Software Requirements

Specification

GYMNASIO

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

**Table of Contents:**

[**1. Introduction 4**](#_e1lxijhnpyn7)

[1.1 Purpose 4](#_qjt3v2pixsc7)

[1.2 Scope 4](#_bn2hlwt9va30)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_c2fl1nqb3laz)

[**2. Overall Description 5**](#_3b3udi19zzmv)

[2.1 Product Perspective 5](#_5k3c0vlhhoh)

[2.2 Product Functions 5](#_n0lbqw95cje7)

[2.3 User Characteristics 6](#_g4nom55n2ib3)

[2.4 Constraints 6](#_tffu6eujqkpy)

[2.5 Assumptions 7](#_7wgyyyl53mri)

[**3. Specific Requirements 7**](#_wzj86bcb9774)

[3.1- Functional Requirements 7](#_h74ax1zfqtl9)

3[.2 Performance Requirements 14](#_ycezmdu9ztru)

[3.3 Reliability 15](#_wgkc7gln18eb)

[3.4 Availability 15](#_kq32yrsxugfo)

[3.5 Security 15](#_bk30nsv63aq3)

[3.6 Maintainability 15](#_w74i5m6l0mu0)

[3.7 Portability 16](#_salyqdjwngq)

[**4. Storyboard 16**](#_l6qpqlm1awne)

[**5. Plan 17**](#_5n1hwgn6wf5n)

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to define the software requirements for a web-based application designed for gym members. The application will allow users to access and manage their personal membership details, track their gym visits, and gain insights into their attendance frequency. Additionally, users will be able to view popular classes and identify peak gym usage times. The system will also provide a reservation feature that enables members to book specific classes and manage their existing reservations. Furthermore, the web application will include a secure payment interface that allows users to pay for classes and manage their financial transactions efficiently. This document serves as a guide for the development team and a reference for stakeholders to ensure that the final product meets the needs of gym members and administrators.

### 1.2 Scope

The Gymnasio application is designed as a web-based platform to enhance the gym experience for members by providing them with real-time access to essential data and features. The application will allow users to review their membership details, track the frequency of their gym visits, and explore analytics related to class popularity and gym traffic. Members will also have the ability to reserve spots in classes and manage their bookings through an intuitive interface. Additionally, a secure payment system will be integrated to facilitate seamless transactions for class payments and other services.

The system will be developed using HTML, CSS, and JavaScript for the frontend, Java for the backend, and SQL for the database. The application will be accessible via a web browser and hosted on a central server, ensuring a smooth and secure user experience.

### **1.3 Definitions, Acronyms, and Abbreviations**

| **Term** | **Definition** |
| --- | --- |
| User | Someone who interacts with the mobile phone application |
| Admin/Administrator | System administrator who is given specific permission for managing and controlling the system |
| Stakeholder | Any person who has interaction with the system who is not a developer. |
| DESC | Description |
| DEP | Dependency |
| TAG | A unique, persistent identifier contained in a PLanguage statement [2] |
| GIST | A short, simple description of the concept contained in a PLanguage statement [2] |
| SCALE | The scale of measure used by the requirement contained in a PLanguage statement [2] |
| METER | The process or device used to establish location on a SCALE contained in a PLanguage statement [2] |
| MUST | The minimum level required to avoid failure contained in a PLanguage statement [2] |
| PLAN | The level at which good success can be claimed contained in a PLanguage statement [2] |
| WISH | A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement [2] |
| DEFINED | The official definition of a term contained in a PLanguage statement [2] |

## **2. Overall Description**

### 2.1 Product Perspective

The Gym Analytics Web Application is designed to provide gym members with an all-in-one digital solution to manage their membership details, track gym attendance, and engage with available classes. The system will replace the need for manual tracking by offering a centralized and user-friendly web portal. Members will have the ability to access real-time insights regarding their gym visits, preferred workout schedules, and class reservations.

Additionally, the application will feature an integrated reservation system, allowing members to book, cancel, and modify class reservations. It will also include a secure payment interface that facilitates seamless payments for classes and membership-related transactions. Administrators will have access to manage system settings, oversee bookings, and ensure proper functionality of the application.

### 2.2 Product Functions

The system will offer a variety of functionalities aimed at improving user engagement and gym management efficiency. Members will be able to:

* View their membership details, including plan type, renewal date, and visit history.
* Track their frequency of gym visits with statistical insights and analytics.
* Explore popular classes and view schedules to make informed booking decisions.
* Reserve, modify, and cancel class bookings through an intuitive interface.
* Make secure payments for their reservations and other transactions.

Additionally, the system will provide real-time updates on class availability, membership status, and transaction history. It will also generate reports on gym usage trends and revenue analytics to assist administrators in optimizing operations.

### 2.3 User Characteristics

There are three primary types of users for the system:

1. **Gym Members**
   * Require access to personal gym data and reservation features.
   * Need the ability to track visits, make class bookings, and manage payments.
2. **Gym Staff Members**
   * Assist members with their bookings and provide support for gym operations.
   * May have limited administrative privileges compared to managers.
3. **Administrators**
   * Oversee system operations, user management, and financial transactions.
   * Maintain data integrity and ensure accurate reporting on class attendance and revenue.

All users are expected to have basic digital literacy and familiarity with web-based interfaces.

### 2.4 Constraints

1. **Simulated Payment Processing:** The payment interface will function for testing and demonstration purposes only. Since the system is not being developed for a real-world client, it will not support actual financial transactions. Instead, it will simulate the process of payments and confirmations without real-time banking integrations.
2. **Manual Data Input:** Class schedules, membership details, and other data must be manually entered by administrators or gym staff. The system does not support automated integration with external scheduling or membership management platforms.
3. **Limited External Integration:** The initial version of the system will not integrate with third-party fitness tracking apps, gym equipment monitoring systems, or customer relationship management (CRM) software.
4. **Internet Dependency:** As a web-based application, the system requires an active internet connection to function. Users will not have access to their membership details, class schedules, or payment history while offline.
5. **Basic Security Measures:** While the system will implement secure authentication (e.g., hashed passwords), it will not include advanced security features such as multi-factor authentication (MFA) or end-to-end encryption for communications.
6. **Limited Scalability in Initial Version:** The first iteration of the system is designed for small to medium-sized gyms. Expansion to support multiple branches or franchise-wide management would require additional development phases.
7. **Static User Roles:** User roles (e.g., gym members, staff, administrators) will be predefined, and there will be no role customization in the initial release. Any changes to user permissions will require manual adjustments by an administrator.

### 2.5 Assumptions

1. **Users Have Stable Internet Access:** It is assumed that gym members, staff, and administrators will have a stable internet connection to access the system since it is entirely web-based.
2. **Users Will Enter Accurate Data:** The system assumes that gym members and administrators will input correct and up-to-date information, including membership details, class schedules, and payment records.
3. **All Users Have Basic Digital Literacy:** Gym members and staff are assumed to have basic proficiency in navigating web applications, managing online reservations, and making online payments.
4. **The Gym Will Manually Update Class Schedules:** Since there is no automated synchronization with external scheduling platforms, it is assumed that gym staff will be responsible for updating and maintaining class schedules regularly.
5. **Payments Are for Demonstration Purposes Only:** It is assumed that users understand the payment interface is only for functional testing and does not process real transactions.
6. **Limited Administrative Staff:** The system assumes that only a small number of administrators will be managing user accounts and class schedules, reducing the complexity of role-based access management.
7. **Data Storage is Sufficient for Expected Usage:** The system assumes that the SQL database will have adequate storage capacity to manage gym membership records, reservation logs, and payment histories without immediate scalability concerns.

## 3. Specific Requirements

### 3.1- Functional Requirements

#### 3.1.1 User Authentication

##### 3.1.1.1 User Registration

**ID:FR1  
TITLE:** User Registration  
**DESC:** The system must allow users to create an account by providing a username, email, and password. The system must validate whether the email is unique before allowing registration. The system must encrypt passwords before storing them in the database using the BCrypt hashing algorithm. The system must send a verification email upon successful registration. The system must generate a unique verification token and include it in the email. The system must provide a verification link in the email that, when clicked, confirms the account and updates the database. The system must restrict login attempts until email verification is completed.  
**DEP:** None

##### 3.1.1.2 User Login

**ID:FR2  
TITLE:** User Login  
**DESC:** The system must authenticate users by verifying their email and password against the stored credentials. The system must display an error message when incorrect login credentials are entered. The system must lock the account for five minutes after three consecutive failed login attempts. The system must redirect logged-in users to their respective dashboards based on their roles. The system must maintain session-based authentication using JSON Web Tokens (JWT).  
**DEP:** FR1

##### 3.1.1.3 Password Encryption

**ID:FR3  
TITLE:** Password Encryption  
**DESC:** The system must use BCrypt hashing with a salt factor of 12 to securely store user passwords. The system must not store plaintext passwords in the database. The system must validate passwords against the stored hash during login attempts. The system must generate a new hash when a user updates their password.  
**DEP:** FR1

##### 3.1.1.4 Email Verification

**ID:FR4  
TITLE:** Email Verification  
**DESC:** The system must generate a unique verification token when a user registers. The system must store the verification token with an expiration period of 24 hours. The system must send an email with a verification link containing the token. The system must validate the token when the verification link is clicked. The system must update the database to mark the email as verified upon successful validation. The system must notify users if their verification token has expired and allow them to request a new one.  
**DEP:** FR1

##### 3.1.1.5 Password Reset

**ID:FR5  
TITLE:** Password Reset  
**DESC:** Users must be able to request a password reset by providing their registered email. The system must generate a secure one-time token for password reset. The system must send a password reset link to the user’s email. The system must verify the token when the user clicks the reset link. The system must allow users to set a new password after verification. The system must use BCrypt hashing to store the new password. The system must invalidate the reset token after use.  
**DEP:** FR1

#### 3.1.2 User Profile Management

##### 3.1.2.1 Edit Profile Information

**ID: FR6  
TITLE:** Edit Profile Information  
**DESC:** The system must allow users to edit their personal details, including their full name, email, and phone number. The system must validate the email format before saving changes. The system must verify that the phone number follows a valid format. The system must display a confirmation message after successful updates.  
**DEP:** FR1

##### 3.1.2.2 Upload Profile Picture

**ID:FR7  
TITLE:** Upload Profile Picture  
**DESC:** The system must allow users to upload a profile picture in supported formats (JPG, PNG). The system must restrict the file size to a maximum of 5MB. The system must display a preview of the uploaded image before confirming the update. The system must replace the previous profile picture with the newly uploaded one.  
**DEP:** FR1

##### 3.1.2.3 Change Password

**ID: FR8  
TITLE:** Change Password  
**DESC:** The system must allow users to change their password by providing the current password and entering a new password. The system must verify that the new password meets complexity requirements (minimum 8 characters, at least one number, one uppercase letter, and one special character). The system must store the new password securely using BCrypt hashing. The system must log the password change and notify the user via email.  
**DEP:** FR3

##### 3.1.2.4 View Profile Details

**ID:FR9  
TITLE:** View Profile Details  
**DESC:** The system must allow users to view their profile information, including full name, email, phone number, and membership details. The system must display the current profile picture alongside the profile information. The system must restrict modifications from this view and provide an "Edit" button to update profile details.  
**DEP:** FR1

#### 3.1.3 Reservation System

##### 3.1.3.1 View Available Services and Classes

**ID: FR10  
TITLE:** View Available Services and Classes  
**DESC:** The system must display a list of all available services and classes, including descriptions, pricing, and schedules. The system must allow users to filter services based on category, date, and instructor. The system must show real-time availability for each class or service.  
**DEP:** None

##### 3.1.3.2 Select Service and Schedule Reservation

**ID: FR11  
TITLE:** Select Service and Schedule Reservation  
**DESC:** The system must allow users to select a service or class and choose a preferred date and time. The system must validate that the selected time slot is available before proceeding. The system must prevent double booking by ensuring that the user does not reserve overlapping sessions.  
**DEP:** FR10

##### 3.1.3.3 Confirm Reservation

**ID: FR12  
TITLE:** Confirm Reservation  
**DESC:** The system must prompt the user to confirm their reservation details before finalizing. The system must generate a unique reservation ID and store the details in the database. The system must send a confirmation email to the user upon successful reservation.  
**DEP:** FR11

##### 3.1.3.4 View Reservation History

**ID: FR13  
TITLE:** View Reservation History  
**DESC:** The system must allow users to view their past and upcoming reservations. The system must display reservation details, including date, time, and service type. The system must allow users to filter reservations based on status (completed, upcoming, canceled).  
**DEP:** FR12

##### 3.1.3.5 Cancel Reservation

**ID: FR14  
TITLE:** Cancel Reservation  
**DESC:** The system must allow users to cancel an upcoming reservation. The system must verify if the cancellation policy permits a refund or penalty. The system must update the reservation status in the database and notify the user via email upon cancellation.  
**DEP:** FR13

##### 3.1.3.6 Track Popular Activities

**ID:FR15  
TITLE:** Track Popular Activities  
**DESC:** The system must track and display data on the most frequently booked services and classes. The system must generate reports on trends over weekly, monthly, and yearly periods. The system must allow administrators to view analytics and adjust service offerings accordingly.  
**DEP:** FR10

#### 3.1.4 Payment System

##### 3.1.4.1 Enter Payment Details

**ID**: FR16

**TITLE**: Enter Payment Details

**DESC:** The system must allow users to enter their payment details, including card number, expiration date, and CVV. The system must validate the entered information before proceeding.

**DEP:** None

##### 3.1.4.2 Process Simulated Payment

**ID:** FR17

**TITLE:** Process Simulated Payment

**DESC:** The system must simulate the processing of payments without conducting real transactions. The system must validate the entered card details, including the format of the card number, expiration date, and CVV. If any of these inputs are invalid, the system must display an error message and prevent further processing.

The system must integrate with Stripe’s sandbox environment to test payment transactions without real financial interactions. Stripe provides a test mode that allows developers to use predefined test card numbers from Visa and MasterCard to simulate various transaction scenarios, including successful payments, declined payments, and insufficient funds.

The system must implement a transaction validation process that follows these steps:

1. **Validate Card Details** – Ensure that the card number follows a valid structure, the expiration date is in the future, and the CVV is the correct length.
2. **Simulate Payment Approval or Decline** – Use Stripe’s test environment to process transactions based on test cases that determine whether a transaction is approved or declined (e.g., insufficient funds, expired card, or invalid CVV).
3. **Error Handling and Messaging** – If an error occurs during processing (such as an invalid card number or network failure), the system must provide a clear error message and allow the user to retry.
4. **Log Transaction Attempts** – All simulated transactions must be logged in the database, including success and failure details, for debugging and reporting purposes.
5. **Send Confirmation Notification** – Upon completion, the system must send a confirmation email to the user indicating whether the transaction was successful or failed.
6. For testing purposes, the system must allow developers to use Stripe’s test card numbers, such as:

* Visa: 4242 4242 4242 4242
* MasterCard: 5555 5555 5555 4444
* Declined Transaction Simulation: 4000 0000 0000 0002
* Insufficient Funds: 4000 0000 0000 9995

These test card numbers must work exclusively within Stripe’s test mode and must not be used for real transactions.

DEP: FR16

##### 3.1.4.3 View Payment History

**ID:** FR18

**TITLE:** View Payment History

**DESC:** The system must allow users to view a record of their past payments, including date, amount, and status.

**DEP:** FR17

#### 3.1.5 Account Balance Management

##### 3.1.5.1 View Current Balance

**ID:** FR19

**TITLE:** View Current Balance

**DESC:** The system must allow users to view their current account balance. The system must display the balance in real-time, reflecting any recent transactions. The system must ensure that the balance updates dynamically whenever a payment or refund is processed.

**DEP:** None

##### 3.1.5.2 Add Funds to Balance

**ID:** FR20

**TITLE:** Add Funds to Balance

**DESC:** The system must allow users to add funds to their balance using the integrated payment system. The system must validate the payment details before proceeding. Upon successful processing of the transaction, the system must update the user's balance accordingly. The system must provide a confirmation message upon completion.

**DEP:** FR17

##### 3.1.5.3 Withdraw Funds from Balance

**ID:** FR21

**TITLE:** Withdraw Funds from Balance

**DESC:** The system must allow users to request withdrawals from their balance. The system must verify that the user has sufficient funds before allowing the withdrawal. The system must log the withdrawal request and update the balance once the transaction is processed. The system must notify the user via email when the withdrawal is successful.

**DEP:** FR19

##### 3.1.5.4 Track Transaction History

**ID:** FR22

**TITLE:** Track Transaction History

**DESC:** The system must provide users with a detailed transaction history, including deposits, withdrawals, and payments. The system must allow users to filter transactions by date range and transaction type. The system must store transaction logs for at least six months for record-keeping and troubleshooting purposes.

**DEP:** FR19

#### 3.1.6 Navigation and Accessibility

##### 3.1.6.1 Access Client Menu

**ID:** FR23

**TITLE:** Access Client Menu

**DESC:** The system must provide users with a structured client menu that allows them to access different sections of the application, such as profile management, reservations, balance, and payments. The menu must be responsive and accessible across different screen sizes.

**DEP:** None

##### 3.1.6.2 Navigate Between Pages

**ID:** FR24

**TITLE:** Navigate Between Pages

**DESC:** The system must allow users to seamlessly navigate between different sections of the application through the client menu and quick-access buttons. The navigation must be smooth, with minimal loading times.

**DEP:** FR23

##### 3.1.6.3 Logout Functionality

**ID:** FR25

**TITLE:** Logout Functionality

**DESC:** The system must provide users with the ability to log out securely from their account. The logout process must invalidate the user's session and redirect them to the login page. If the user attempts to access restricted content after logging out, they must be redirected back to the login page.

**DEP:** FR2

##### 3.1.6.4 Accessibility Features

**ID:** FR26

**TITLE:** Accessibility Features

**DESC:** The system must include accessibility options such as high-contrast mode, keyboard navigation support, and screen reader compatibility. The system must allow users to enable or disable accessibility settings based on their needs.

**DEP:** None

### 3.2 Performance Requirements

##### 3.2.1 Response Time

**ID**: QR1

**TITLE:** Response Time

**DESC:** The system must ensure that all user interactions, including logging in, dashboard access, reservations, and payments, respond within 2 seconds under normal load. Under peak load, response times should not exceed 5 seconds.

**DEP**: None

##### 3.2.2 System Scalability

**ID:** QR2

**TITLE:** System Scalability

**DESC:** The system should support 200 concurrent users on a Tomcat server hosted on a PC. Performance optimizations should allow scalability to 500 users through caching and indexing strategies.

**DEP:** None

##### 3.2.3 Database Performance

**ID:** QR3

**TITLE:** Database Performance

**DESC:** Database queries for user authentication, dashboard access, and payment processing must execute within 1 second under normal conditions. Caching and indexing must be utilized to ensure performance consistency.

**DEP:** None

##### 3.2.4 Payment Processing Performance

**ID:** QR4

**TITLE:** Payment Processing Performance

**DESC**: Simulated payments using Stripe’s sandbox must complete within 3 seconds. Failed transactions should be identified and flagged within 1 second.

**DEP:** QR1

##### 3.2.5 System Uptime

**ID:** QR5

**TITLE:** System Uptime

**DESC:** The system must maintain at least 98% uptime on a local Tomcat-hosted PC. Downtime due to system maintenance must be scheduled during off-peak hours.

**DEP:** None

##### 3.2.6 Mobile and Browser Compatibility

**ID:** QR6

**TITLE:** Mobile and Browser Compatibility

**DESC:** The system must be fully responsive and optimized for the latest versions of Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. Mobile support must include Android and iOS devices.

**DEP:** None

### 3.3 Reliability

##### 3.3.1 System Reliability

ID: QR7

TITLE: System Reliability

DESC: The system must return accurate search and user data 98% of the time, based on 1000 test cases.

DEP: None

### 3.4 Availability

##### 3.4.1 System Availability

**ID:** QR8

**TITLE:** System Availability

DESC: The system must be available **98% of the time**, excluding planned maintenance.

DEP: None

##### 3.4.2 Internet Connection

ID: QR9

TITLE: Internet Connection

DESC: The system must maintain a stable internet connection to access the database and complete transactions.

DEP: None

### 3.5 Security

##### 3.5.1 Secure Communication

**ID**: QR10

**TITLE:** Secure Communication

DESC: All login credentials and sensitive user data must be encrypted using BCrypt hashing algorithm to prevent unauthorized interception.

**DEP:** None

##### 3.5.2 Account Security

**ID:** QR11

**TITLE:** Account Security

**DESC:** Users attempting to log in with incorrect credentials three times must be locked out for 30 minutes.

**DEP:** None

### 3.6 Maintainability

##### 3.6.1 Code Maintainability

**ID:** QR12

**TITLE:** Code Maintainability

**DESC:** The system must be structured using modular components to facilitate future expansion and bug fixes.

**DEP:** None

##### 3.6.2 Testability

**ID:** QR13

**TITLE:** Testability

**DESC:** The system must provide a test environment where new features can be validated before deployment.

**DEP:** None

### 3.7 Portability

##### 3.7.1 System Portability

**ID:** QR14

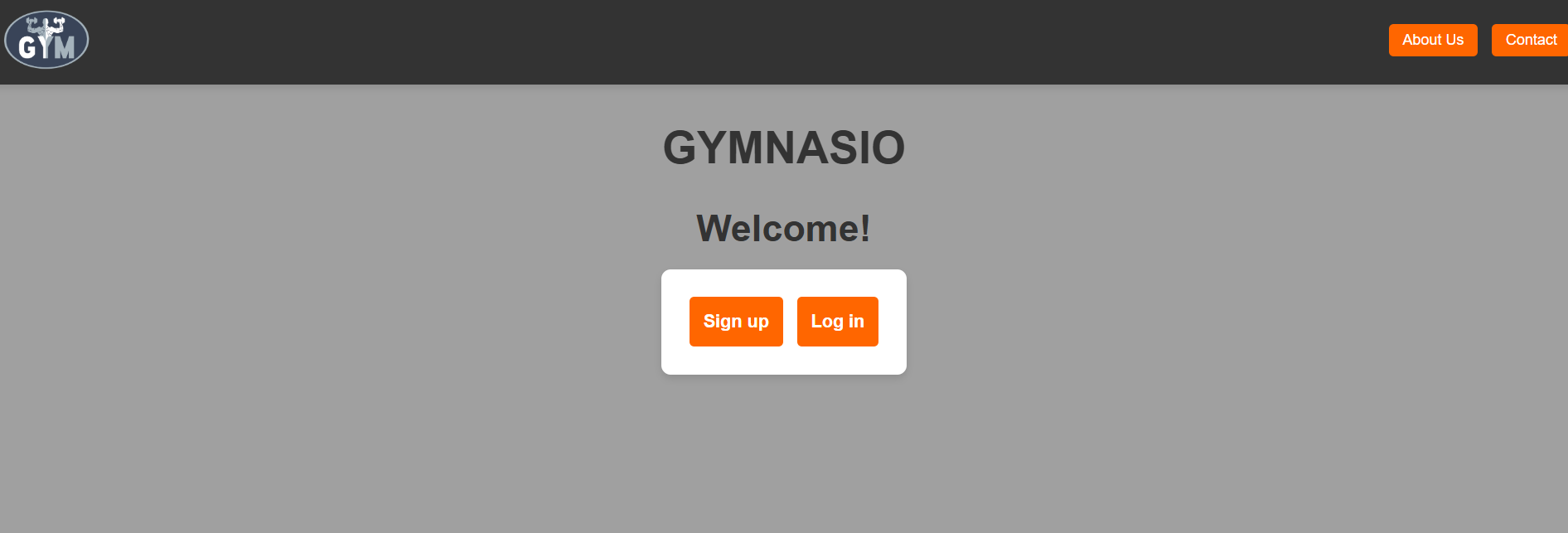
**TITLE:** System Portability

**DESC:** The system should be portable across Windows, Linux, and macOS environments with minimal changes to configuration settings.

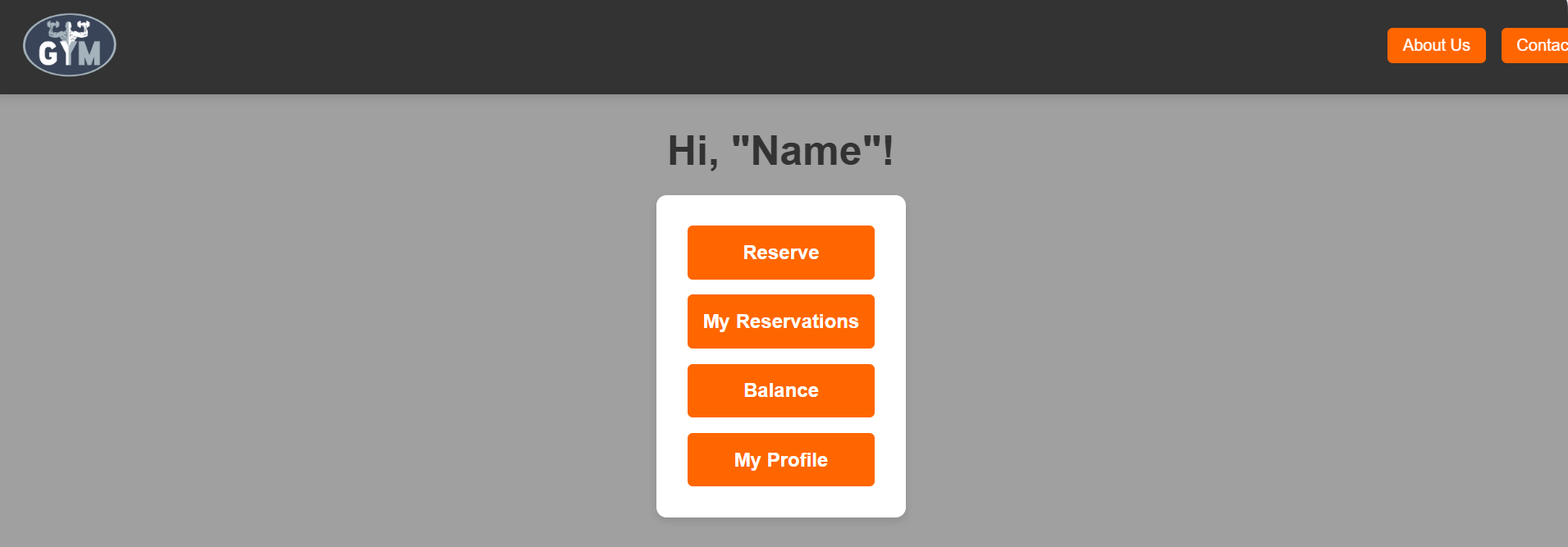
**DEP:** None

## 4. Storyboard

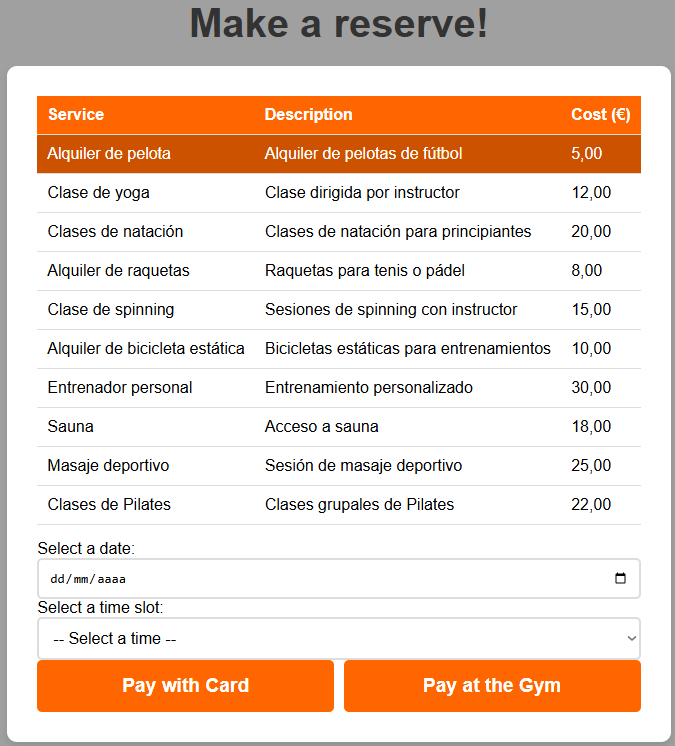
The storyboard for the gym website presents an elegant and structured flow that ensures a seamless user experience. The homepage (Index.html) begins with a well-defined header that contains two essential buttons. One provides an overview of the gym’s identity, detailing its philosophy, facilities, and values, while the other is a contact button that allows users to connect directly. This button offers two functionalities: calling the gym’s phone number with a simple tap and sending an email via an integrated form. The form requires users to enter their name, email address, and a message, ensuring a direct communication channel between the customer and the gym’s administration.

 *index*

The main section of the homepage features two additional buttons, one for registration and another for logging in. The registration button directs new users to a form where they must input their name, email, phone number, national identification number (DNI), and the date they wish to start their membership. The login button, on the other hand, leads to a page where existing users can enter their email and password. A visibility toggle allows them to view their entered password, and if forgotten, a 'Forgot Password?' link provides a simple recovery process. Upon clicking the link, users are prompted to enter their email to receive a verification code. Once the code is submitted and verified, they can reset their password and return to the login page.

*ClientMenu*

Upon successful login, users are directed to the Client Menu, an intuitive dashboard offering various functionalities. One of the primary features is the Reservation system. Users can browse through the available gym activities, select their preferred session, and specify the date and time they wish to attend. Two payment options are available: paying via card or paying at the gym counter. If a user opts for card payment, a secure interface appears where they must enter their card details, including the card number, expiration date, and CVV. A confirmation message is displayed upon successful payment. Alternatively, if they choose to pay at the gym, they are redirected back to the Client Menu without additional steps.

 *ReservationMenu*

The Client Menu also includes the 'My Reservations' section, where users can track their scheduled activities. A table displays the booked services, the corresponding dates and times, and the payment status, indicating whether the session has been paid for. If payment is pending, an action button allows users to complete the payment directly from this page. Below this, another table summarises activities completed within the last 30 days, offering a detailed history without the payment option. Additionally, a graphical representation provides insights into the user’s activity frequency, visually illustrating participation trends over time.

Another key feature within the Client Menu is the Balance section. This interface presents the user’s gym account balance, displaying available funds with two interactive buttons: 'Add' for depositing money and 'Cash Out' for withdrawing funds. Users can specify the desired amount for either action, ensuring financial flexibility and control over their gym-related expenses.

The final component of the Client Menu is the 'My Profile' section, designed to provide users with personalisation options. Here, members can view and update their profile details, including their profile picture, username, phone number, and password. The DNI remains visible but non-editable, ensuring identity consistency. A dedicated button allows users to return to the Client Menu effortlessly, completing the user experience with a seamless transition between sections.

This storyboard reflects an intuitive, sophisticated, and user-centric approach, ensuring that each interaction within the gym website is both efficient and engaging.

## 5. Plan

| Requirement | Rishi | Alvaro | Jorge | Mikel |
| --- | --- | --- | --- | --- |
| FR1 | 0 | 1 | 0 | 1 |
| FR2 | 0 | 2 | 0 | 1 |
| FR3 | 0 | 1 | 0 | 0 |
| FR4 | 0 | 1 | 0 | 0 |
| FR5 | 0 | 2 | 0 | 0 |
| FR6 | 0 | 1 | 0 | 2 |
| FR7 | 0 | 1 | 0 | 2 |
| FR8 | 0 | 1 | 1 | 3 |
| FR9 | 0 | 1 | 1 | 3 |
| FR10 | 0 | 1 | 1 | 3 |
| FR11 | 0 | 0 | 2 | 0 |
| FR12 | 0 | 0 | 3 | 0 |
| FR13 | 0 | 0 | 1 | 0 |
| FR14 | 0 | 0 | 2 | 0 |
| FR15 | 0 | 0 | 3 | 0 |
| FR16 | 2 | 0 | 1 | 1 |
| FR17 | 3 | 1 | 1 | 1 |
| FR18 | 2 | 1 | 1 | 1 |
| FR19 | 1 | 1 | 1 | 1 |
| FR20 | 2 | 1 | 1 | 1 |
| FR21 | 1 | 1 | 1 | 1 |
| FR22 | 2 | 1 | 1 | 1 |
| FR23 | 1 | 1 | 1 | 1 |
| FR24 | 0 | 1 | 0 | 0 |
| FR25 | 1 | 1 | 1 | 1 |
| FR26 | 1 | 1 | 1 | 1 |