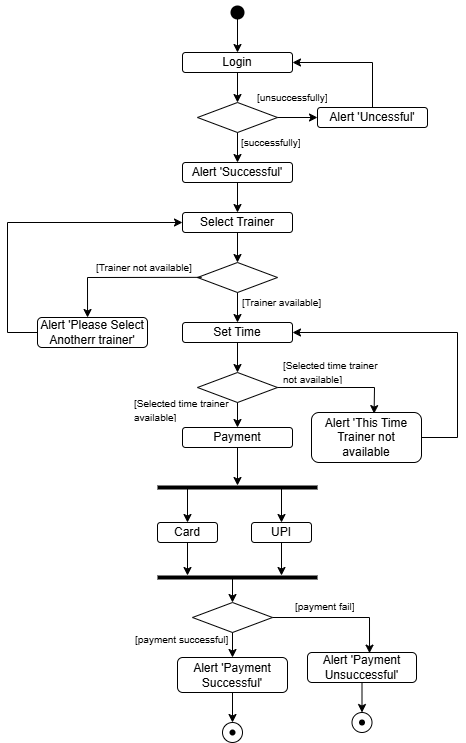


Figure 2.2‑2 Activity diagram for trainer selection

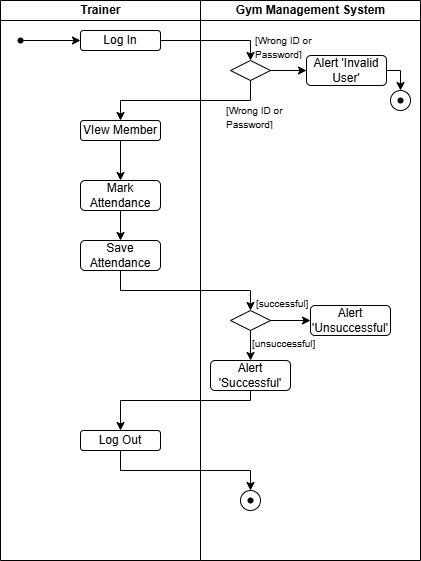


Figure 2.2‑3 Swimlane diagram for fill attendance

## Sequence diagram

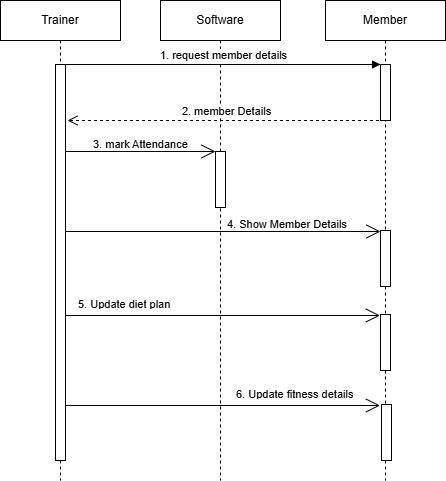


Figure 2.3‑1 Sequence diagram for Trainer

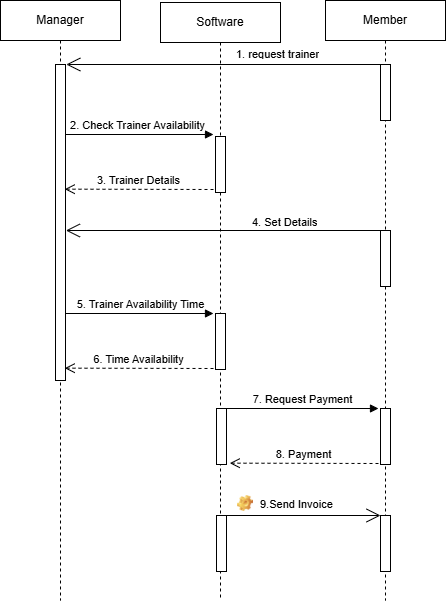


Figure 2.3‑2 Sequence diagram for Manager

## State diagram

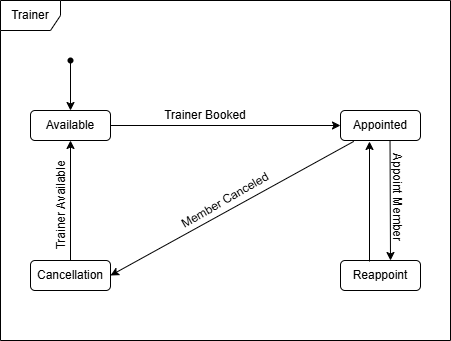


Figure 2.4‑1 State diagram of Trainer

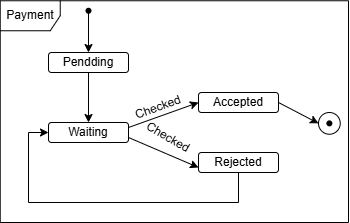


Figure 2.4‑2 State diagram for Payment

## Class diagram

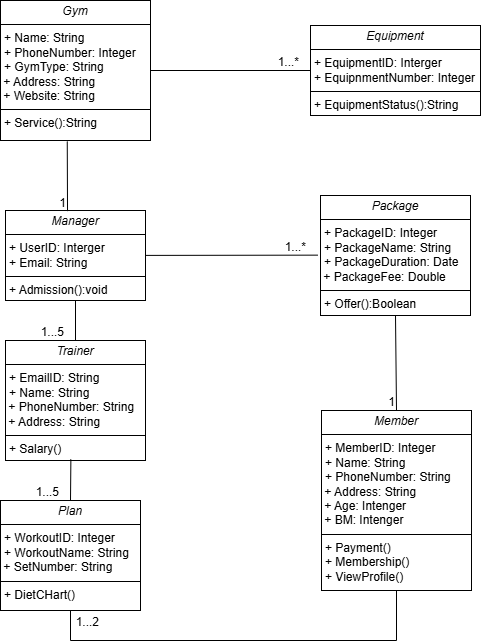


Figure 2.5‑1 Class diagram for Gym management system

## Data flow diagram

### Context diagram (level-0)

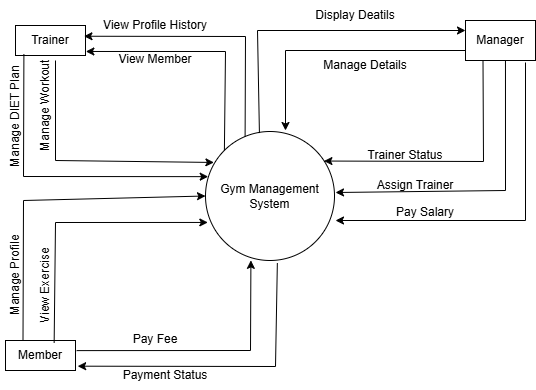


Figure 2.6‑1 Context diagram for gym management system

### DFD Level-1

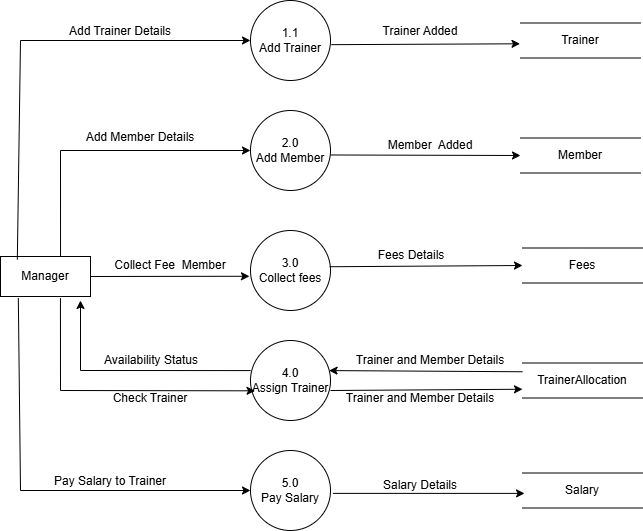


Figure 2.6‑2 DFD level-1 for gym management system

### DFD Level-2

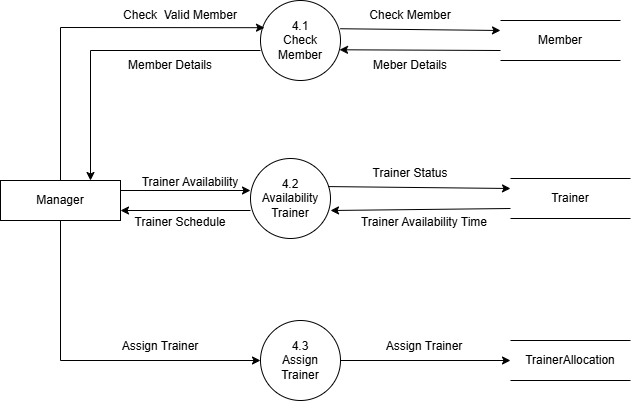


Figure 2.6‑3 DFD level-2 for Assign Trainer

# External interface requirement (Screens)

## Screen-1: Add Cleaning Staff

Figure 3.1‑1 Screen-1: Add Cleaning Staff

**Purpose:** This screen specifically for managing Cleaning Staff. It allows manager to add new cleaning staff members by entering their details, such as name, phone number, and shift schedule.

Table 3.1‑1 Screen element of Registration form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Name | Textbox | M | 1 | Name field should be editable and accept the full name of the staff. |
| 2 | Phone | Textbox | M | 1 | Phone number field should be editable and accept a valid phone number. |
| 3 | Shift | Dropdown | M | 1 | Dropdown field to select the shift for the staff member. |
| 4 | Add Staff | Password | ------ | ------- | Button to submit the form and save the cleaning staff details. |

## Screen-2: Assign Task To Member

Figure 3.2‑1 Screen-2: Assign Task to Member

**Purpose:** This form is used to assign tasks to a selected member. The user can enter task details, set a deadline, and define priority before submitting the assignment. It helps in organizing and tracking task distribution efficiently.

Table 3.2‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 3 | Remember Me | Checkbox | M | 1 | Saving login credentials through remember me checkbox |
| 4 | I forgot my password | Link | ------ | ------ | Link for navigate to Forgot password page for allows users to recover password. |
| 5 | Register a new membership | Link | ------ | ------ | Link for navigate to membership registration. |
| 6 | Sign in | Button | ------ | ------ | Login button navigates to another page even if valid login credentials. |

## Screen-3: View Exercise

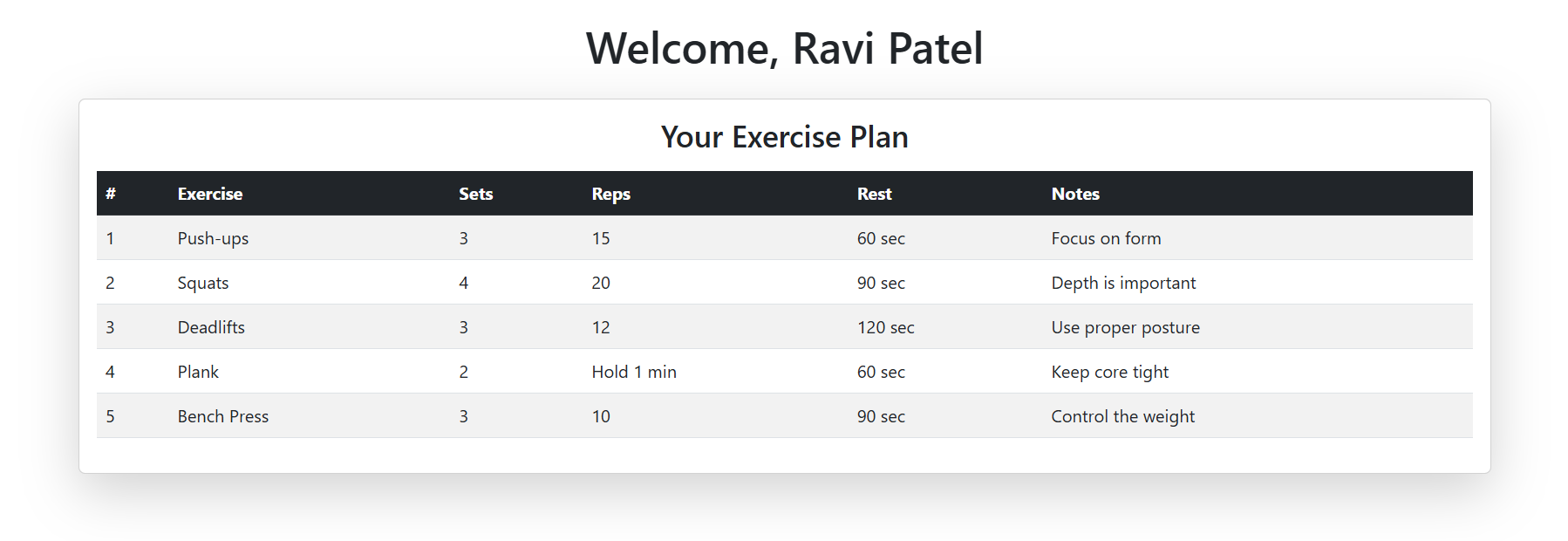


Figure 3.3‑1 Screen-3: View Exercise

**Purpose:** This screen displays a personalized exercise plan for the user. It provides details such as exercise name, number of sets and repetitions, rest duration, and additional notes to ensure proper form and effectiveness..

Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Exercise Name | Text | M | 1 | Lists the name of each exercise in the plan. |
| 2 | Sets | Textbox | M | 1 | Displays the number of sets for each exercise. |
| 3 | Reps | Text/Numeric | M | 1 | Shows the repetitions or duration for each exercise. |
| 4 | Rest Time | Numeric | M | 1 | Displays the rest period between sets. |
| 5 | Notes | Textbox Text | O | 1 | Provides additional instructions or tips for each exercise. |

## Screen-4: Overtime Request

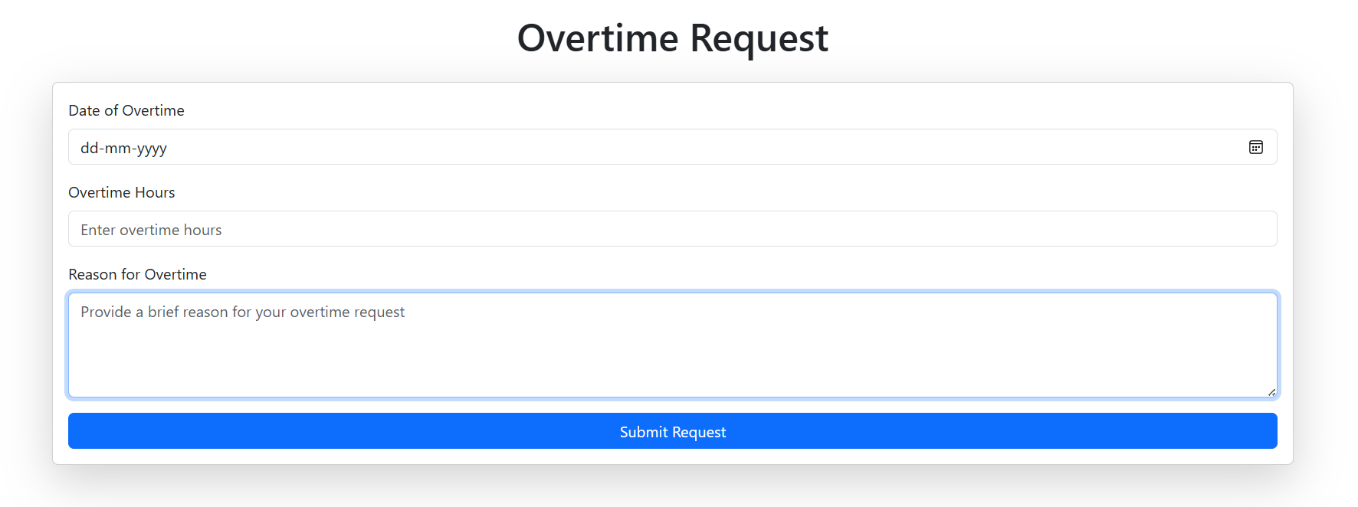


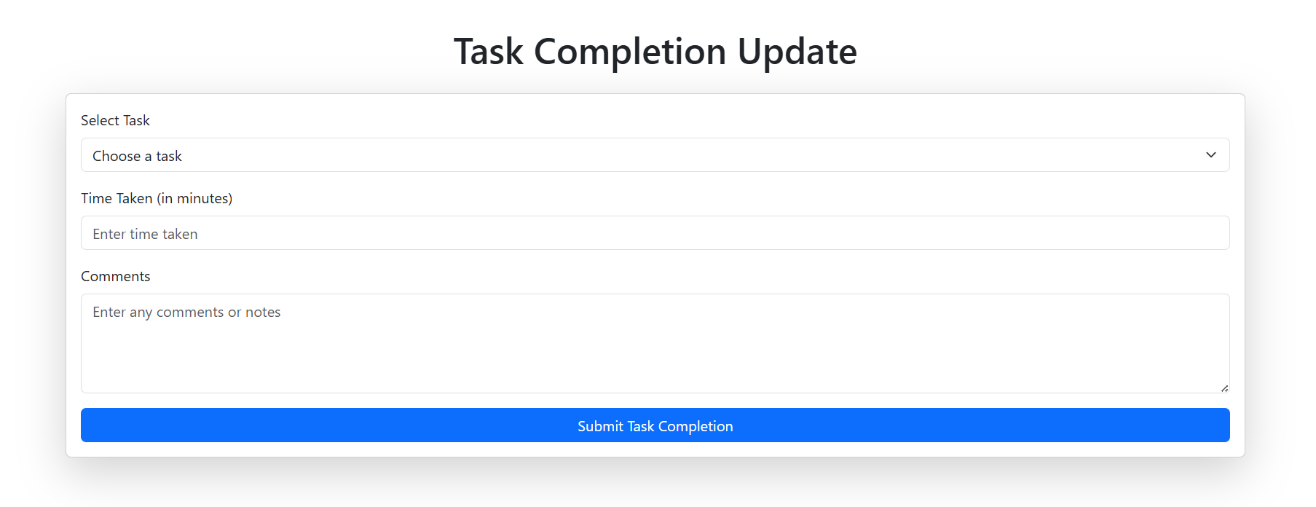
Figure 3.3‑1 Screen-3: View Exercise

**Purpose:** This screen is used for submitting an Overtime Request. Employees can input details such as the date of overtime, number of hours worked, and the reason for the request before submitting it for approval.

Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Date of Overtime Input | Date Picker | M | 1 | Allows users to select the overtime date. |
| 2 | Overtime Hours | Text Box | M | 1 | Allows users to enter the number of overtime hours. |
| 3 | Reason for Overtime | Text Area | M | 1 | Allows users to provide a brief reason for the overtime request. |
| 4 | Submit Request Button | Button | M | 1 | Submits the overtime request. |

## Screen-3: Overtime Request

Figure 3.3‑1 Screen-3: View Exercise

**Purpose:** This screen is used for submitting a Task Completion Update. Employees can provide details about the completed task, including task selection, time taken, and additional comments before submitting the update.

Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Select Task | Dropdown | M | 1 | |  | | --- | | Allows users to choose a task from a predefined list. |  |  | | --- | |  | |
| 2 | Time Taken | |  | | --- | | Text Box |  |  | | --- | |  | | M | 1 | |  | | --- | | Allows users to enter the time taken (in minutes) to complete the task. |  |  | | --- | |  | |
| 3 | Comments | |  | | --- | | Text Area |  |  | | --- | |  | | O | 1 | |  | | --- | | Allows users to enter additional notes or comments. |  |  | | --- | |  | |
| 4 | |  | | --- | | Submit |  |  | | --- | |  | | Button | M | 1 | Submits the task completion update. |

# Database design

## List of Tables

* Manager
* Trainer
* Member
* Equipment
* Package

Table 4.1‑1 Table: Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| ManagerID | Int | NN | PK (Auto Increment) |  |
| ManagerName | varchar(100) | NN |  |  |
| Email | varchar(100) | AN |  |  |
| MobileNo | String | AN |  |  |

Table 4.1‑2 Table: Trainer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| TainerID | Int | NN | PK (Auto Increment) |  |
| TrinerName | varchar(100) | NN |  |  |
| JoiningDate | DateTime | NN |  |  |
| Availabilty | Boolean | NN |  |  |
| PhoneNo | String | AN |  |  |
| Email | Varchar(100) | NN |  |  |

Table 4.1‑3 Table: Member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| MemberID | Int | NN | PK (Auto Increment) |  |
| MemberName | varchar(100) | NN |  |  |
| JoiningDate | DateTime | NN |  |  |
| Gender | Varchar(50) | AN |  |  |
| BMI | Int | NN |  |  |
| Age | int | AN |  |  |
| PhoneNo | String | AN |  |  |
| Address | Varchar(100) | AN |  |  |
| TrainerID | int | NN | FK | Reference on Trainer Table |
| PackageID | Int | NN | FK | Reference on Package Table |

Table 4.1‑4 Table: Equipment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| EquipmentID | int | NN | PK (Auto Increment) |  |
| EquipmentNo | Int | NN |  |  |

Table 4.1-5 Table: Package

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| PackageID | Int | NN | PK (Auto Increment) |  |
| PackageName | varchar(100) | NN |  |  |
| PackageDuration | DateTime | NN |  |  |
| PackageFee | Double | NN |  |  |

# Stories and Scenario

## Story-1: Add New Member

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Manager,  I want to add a new member  So that member can access his portal |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | New member request come to gym |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Member with Valid Information |
| Prerequisite | **:** | Manager is logged in to the Gym management system. |
| Acceptance Criteria | **:** | **Given:**  The Manager is navigated to the Dashboard.  **When:**  The manager selects the "Add Member" option  And The manager enters valid member details  The manager clicks the "Save" button to add the member  **Then** the system successfully adds the member to the database and the manager receives a confirmation message with the member id. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New Member with invalid Information |
| Prerequisite | **:** | Manager is logged in to the Gym management system. |
| Acceptance Criteria | **:** | **Given:**  The Manager is navigated to the Dashboard.  **When:**  The manager selects the "Add Member" option  And The manager enters invalid member details  The manager clicks the "Save" button to add the member  **Then** the system error adds the member to the database and the manager receives a error dialog box. |

## Story-2: Add Trainer

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Manager,  I want to add new trainer,  So that trainer can access his portal. |
| Priority | **:** | High |
| Estimate | **:** | XXL |
| Reason | **:** | New Trainer appointed |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Trainer with Valid Information |
| Prerequisite | **:** | Manager is logged in to the Gym management system. |
| Acceptance Criteria | **:** | **Given:**  The Manager is navigated to the Dashboard.  **When:**  The manager selects the "Add Trainer" option  And The manager enters valid trainer details  The manager clicks the "Save" button to add the member  **Then** the system successfully adds the trainer to the database and the manager receives a confirmation message with the member id. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New Trainer with invalid Information |
| Prerequisite | **:** | Manager is logged in to the Gym management system. |
| Acceptance Criteria | **:** | **Given:**  The Manager is navigated to the Dashboard.  **When:**  The manager selects the "Add trainer" option  And The manager enters invalid trainer details  The manager clicks the "Save" button to add the member  **Then** the system error adds the trainer to the database and the manager receives a error dialog box. |

## Story-3: Assign Task

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Trainer,  I want to assign task to member  So that the member can access their portal and track their progress |
| Priority | **:** | Medium |
| Estimate | **:** | L |
| Reason | **:** | Trainers can assign tasks to members so they can track workouts, stay motivated, and make progress easily. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Assigning a Workout Task to a Member |
| Prerequisite | **:** | The Trainer is logged in to the Gym Management System. |
| Acceptance Criteria | **:** | **Given:**  A trainer wants to assign a workout task to a member  **When:**  The trainer selects a member from the system.  The trainer creates a task  The trainer sets a deadline and assigns it.  **Then** The member gets a notification about the new task And The task appears in the member’s portal. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Member Completes Assigned Task |
| Prerequisite | **:** | The Trainer is logged in to the Gym Management System. |
| Acceptance Criteria | **:** | **Given**: A member has an assigned task in their portal.  **When:**  The member completes the workout and marks it as done.  **Then** The trainer gets notified of completion and The task status updates in the system. |

# 

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | EMI Calculator | Test Designed by: | P. U. Jadeja |
| Module Name: | **Login** | **Test Designed date:** | 01-10-2023 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-01-2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: Web application should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Login to web application with valid credential | Functional | Login to Library management system web application through valid credential | TC\_001 |
| TC\_002 | Login to web application with invalid credential | Functional | Login to Library management system web application through invalid credential | TC\_002 |
| TC\_003 | Varify login page elements | GUI | varify that all elements are availabe on login page | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  Rbgondaliya@gmail.com |  |
| 3 | Enter valid Password in Password field | Password field should be editable and accept the password and display as star or dot | Password input displayed in dot and accepted | pass |  | Password: rbgondaliya |  |
| 4 | Enter valid captcha code in captch field | Captch field should editable and accept captcha and captcha is case sensitive | Captcha input accepted | Pass | Step required when human action validation perform | get captcha from image which is near by captcha field |  |
| 5 | Click on login button | User should login into site and navigated to dashboard | User navigated to dashboard and username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to Login with invalid Username and invalid Password | Should be display an error message enter wrong username or password | Display an error of wrong username and password | Pass |  |  |  |
| 2 | Verify that User is not able to Login with Valid Username and invalid Password | Should be display an error message enter wrong password | Display an error of wrong password | Pass |  |  |  |
| 3 | Verify that User is not able to Login with invalid Username and Valid Password | Should be display an error message User not found | Display an error Username not found | Pass |  |  |  |
| 4 | Verify that User is not able to Login with blank Username or Password | Set required field validation message for Username and Password | Display an error of wrong username and password | Fail | Not performa a validation function fix it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify login page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and Create an account link. | All listed control displayed properly on the page | Login page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, Password has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |
| 7 | verify that forgot password link working properly | when click on forgot password load forgot passworg page | forgot password link not working | Fail |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml