**Software Requirement and Design**

**Specifications Report**

**GYM WORKOUT AND REGISTERATION SYTEM**

|  |  |
| --- | --- |
| Course Code | CS 3004 |
| Instructor | Sir Abdul Rehman |
| Project Team | MUHAMMAD UMER 20K-0225  ANAS ALI 20K-0181  ZEESHAN AIJAZ 20K-0361 |
| Submission Date | 05/Dec/2022 |

**[Instructions]**

* No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.
* All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.
* This’ Instruction’ section should also be removed in final document.

Table of Contents

1. INTRODUCTION 5

* 1. Purpose of Document 5
  2. Intended Audience 5

2. OVERALL SYSTEM DESCRIPTION 6

* 1. Project Background 6
  2. Project Scope 6
  3. Not In Scope 6
  4. Project Objectives 6
  5. Stakeholders 6
  6. Operating Environment 6
  7. System Constraints 6
  8. Assumptions & Dependencies 6

3. EXTERNAL INTERFACE REQUIREMENTS 7

* 1. Hardware Interfaces 7
  2. Software Interfaces 7
  3. Communications Interfaces 7

4. FUNCTIONAL REQUIREMENTS 8

* 1. FUNCTIONAL HIERARCHY 8
  2. Use Cases 8

4.2.1. [Title of use case] 8

5. NON-FUNCTIONAL REQUIREMENTS 9

* 1. Performance Requirements 9
  2. Safety Requirements 9
  3. Security Requirements 9
  4. User Documentation 9

SDS 10

1. **SYSTEM ARCHITECTURE 11**
   1. SYSTEM LEVEL ARCHITECTURE 11
   2. SOFTWARE ARCHITECTURE 11
2. **DESIGN STRATEGY 12**
3. **DETAILED SYSTEM DESIGN 13**
   1. DATABASE DESIGN 13
4. APPLICATION DESIGN 15
5. REFERENCES 15
6. APPENDICES 17

# 1. Introduction

### 1.1. Purpose of Document

The purpose of this document is to provide a detailed view of the project we have been working on.

### 1.2. Intended Audience

This application is built keeping in view the gym freak youngsters, body builders and new users.

#### [Definition of Terms, Acronyms and Abbreviations

[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]

|  |  |
| --- | --- |
| **Term** | **Description** |
| ASP | Active Server Pages |
| DD | Design Specification |
| UCD | USE CASE DIAGRAM |
| URL | UNIFORM RESOURCE LOCATOR |
|  |  |
|  |  |
|  |  |
|  |  |

##### 1.3 Document Convention

Font style: Arial.

Font size: 10

Font decoration: none.

For headings: (As given) Font style: Arial.

Font size: 12

Font decoration: Italic, and Bold.

# 2. Overall System Description

### 2.1. Project Background

The idea behind this project is to provide an easiest and free way to get a person register in gym. Storing the huge data of members efficiently. Handle all the payments of members efficiently and have a quick search of all the members’ information.

### 2.2. Project Scope

### This project is a fitness cum gym application built on Flutter. This will be used by three type of users; 1. Admins, 2. Members, 3. Non-Members.

### There will be preset Admins, who will be allowed to sign-in with their own email and password, stored in the database, and no other email and password will be accepted. Admin can register members and trainers and can search all the members with the help of this app. Admin can also mark attendance of members coming into the gym and can have a check on fees payment of the members and can mark the fees paid if member pays the amount. Existing users registered as Members will be allowed to sign-in as Members with their own email and password. Their data will be stored in the database as well. Member can calculate his/her BMI, can view, and update his/her information.

### Non-members users, who are not registered, can also use the app to view the gym packages provided by the trainers.

### 2.3. Not In Scope

Not Applicable.

### 2.4. Project Objectives

In today’s world, people have started to rely more on applications. They just need everything to be done within few taps on the screen. Instead of going to the gym and stand for hours just to get yourself registered, this application will help users to get registered anywhere anytime within seconds.

This application is absolutely free, with easiest UI and interface possible so that people with least knowledge of technology can also operate it.

### 

### 2.5. Stakeholders

* The admin, who can perform various operations that no one else is authorized to do.\
* Developer who developed the project, for further releases and new versions (Both front end, and back end).
* Internal database engineer who manages the database working.

### 2.6. Operating Environment

**Hardware platform:** It needs pretty basic hardware requirements. Any average system will do the work.

The hardware we used to run this is mentioned below:

* Core i5 6th Generation
* 8gb ram (hardly uses 1 or 2gb(s) of ram for all the softwares we used) - Less than 200mb of disk space for project, and related data.

**Operating System:** Windows 10 any variant (Recommended)

Network Environment: Should have a decent internet connection.

**Applications:**

❏ Visual Studio Code

❏ Android Studio

❏ Flutter

❏ Supabase

❏ MySql Server

### 2.7. System Constraints

* **Software constraints:**

This system doesn’t require any software constraints so far.

* **Hardware constraints:**

It needs pretty basic hardware requirements. Any average system will do the work. The hardware we used to run this is mentioned below:

* Core i5 8th Generation
* 16gb ram (hardly uses 1 or 2gb(s) of ram for all the softwares we used)
* Less than 200mb of disk space for project and its dependencies
* **Cultural constraints:**

The person should know how to read, and understand basic English language.

* **Legal constraints:**

Not applicable.

* **Environmental constraints:**
* There are no environmental constraints. The system can be used anywhere anytime.
* The project is developed for people of all ages and ethnicities
* **User constraints:**

16+. People below 16 are not advised to use this application because they are not allowed to get register in gym either.

**2.8. Assumptions & Dependencies Assumptions**:

We assume that people with some knowledge of gym will use this application.

Admin will have to add column to the database for every new day to mark member’s attendance of that day.

# 3. External Interface Requirements

[This section is intended to specify any requirements that ensure that the new system will connect properly to external components. Place a context diagram showing the external interfaces at a high level of abstraction.]

### 3.1. Hardware Interfaces

The system is perfectly supported by desktop computers, laptops, tablets and mobile phones as well.

The type of data we are taking from the user are login credentials, all the basic information e.g. Name, contact, gender, package etc. which is stored in database.

Processor: Intel

Installed Memory: 2 GB or Higher

Speed: 1.40 GHz or Higher

Operating System: 32/64-bit Operating System

### 3.2. Software Interfaces

**Operating System:** Windows 11

**Database:** Supabase

**Webserver**: XAMPP

**Web Technologies**: HTMLL/ CSS/ JS/ FLASK

**Ide and tools:** Visual Studio Code, Android Studio, Flutter

#### Communications Interfaces

* User email for signup should be valid.
* No encryption usage, or data transfer or synchronization issues.

# 4. Non-functional Requirements

### 4.1. Performance Requirements

1. Performance wise, our project is built to be very responsive and fast. The transitions between the interfaces take no time.
2. Capacity wise, our system is very storage friendly, hardly requires some Mbs of data.
3. Safety wise, the data of users, interacting with our system stays safe and can only be accessed by the system owner.
4. The software is reliable in a sense that it fulfills all the needs that it is promised to fulfill. It was tested for any sort of bugs/ issues and was fixed by the developers eventually.

### 4.2. Safety Requirements

1. We took extra care that our system must not cause any damage on the machine on which the user is running our system.
2. The only thing that the user should take care of is the entry of dummy (fake) results in the database. Dummy data must be deleted from the database.

### 4.3. Security Requirements

1. External users such as someone out of the organization must not be given access to the system’s Admin panel. Login ensures this.
2. Only the Stakeholders should have access to the system.
3. The data of the user stays safe and untouchable. So, privacy is maintained.

## 4.4. Software Quality Attributes

No additional quality attributes.

**Other Requirements:**

# 5. Functional Requirements

##### 5.1. Functional Hierarchy

##### 5.2. Use Cases

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Gym workout and registration system |
| **Use Case Description:** | |
| **Primary actor:** admin | **Other actors:** admin, member, non-member |
|  |  |
| **Pre-conditions:**  ▪ customer must have an adequate internet connection, and a desktop computer to run the webpage, and he should signup before moving ahead. | |
|  | |

#### 6. System Architecture

The architecture embodies the major static and dynamic aspects of a system. It is a view of the whole system highlighting the important characteristics and ignoring unnecessary details. In the context of our approach, architecture is primarily specified in terms of views of tier architecture which is a client-server architecture in which the presentation, the application processing and data management are logically separate processes.

**6.2. Software Architecture**

**User Interface Layer:** not applicable

**Middle tier:** not applicable

#### Data access Layer: not applicable

**Functions:**

**ADMIN**

* + Signup – takes customer information, to register customer as a user.
  + Login - takes customer information, and verify the information from the database. If the information matches, customer information is loaded, which is used when the customer places order. It prevents asking for repeated information from the user.
  + Attendance Marking
  + Registration
  + Fee Payment
  + Updation, insertion, deletion of information from database

## Member:

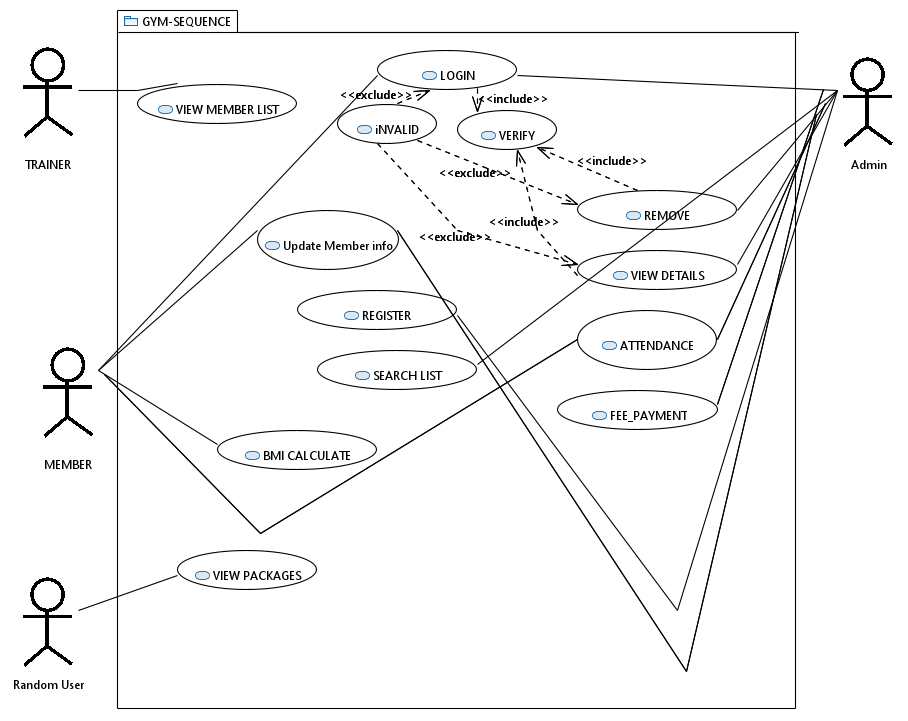
* Update Information
* Sign-in
* BMI Calculate

**Random User:**

* View packages

## 8.1. Database Design

**8.1.1. USE-CASE**



**CLASS DIAGRAM**

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

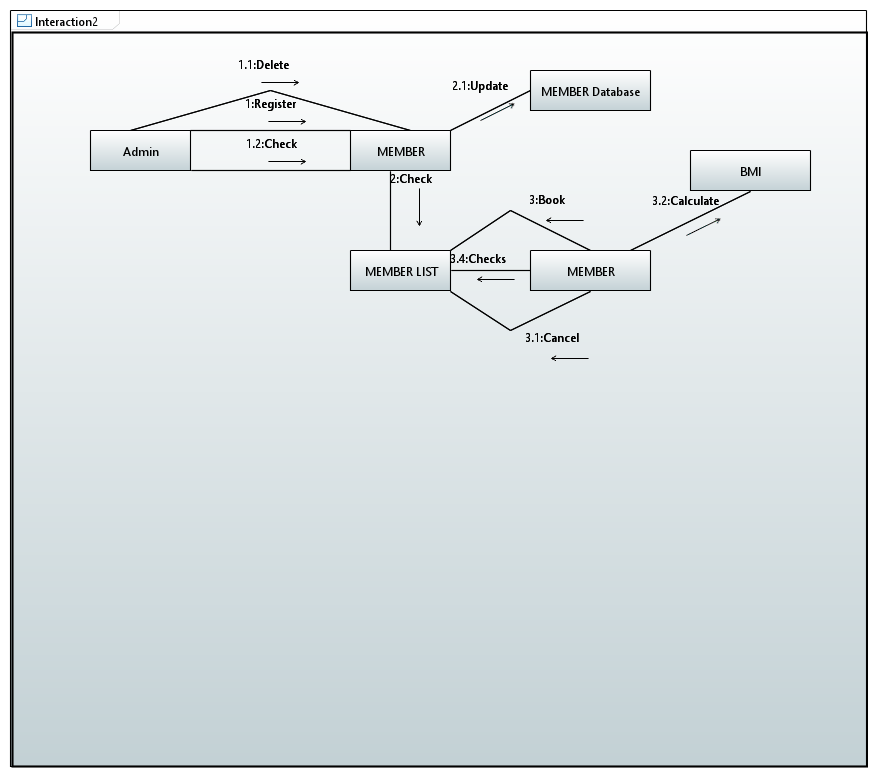
**ACTIVITY DIAGRAM**

**CLASS DIAGRAM**

Diagram

Description automatically generated

**COLLABORATION DIAGRAM**



**COMPONENT DIAGRAM**

Diagram, schematic

Description automatically generated

**ERD**

Diagram

Description automatically generated

**STATE DIAGRAM**

Diagram

Description automatically generated

**SEQUENCE DIAGRAM**

Diagram

Description automatically generated

