A black background with a black square

Description automatically generated with medium confidence

CSE6224 Software Requirements Engineering

Campus Wellness Portal with

Medical System and Fitness Center

Integration Project – Part 1

SRS Document

Tutorial Section: TT6L

Group 4

|  |  |
| --- | --- |
| **Group member** | **Student ID** |
| Nicholas Thong Meng Shui | 241UC2415Y |
| Muhammad Anas bin Khairul Azman | 241UC2401Z |
| Fikrul Amsyar Azmin | 241UC24167 |

**Table of Contents**

[Changelog Table: 2](#_Toc891453659)

[1. Introduction 5](#_Toc1567889714)

[1.1 Purpose 6](#_Toc80877773)

[1.2 Scope 6](#_Toc2028780717)

[1.3 Product Overview 6](#_Toc653958687)

[1.3.1 Product Perspective 7](#_Toc1262828858)

[1.3.2 Product functions 11](#_Toc1519241159)

[1.3.3 User Characteristics 11](#_Toc495272960)

[1.3.4 Product Limitations 12](#_Toc1744784406)

[2.0 References 12](#_Toc1974997343)

[3.0 Requirements 13](#_Toc71625522)

[3.1 Functional Requirement 13](#_Toc1552771182)

[Context diagram 13](#_Toc401706296)

[Use Case Diagram 14](#_Toc90622189)

[Use Case ID table 14](#_Toc629219590)

[3.1.1Login 15](#_Toc426538219)

[3.1.2Book Fitness Class 17](#_Toc1283431892)

[3.1.3Cancel Fitness Class 19](#_Toc314089293)

[3.1.4Schedule Health Centre Appointment 20](#_Toc588510697)

[3.1.5 Cancel Health Centre Appointment 22](#_Toc1773640513)

[3.1.6 Track Wellness Goals 24](#_Toc93029673)

[3.1.7 View System Notifications 27](#_Toc244002969)

[3.1.8 View Personalized Health Tips 28](#_Toc1814533166)

[3.1.9 Manage Health Centre Appointments 31](#_Toc628128603)

[3.1.10 Manage Fitness Class Booking 32](#_Toc313918574)

[3.1.11Manage User Access Control 35](#_Toc1189493306)

[3.1.12 View system logs 39](#_Toc1689490394)

[3.2 Performance Requirements 42](#_Toc876783399)

[3.3 Usability Requirements 43](#_Toc768332021)

[3.4 Interface Requirements 43](#_Toc169774433)

[3.4.1 System Interfaces 43](#_Toc2132237545)

[3.4.2 User Interface 44](#_Toc1370358017)

[3.4.3 Hardware Interfaces 50](#_Toc452980776)

[3.5 Logical Database Requirements 50](#_Toc2044139160)

[3.6 Design Constraints 50](#_Toc934111564)

[3.7 Software System Attributes 51](#_Toc2787631)

[3.7.1 Security 52](#_Toc1947282053)

[3.8 Supporting Information 52](#_Toc22177955)

[4.0 Verification 53](#_Toc1770675368)

[4.1 Verification Approach 53](#_Toc119961099)

[4.2 Verification Criteria 53](#_Toc1140710821)

[4.2.1 Unit testing verification criteria 54](#_Toc76648252)

[4.2.2 Integration testing verification criteria 54](#_Toc320172732)

[4.2.3 Functional testing verification criteria 55](#_Toc1390400527)

[4.2.4 User acceptance testing (UAT) verification criteria 56](#_Toc1937554312)

[4.2.5 Performance Verification Criteria 56](#_Toc1324735346)

[4.2.6 Usability Verification Criteria 56](#_Toc1349253104)

[5.0 Appendices 57](#_Toc1816687531)

[5.1 Assumptions and Dependencies 57](#_Toc1110922226)

[5.2 Acronyms and Abbreviations 57](#_Toc1703892346)

# Changelog Table:

|  |  |  |
| --- | --- | --- |
| Version | Author | Description |
| v1.0 | Nicholas Thong Meng Shui | * Created this document |
| V1.1 | Nicholas Thong Meng Shui | * Added information and content to 1.3.1 Product Perspective |
| V1.2 | Nicholas Thong Meng Shui | * Added information and content to 1.3.3 User Characteristics |
| V1.3 | Muhammad Anas bin Khairul Azman | * Added content to 1.3.2 Product Functions |
| V1.4 | Nicholas Thong Meng Shui | * Added Use Case Diagram to 3.1 Functions |
| V1.5 | Nicholas Thong Meng Shui | * Removed EHR System from 1.3.1 * Updated 1.3.2 Campus Recreation Facility Integration * Created and added content to table 3-1 in 3.1 functions * Added table labels to tables 1-1 and 3-1 |
| V1.5.1 | Muhammad Anas bin Khairul Azman | * Added Product Limitations in 1.3.4 * Added pretext for 3.1 Performance Requirements |
| V1.5.2 | Muhammad Anas bin Khairul Azman | * Added performance requirement and use case table for 3.1.5 |
| V1.5.3 | Muhammad Anas bin Khairul Azman | * Added functional requirement and use case table for 3.1.6, 3.1.7, 3.1.9 |
| V1.5.4 | Muhammad Anas bin Khairul Azman | * Added sequence diagrams for 3.1.5, 3.1.6, 3.1.7, 3.1.9 |
| V1.5.5 | Muhammad Anas bin Khairul Azman | * Fixed numbering for functional requirements |
| V.1.5.6 | Nicholas Thong Meng Shui | * Added Functional Requirements and use case table for 3.1.8, 3.1.10, 3.1.11 * Added Sequence Diagram for 3.1.8 |
| V1.5.7 | Muhammad Anas bin Khairul Azman | * Added cover page. * Added proper Table of Contents. * Added pre-text for upcoming chapters. * Added 3.2 Performance Requirements * Added 3.7 Security Requirements |
| V1.6 | Muhammad Anas bin Khairul Azman | * Added 3.4 Interface Requirements * Login interface and appointment interface |
| V1.6.1 | Nicholas Thong Meng Shui | * Updated figure 1.0 Use case diagram in 3.1 Functional Requirements * Updated table 3-1 Use case ID in 3.1 Functional Requirements * Updated use case table for 3.1.10, 3.1.11 * Added Sequence Diagrams for 3.1.10, 3.1.11 |
| V1.6.2 | Fikrul Amsyar Azmin | * Added functional requirements, use case table, and sequence diagram for 3.1.1, 3.1.2, 3.1.3, 3.1.4 * Added Project Purpose and Project Scope |
| V1.6.3 | Nicholas Thong Meng Shui | * Added functional requirements tables for 3.1.11 * Updated figure 1.0 Use case diagram in 3.1 Functional Requirements * Updated functional requirements and removed sequence diagram for 3.1.9 * Updated figure 1.0 Use case diagram in 3.1 Functional Requirements |
| V1.6.4 | Muhammad Anas bin Khairul Azman | * Updated the sequence diagrams for 3.1.6, 3.1.7 and 3.1.8. * Fixed numbering issues for use cases 6, 7, 8. |
| V.1.6.5 | Fikrul Amsyar Azmin | * Updated Project Scope * Update sequence diagrams for 3.1.1, 3.1.2, 3.1.3, 3.1.4 * Added Usability Requirements * Added Design Constraints * Added functional requirements, use case table, sequence diagram for 3.1.11 |
| V1.6.6 | Nicholas Thong Meng Shui | * Updated sequence diagrams for 3.1.8, 3.1.10, 3.1.12 * Updated 3.4.2, use case table and 3.5 |
| V1.6.7 | Muhammad Anas bin Khairul Azman | * Added 4.1 Verification Approach and 4.2 Verification Criteria * Completed 4.1 and 4.2 * Added labels and captions for every figure and table. * Added 3.8 Supporting Information. * Added 5.1 Assumptions and Dependencies and 5.2 Acronyms and Abbreviations |

# Introduction

## Purpose

This project involves developing a wellness platform that integrates with the university health centre’s appointment system and campus recreation facility management software. The system will enable students to manage their holistic wellness by scheduling health centre appointments, booking fitness classes, tracking personal wellness goals, and receiving tailored health resources.

## Scope

The system shall:

1. Schedule and cancel health centre appointments
2. Book for a fitness class
3. Tracks health goals
4. Receive tailored health resources
5. View personalized health tips
6. Manage health centre appointments and fitness class booking
7. Allow admins to manage user access control
8. Allow admins to view system logs

The system is set to operate only for university students.

## 1.3 Product Overview

### 1.3.1 Product Perspective

The Wellness Platform is a system that integrates the university health centre’s appointment system and campus recreation facility management software. Its primary role is to support student wellness through personalized health resources and wellness goals, integration with university medical appointment records, and interaction with recreational facilities.

As such, this system communicates and interoperates with several other components of the university's infrastructure. These include the university authentication system, recreation portal, and health portal. The system utilizes these interfaces to authenticate users, retrieve health data, enable health appointment scheduling, enable fitness class scheduling, and provide wellness recommendations.

#### System Interfaces

* **Authentication System**: The system relies on the university's login infrastructure to authenticate student users securely.
* **Recreation Portal**: Interfaces with the university’s recreation facility management software to enable fitness class booking and data sharing.
* **Health Portal**: Interfaces with the university’s health centre’s appointment system to allow students to schedule health appointments.

#### User Interfaces

1. Wellness Platform shall include intuitive designs customised by their roles:

|  |  |
| --- | --- |
| User Role | Interface Description |
| Student | Dashboard for setting, tracking and updating wellness goals, navigation bar to switch between managing fitness classes bookings screen, managing health centre appointments screen and a notifications panel. |
| System Admin | Buttons to access admin console and system logs. |

1. Wellness Platform shall make use of light platinum (RGB Hex code: #E1E2E4) colour for its overall background.
2. Wellness Platform shall always have black (RGB Hex code: #000000) font for non-navigation bar items unless specified otherwise.
3. Wellness Platform shall always have a blue (RGB Hex code: #000EBF)  
   navigation bar present at the top of the page.
4. Wellness Platform shall always have white (RGB Hex code: #FFFFFF) font for navigation bar items.
5. Wellness Platform shall always have white (RGB Hex code: #FFFFFF) for any frames within a page such as student notification frame, student wellness goal tracking frame, student manage fitness class booking and manage health centre appointment frame.
6. Wellness Platform shall always have blue (RGB Hex code: #000EBF)  
   for any buttons present on the platform unless specified otherwise.

#### Hardware Interfaces

The system supports standard student devices such as personal computers and mobile phones. No specialized hardware is required.

#### Software Interfaces

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Mnemonic** | **Specification Number** | **Version** | **Reference** |
| University Health Portal | UNIV\_HEALTH\_PORT | N/A | Current | [University Health Patient Portal](https://www.universityhealth.com/patient-visitor-resources/patients/patient-portal) |
| University Recreation Portal | UNIV\_REC\_PORTAL | N/A | Current | [Columbia University Recreation Portal](https://perec.columbia.edu/content/member-portal-resources) |
| OAuth 2.0 Authentication Framework | OAUTH2 | RFC 6749 | Latest | [OAuth 2.0 Specification](https://tools.ietf.org/html/rfc6749) |
| JSON Web Token Standard | JWT | RFC 7519 | Latest | [JWT Specification](https://tools.ietf.org/html/rfc7519) |
| RESTful API Interface | REST\_API | OpenAPI 3.0 | Latest | [OpenAPI Specification](https://swagger.io/specification/) |

Interfaces:

* University Health Portal
  + **Purpose**: To allow students to manage medical appointments and view personal health records.
  + **Interface Definition**: Web-based interface; provides scheduling, messaging, and medical record access. Integration through RESTful API.
  + **Reference**: University Health Patient Portal
* University Recreation Portal
  + **Purpose**: To enable scheduling and tracking of fitness classes and recreational activities.
  + **Interface Definition**: Booking and facility access information exchanged over REST API. Includes user authentication via SSO.
  + **Reference**: Columbia University Recreation Portal
* OAuth 2.0 Authentication Framework
  + **Purpose**: To authenticate students securely across university systems.
  + **Interface Definition**: Standard OAuth 2.0 protocol to handle login tokens and authorization. Interfaces with the university’s identity provider.
  + **Reference**: OAuth 2.0 Specification
* JSON Web Token (JWT)
  + **Purpose**: To transmit session data securely between frontend and backend systems.
  + **Interface Definition**: Encodes claim-based tokens using standard JWT format (RFC 7519), embedded in HTTP headers.
  + **Reference**: JWT Specification
* RESTful API Interface
  + **Purpose**: To facilitate integration between the wellness platform and external systems such as the university’s health portal and recreation systems.
  + **Interface Definition**: Uses HTTP(S) with standard methods (GET, POST, PUT, DELETE). Data formatted as JSON. Described using OpenAPI Specification.
  + **Reference**: OpenAPI Specification

#### Communication Interfaces

* HTTPS protocol is used for secure communication with internal university systems.
* Authentication via OAuth 2.0 and JWT for cross-system data sharing.

#### Memory Constraints

The system is lightweight and web-based, requiring minimal local memory. All user data is stored and processed in cloud/university-hosted servers.

#### Operations

The Wellness Platform is intended to function under both user-driven and automated operations. The following operational requirements are identified:

1) Modes of Operation

* **User-Initiated Mode**: Students interact with the web application to:
  + Set and view wellness goals.
  + Book or cancel health centre or fitness appointments.
  + View personalized health resources.
  + View system notifications.
* **Admin Mode**: Authorized staff access logs or assign admins roles to others through secure dashboards.

2) Periods of Interactive and Unattended Operations

* Interactive Operations:
  + Occur during active user sessions.
  + Includes booking appointments, viewing data, or modifying goals.
* Unattended Operations:
  + Background data syncing with Health Portal and Recreation Facility systems.
  + Periodic refresh of user statistics (e.g., wellness goals, appointment reminders).
  + Runs on a scheduled basis (e.g., nightly or hourly intervals).

3) Data Processing Support Functions

* **Data Aggregation**: Collects fitness and medical data from multiple systems and consolidates it in the student dashboard.
* **Data Validation**: Ensures data accuracy and compliance with privacy regulations (e.g., HIPAA).
* **Error Logging**: Logs failures during API communication or user actions for diagnostics.
* **Monitoring**: System health checks and logging for server uptime and performance metrics.

#### Site Adaptation Requirements

The system requires configuration of university-specific endpoints (e.g., Recreation API URLs, fitness class schedule formats) during deployment. It also requires institution-specific policy parameters (e.g., data retention limits, HIPAA compliance flags).

#### Interfaces with Services

Integrates with:

* University SaaS-based portals (e.g., health appointment scheduling)
* Cloud storage services (if enabled)

### 1.3.2 Product functions

Campus Recreation Facility Integration

* Search and filter fitness classes by their type, date and location.
* Book fitness classes.
* Notification for open slots in fitness classes.

University Health Portal Integration

* Track personal wellness goals.
* Set personal wellness goals (steps, workout, sleep).
* Update existing wellness goals.
* Receive tailored health resources based on user preferences.
* Schedule appointments with the health centre.
* Cancel scheduled appointments.

### 1.3.3 User Characteristics

Table 1-1 User Characteristics

|  |  |  |
| --- | --- | --- |
| **User group** | **Description** | **Minimum Ability/Knowledge** |
| Students | Students currently enrolled in the university | * Possess basic computer skills. |
| System Admin | Administrators assigned by the university for this platform | * Possess basic computer skills * Possess basic grasp of system commands * Possess knowledge on web development. |

### 1.3.4 Product Limitations

#### 1.3.4.1 General Limitations

The main limitations of our product comes from the fact that it is only accessible via web browser for ease of access for students. Furthermore, the use of this wellness portal is only accessible for students within the university and is not available for staff members (excluding system admins) of the university or students outside of the university.

Student functions such as managing fitness classes and health centre appointments also depend on integration with the University Recreation Portal and University Health Portal. As such, any disruptions from either systems will also disrupt the Wellness Platform’s functionality.

The Wellness Platform is a web-based platform thus an internet connection is required to access it. Weak internet connections may also lead to buffering or slow loading of website elements.

# 2.0 References

HIPAA Document - [HIPAA for Dummies - 2025 Update](https://www.hipaaguide.net/hipaa-for-dummies/)

[University Health Patient Portal](https://www.universityhealth.com/patient-visitor-resources/patients/patient-portal)

[Columbia University Recreation](https://perec.columbia.edu/content/member-portal-resources)

[OAuth 2.0 Specification](https://tools.ietf.org/html/rfc6749)

[JWT Specification](https://tools.ietf.org/html/rfc7519)

[OpenAPI Specification](https://swagger.io/specification/)

# 3.0 Requirements

## 3.1 Functional Requirement

### Context diagram

A diagram of a campus wellness portal

AI-generated content may be incorrect.

Figure 1: Context Diagram for Campus Wellness Portal

### Use Case Diagram

A diagram of a company

AI-generated content may be incorrect.

Figure 2: Use Case Diagram for Campus Wellness Portal

### Use Case ID table

Below are the use cases mapped to their relevant requirements from the Task 4/Kano Model.docx.

Table 1: Use Case ID Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | **Actor(s)** | **Use Case (Function)** | **Relevant Requirement**  **ID(s)** | **Brief Description** |
| **UC\_001** | Student | Login | REQID\_01 | Allows the student to log into the system using their credentials. |
| **UC\_002** | Student | Book Fitness Class | REQID\_04 | Enables the student to select and book a fitness class from the available options. |
| **UC\_003** | Student | Cancel Fitness Class | REQID\_05 | Allows the student to cancel a previously booked fitness class. |
| **UC\_004** | Student | Schedule Health Centre Appointment | REQID\_02 | Allows the student to choose a time slot and book an appointment with the health centre. |
| **UC\_005** | Student | Cancel Health Centre Appointment | REQID\_03 | Enables the student to cancel a scheduled health centre appointment. |
| **UC\_006** | Student | Track Wellness Goals | REQID\_06 | Allows the student to set, monitor, and update personal wellness goals. |
| **UC\_007** | Student | View System Notification | REQID\_10, REQID\_11 | Enables the student to view any and all notifications from the system. |
| **UC\_008** | Student | View Personalised Health Tips | REQID\_12 | Provides the student with curated health and wellness tips. |
| **UC\_009** | Student | Manage Health Centre Appointments | REQID\_08 | Receives and updates student health appointment bookings and cancellations. |
| **UC\_010** | Student | Manage Fitness Class Booking | REQID\_09 | Updates the fitness class schedule and availability based on bookings or cancellations. |
| **UC\_011** | Wellness System Admin | Manage User Access Control | - | Allows the admin to create, modify, and delete user accounts and assign or revoke roles/permissions. |
| **UC\_012** | Wellness System Admin | View System Logs | - | Enables the admin to retrieve, review and export system logs for monitoring and auditing purposes. |

### 3.1.1 Login

Functional Requirement for Login

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_101 |
| **Description** | System shall be able to validate user login. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_102 |
| **Description** | System shall be able to process user registration. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_103 |
| **Description** | System shall display an appropriate message when an error occurred where login credentials are incorrect. |
| **Author** | Fikrul Amsyar |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_001 | |
| **Feature** | F001 Login | |
| **Purpose** | To allow users to login to their own account | |
| **Actor** | Student | |
| **Trigger** | User clicks on “Login” button on the wellness platform portal | |
| **Precondition** | User is in the wellness platform portal | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | User clicks on “Login” button on the wellness platform portal. |
|  | 2 | System displays the login page to the user. |
|  | 3 | User inputs login credentials (studentID, studentPassword) and clicks the “Login” button. |
|  | 4 | System displays the wellness platform portal with the user account logged in if the login credentials is valid. |
| **Alternate Flow - Invalid Login Credentials** | 1 | System displays an appropriate error message to the user. |
|  | 2 | Back to Main Flow Step 2. |
| **Alternate Flow – Cancel Login** | 1 | User clicks on “Cancel Login” button on the login page. |
|  | 2 | System displays the wellness platform portal without the user account logged in. |
| **Rules** | * Login credentials must be within 25 characters. | |
| **Author** | Fikrul Amsyar Azmin | |

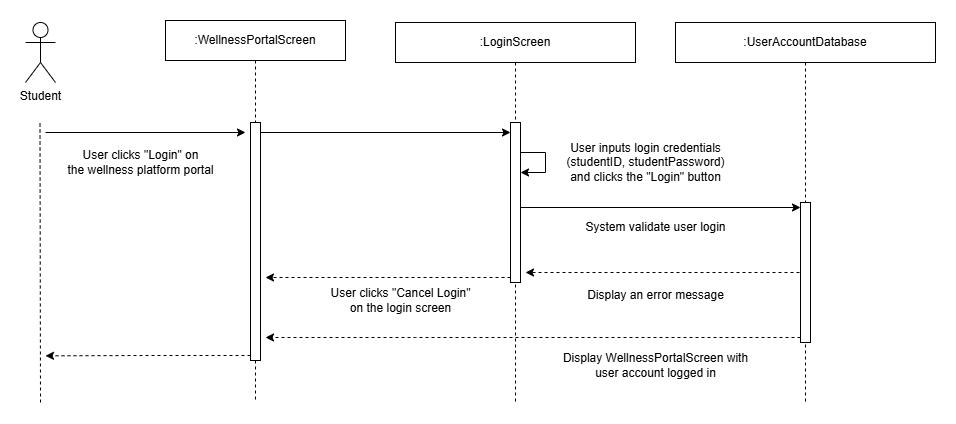


Figure 3: Sequence Diagram for Login

### 3.1.2 Book Fitness Class

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_201 |
| **Description** | System shall allow users to book a class if spots are available. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_202 |
| **Description** | System shall decrease the number of available spots after a successful booking. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_203 |
| **Description** | System shall display a confirmation message when a booking is successful. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_204 |
| **Description** | System shall display a table of available fitness classes. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_205 |
| **Description** | System shall prompt the user if they want to book the selected class. |
| **Author** | Fikrul Amsyar Azmin |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_002 | |
| **Feature** | F002 Book Fitness Class | |
| **Purpose** | To allow users to book for a fitness class. | |
| **Actor** | Student | |
| **Trigger** | User clicks on “Book Fitness Class” button on the wellness platform portal. | |
| **Precondition** | User account is logged in; User is in the wellness platform portal | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | User clicks “Book Fitness Class” button on the wellness platform portal. |
|  | 2 | System displays a table of available fitness classes with “Book” button on each row, and a “Go back” button at the top left near the table. |
|  | 3 | User clicks a “Book” button on the fitness classes table. |
|  | 4 | System displays a dialog box to the user if they want to book a class. |
|  | 5 | User clicks “Yes” button in the dialog box. |
|  | 6 | System displays a confirmation message that a booking is successful. |
|  | 7 | Back to Main Flow Step 2 |
| **Alternate Flow – User clicks “No” in the dialog box** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | Back to Main Flow Step 2 |
| **Alternate Flow – User clicks “Go back” in the table of available fitness classes** | 1 | User clicks “Go back” button in the table of available fitness classes page. |
|  | 2 | System displays the wellness platform portal. |
| **Rules** |  | |
| **Author** | Fikrul Amsyar Azmin | |

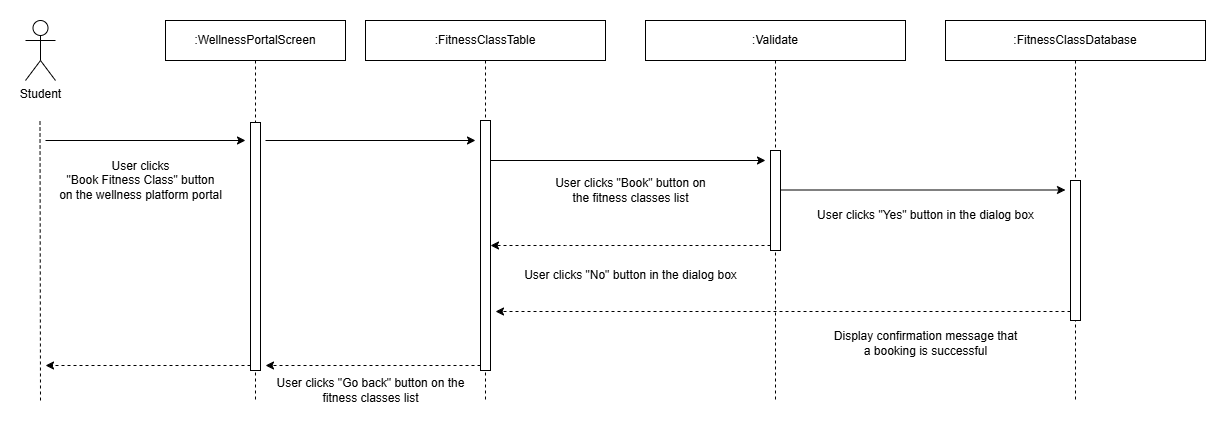


Figure 4: Sequence Diagram for Fitness Class Booking

### 3.1.3 Cancel Fitness Class

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_301 |
| **Description** | System shall allow users to cancel the selected class that the users booked. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_302 |
| **Description** | System shall increase the number of available spots after a cancellation. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_303 |
| **Description** | System shall display a confirmation message after a successful booked class cancellation. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_304 |
| **Description** | System shall prompt the user if they want to cancel the selected booked class. |
| **Author** | Fikrul Amsyar Azmin |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_003 | |
| **Feature** | F003 Cancel Fitness Class | |
| **Purpose** | To allow users to cancel a booked fitness class. | |
| **Actor** | Student | |
| **Trigger** | User clicks “Cancel” button on the manage fitness class screen. | |
| **Precondition** | User is in the manage fitness class booking screen | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | User clicks “Cancel” button on the fitness class table. |
|  | 2 | System displays a dialog box to the user if they want to cancel the selected class. |
|  | 3 | User clicks “Yes” button in the dialog box. |
|  | 4 | System displays a confirmation message that the selected booked class is cancelled. |
|  | 5 | System displays the manage fitness class booking screen. |
| **Alternate Flow – User clicks “No” in the dialog box** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | System displays the manage fitness class booking screen. |
| **Rules** |  | |
| **Author** | Fikrul Amsyar Azmin | |

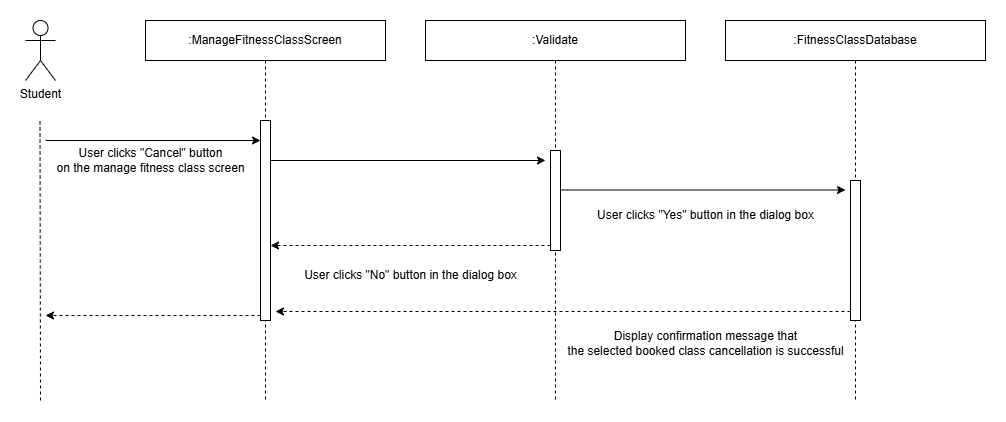


Figure 5: Sequence Diagram for Cancelling Fitness Class

### 3.1.4 Schedule Health Centre Appointment

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_401 |
| **Description** | System shall allow users to schedule health centre appointments. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_402 |
| **Description** | System shall display a table of available health centre appointments. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_403 |
| **Description** | System shall display a confirmation message after a successful schedule health centre appointment. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_404 |
| **Description** | System shall prompt the user if they want to make an appointment on the selected schedule. |
| **Author** | Fikrul Amsyar Azmin |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_004 | |
| **Feature** | F004 Schedule Health Centre Appointments | |
| **Purpose** | To allow users to schedule health centre appointments. | |
| **Actor** | Student | |
| **Trigger** | User clicks “Schedule Health Centre Appointments” button on the wellness platform portal. | |
| **Precondition** | User account is logged in; User is in the wellness platform portal | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | User clicks “Schedule Health Centre Appointments” button on the wellness platform portal. |
|  | 2 | System displays a table of available health centre appointments with “Schedule appointment” button on each row, and a “Go back” button at the top left near the table. |
|  | 3 | User clicks a “Schedule appointment” button on the table of available health centre appointments. |
|  | 4 | System displays a dialog box to the user if they want to make an appointment on the selected schedule. |
|  | 5 | User clicks “Yes” button in the dialog box. |
|  | 6 | System displays a confirmation message that the appointment schedule is successful. |
|  | 7 | Back to Main Flow Step 2. |
| **Alternate Flow – User clicks “No” in the dialog box** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | Back to Main Flow Step 2. |
| **Alternate Flow – User clicks “Go back” in the table of available health centre appointments** | 1 | User clicks “Go back” button in the table of available health centre appointments. |
|  | 2 | System displays the wellness platform portal. |
| **Rules** |  | |
| **Author** | Fikrul Amsyar Azmin | |

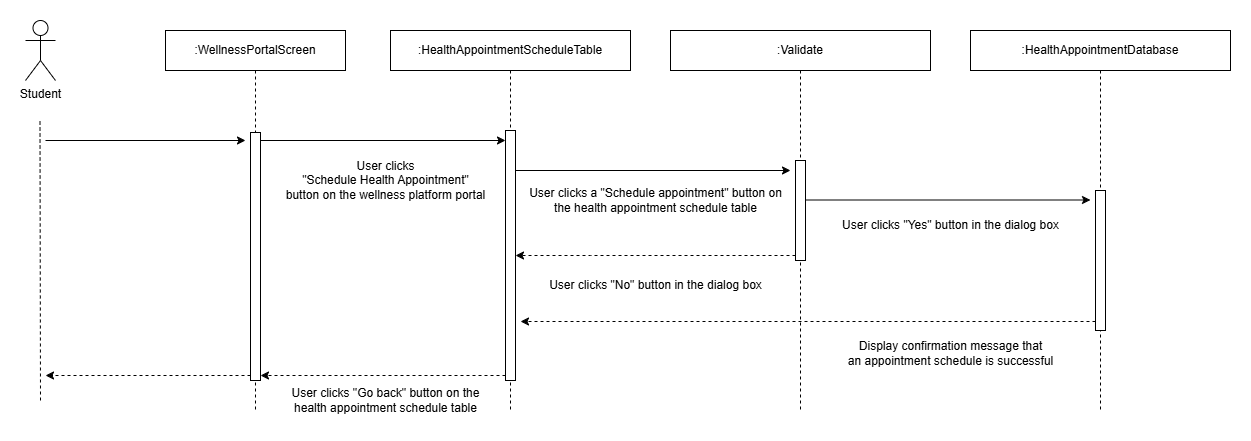


Figure 6: Sequence Diagram for Scheduling Health Centre Appointment

### 3.1.5 Cancel Health Centre Appointment

Functional Requirement for Cancel Health Centre Appointment

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_501 |
| **Description** | System shall allow the user to cancel a health centre appointment. |
| **Author** | Muhammad Anas |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_502 |
| **Description** | System shall prompt the user to confirm if they want to cancel the selected appointment. |
| **Author** | Muhammad Anas |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_503 |
| **Description** | System shall display a confirmation message that the selected appointment has been cancelled. |
| **Author** | Muhammad Anas |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_504 |
| **Description** | System shall update user’s appointments to reflect cancellation of appointment. |
| **Author** | Muhammad Anas |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_005 | |
| **Feature** | F005 Cancel Health Centre Appointments | |
| **Purpose** | To allow users to cancel their scheduled health centre appointments. | |
| **Actor** | Student | |
| **Trigger** | User clicks on “Cancel Appointment” button on a given appointment. | |
| **Precondition** | User is in the manage health centre appointment screen | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | User clicks on manage health centre appointment screen. |
|  | 2 | User clicks on “Cancel Appointment” for a given appointment. |
|  | 3 | System displays dialog box to confirm if the user wants to cancel said appointment. |
|  | 4 | User clicks on “Yes” in the dialog box. |
|  | 5 | System updates user’s appointment table to reflect the cancelled appointment. |
|  | 6 | System displays confirmation message that the appointment has been cancelled. |
|  | 8 | System displays the health centre appointment screen with the cancelled appointment now updated. |
| **Alternate Flow – User clicks “No” in the dialog box** | 1 | User clicks on “No” in the dialog box. |
|  | 2 | System displays the health centre appointment screen. |
| **Rules** | * User must already have an existing scheduled appointment. | |
| **Author** | Muhammad Anas | |

A diagram of a flowchart

AI-generated content may be incorrect.

Figure 7: Sequence Diagram for Cancelling Health Centre Appointment

### 3.1.6 Track Wellness Goals

Functional Requirement for Tracking Wellness Goals

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_601 |
| **Description** | System shall display the user’s wellness goals over the past month in the dashboard. |
| **Author** | Muhammad Anas |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_602 |
| **Description** | System shall allow the user to set new wellness goals. |
| **Author** | Muhammad Anas |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_603 |
| **Description** | System shall allow the user to update their wellness goals. |
| **Author** | Muhammad Anas |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_006 | |
| **Feature** | F006 Track Personal Wellness Goals | |
| **Purpose** | To allow users to monitor, set and update their personal wellness goals. | |
| **Actor** | Student | |
| **Trigger** | User clicks on “Wellness Goals” button on the platform portal. | |
| **Precondition** | User is in the wellness platform portal. | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow – Monitoring wellness goals** | 1 | User clicks on “Wellness Goals” button on the platform portal. |
|  | 2 | System displays wellness goals dashboard to the user. |
| **Alternate Flow – Setting new wellness goals** | 3 | User clicks on “Set new wellness goal” button. |
|  | 4 | System prompts the user to input their wellness goal. |
|  | 5 | System updates the user’s wellness goal database. |
|  | 6 | System displays confirmation message that the wellness goal has been added. |
|  | 7 | System displays the updated wellness goal dashboard. |
| **Alternate Flow – Updating existing wellness goals** | 1 | User clicks on “Update” button on a given wellness goal. |
|  | 2 | System prompts the user to update the values for the wellness goal. |
|  | 3 | System updates the user’s wellness goal database. |
|  | 4 | System displays confirmation message that the wellness goal has been updated. |
|  | 6 | System displays the updated wellness goal dashboard |
| **Rules** | - | |
| **Author** | Muhammad Anas | |

A diagram of a diagram

AI-generated content may be incorrect.

Figure 8: Sequence Diagram for Tracking Wellness Goals

### 3.1.7 View System Notifications

Functional Requirement for Viewing Notifications

|  |  |
| --- | --- |
| Requirement ID | REQ\_701 |
| Description | System shall allow the user to view any fitness class and/or health centre notifications. |
| Author | Muhammad Anas |

|  |  |
| --- | --- |
| Requirement ID | REQ\_702 |
| Description | System shall display the fitness class and/or health centre notifications to the user. |
| Author | Muhammad Anas |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_007 | |
| **Feature** | F007 View System Notifications | |
| **Purpose** | To allow users to view their fitness class and health centre appointment notifications. | |
| **Actor** | Student | |
| **Trigger** | User clicks on “Notifications” button on the platform portal. | |
| **Precondition** | User is in the wellness platform portal. | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | User clicks on “Notifications” button on the platform portal. |
|  | 2 | System displays the fitness class and/or health centre notifications to the user. |
| **Rules** | * Users must be signed up/have booked a fitness class. * Without it, notifications will be empty. * Or – user must have an appointment booked. * Without it, no notifications from the health centre will be sent. | |
| **Author** | Muhammad Anas | |

A diagram of a health care system

AI-generated content may be incorrect.

Figure 9: Sequence Diagram for Viewing System Notifications

### 3.1.8 View Personalized Health Tips

Functional Requirements for View Personalized Health Tips

|  |  |
| --- | --- |
| Requirement ID | REQ\_801 |
| Description | |  | | --- | | System shall retrieve health tips tailored to the student’s profile and goals. |  |  | | --- | |  | |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_802 |
| Description | System shall display retrieved health tips in the student’s dashboard. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_803 |
| Description | System shall allow the student to request more details on any displayed tip. |
| Author | Nicholas Thong Meng Shui |

Use Case Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC\_008 | | |
| **Feature** | |  | | --- | |  |  |  | | --- | | F008 View Personalized Health Tips | | | |
| **Purpose** | Provide curated health and wellness tips based on student profile. | | |
| **Actor** | Student | | |
| **Trigger** | Student selects “View Health Tips” on Student Dashboard. | | |
| **Precondition** | Student is logged in and on Student Dashboard page; tips database is accessible. | | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow – User viewing health tips** | 1 | |  |  | | --- | --- | |  | Student clicks “Health Tips” menu. | |
| 2 | |  | | --- | | System retrieves tailored tips (REQ\_801). | |
| 3 | |  |  | | --- | --- | |  | System displays tips list (REQ\_802). | |
| 4 | |  | | --- | | Student selects a tip for more details. | |
| 5 | System shows full tip details (REQ\_803). |
| **Alternate Flow – No Tips Available** | 2a | If no tips match profile, System displays “No personalized tips available.” |
| **Rules** | Tips must be relevant to current wellness goals. | | |
| **Author** | Nicholas Thong Meng Shui | |

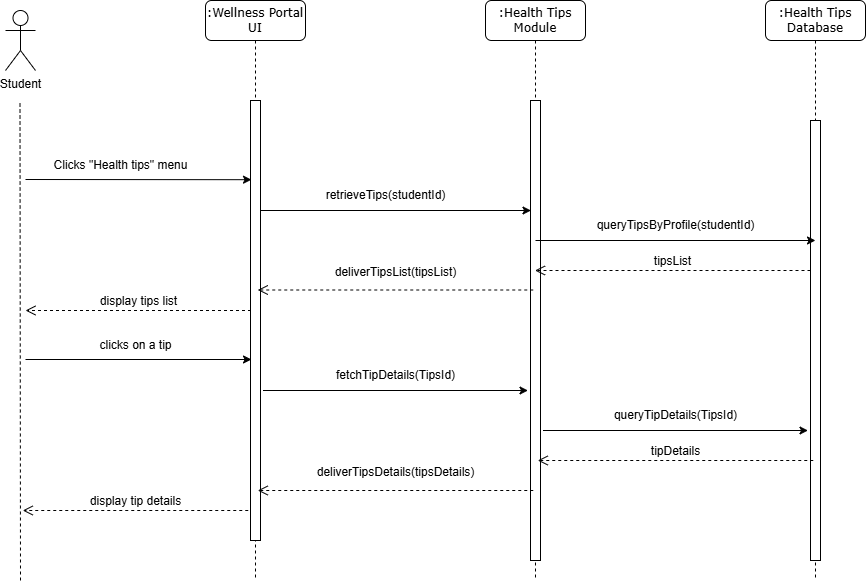


Figure 10: Sequence Diagram for Viewing Personal Health Tips

### 3.1.9 Manage Health Centre Appointments

Functional Requirement for Manage Health Centre Appointments

|  |  |
| --- | --- |
| Requirement ID | REQ\_901 |
| Description | System shall allow the user the view their health centre appointments (scheduled and cancelled). |
| Author | Muhammad Anas |

|  |  |
| --- | --- |
| Requirement ID | REQ\_902 |
| Description | System shall allow the user to schedule new health centre appointments. |
| Author | Muhammad Anas |

|  |  |
| --- | --- |
| Requirement ID | REQ\_903 |
| Description | System shall allow the user to cancel their booked health centre appointments. |
| Author | Muhammad Anas |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC\_009 | |
| **Feature** | F009 Manage Health Centre Appointments | |
| **Purpose** | To receive and update a user’s scheduled and cancelled health centre appointments. | |
| **Actor** | University Health Portal | |
| **Trigger** | User clicks on “Schedule Appointment” or “Cancel Appointment” button on the “Health Centre Appointments” screen. | |
| **Precondition** | User is in the Health Centre Appointments screen. | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow – User scheduled an appointment** | 1 | User clicks on “Health Centre Appointments” button on the Wellness dashboard. (REQ\_901) |
|  | 2 | User clicks on “Schedule Appointment” button – invoke UC\_004 |
| **Alternate Flow – User cancelled an appointment** | 1 | User clicks on “Health Centre Appointments” button on the Wellness dashboard. |
|  | 2 | User clicks on “Cancel Appointment” button on desired appointment – invoke UC\_005 |
| **Rules** | - | |
| **Author** | Muhammad Anas | |

A diagram of a process

AI-generated content may be incorrect.

Figure 11: Sequence Diagram for Scheduling Health Centre Appointment

### 3.1.10 Manage Fitness Class Booking

Functional Requirements for Manage Fitness Class Booking

|  |  |
| --- | --- |
| Requirement ID | REQ\_1001 |
| Description | System shall display a list of the student’s current class bookings. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1002 |
| Description | System shall allow branching to either the Book or Cancel use case based on action. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1003 |
| Description | Upon completion of the child use case, system shall refresh the bookings list. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1004 |
| Description | If the Book or Cancel use case produces an error, system shall display it. |
| Author | Nicholas Thong Meng Shui |

Use Case Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC\_010 | | |
| **Feature** | |  | | --- | |  |  |  | | --- | | F010 Manage Fitness Class Booking | | | |
| **Purpose** | Allows students to view, book and cancel fitness-class reservations from one page. | | |
| **Actor** | |  | | --- | |  |  |  | | --- | | Student | | | |
| **Trigger** | |  | | --- | |  |  |  | | --- | | Student navigates to the “Fitness Class Bookings” screen. | | | |
| **Precondition** | Student is logged in; “Fitness Class Bookings” screen is displayed. | | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow – Student view, book or cancel fitness class** | 1 | |  |  | | --- | --- | |  | System displays a table of the student’s current class reservations. | |
| 2 | Student views scheduled fitness classes |
| 3 | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | |  | | --- | |  |  |  |  | | --- | --- | | |  | | --- | | Student clicks either “Book Class” or “Cancel Class.” | | | |
| 3.1 | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | If “Book Class”: invoke **UC002 Book Fitness Class** | | |
| 3.2 | If “Cancel Class”: invoke **UC003 Cancel Fitness Class** |
| 4 | System refreshes and re-displays the updated reservations table. |
| **Alternate Flow – Table error** | 1a | Recreation Portal API returns error during table request. |
| 1b | System logs and displays error |
| **Rules** | Booking requests must include valid student and class identifiers. | | |
| **Author** | Nicholas Thong Meng Shui | |

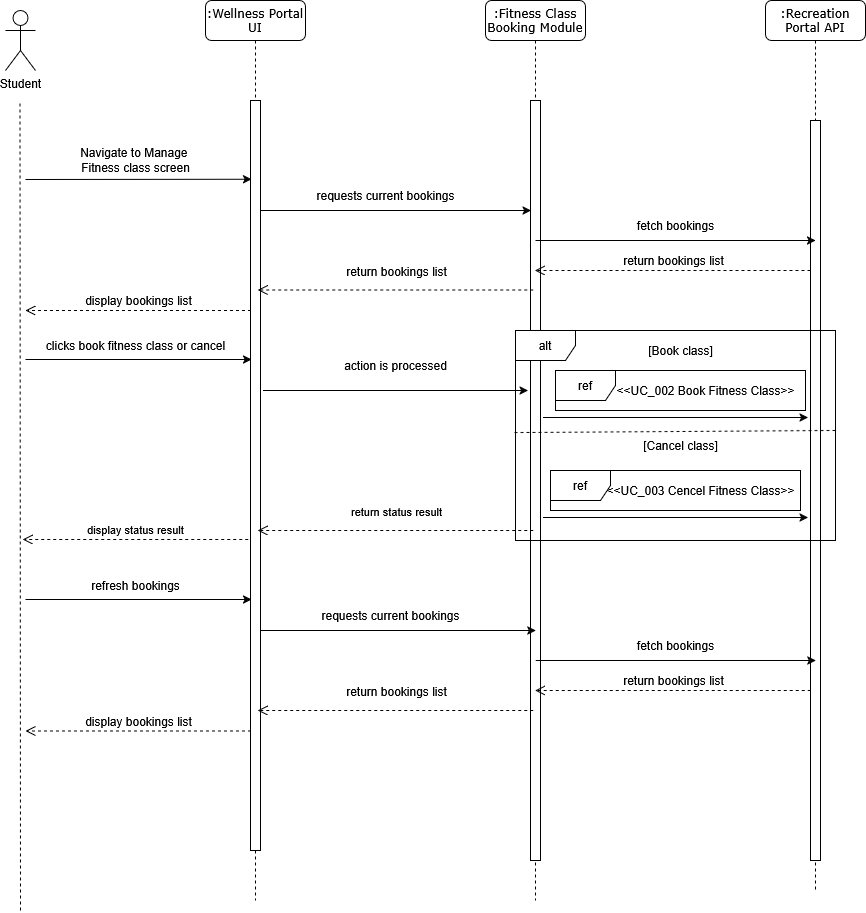


Figure 12: Sequence Diagram for Managing Fitness Classes

### 3.1.11 Manage User Access Control

Functional Requirement for Manage User Access Control

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_1101 |
| **Description** | System shall allow an admin to create a new user account. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_1102 |
| **Description** | System shall allow an admin to view and edit existing user details. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_1103 |
| **Description** | System shall allow an admin to delete or deactivate a user account. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_1104 |
| **Description** | System shall allow an admin to assign one or more roles to a user. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_1105 |
| **Description** | System shall allow an admin to assign or revoke specific permissions. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_1106 |
| **Description** | System shall display a table of all users, including their roles. |
| **Author** | Fikrul Amsyar Azmin |

|  |  |
| --- | --- |
| **Requirement ID** | REQ\_1107 |
| **Description** | System shall prompt for confirmation to the admin before making changes. |
| **Author** | Fikrul Amsyar Azmin |

Use Case Table

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC011 | |
| **Feature** | F011 Manage User Access Control | |
| **Purpose** | To allow the admin to create, modify, and delete user accounts and assign or revoke roles/permissions. | |
| **Actor** | Wellness System Admin | |
| **Trigger** | User clicks on “Manage User Accounts” button on the wellness system admin screen. | |
| **Precondition** | User account is logged in; User is in the wellness system admin screen. | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | User clicks “Manage User Accounts” button on the wellness system admin screen. |
|  | 2 | System displays a table of user accounts with “Edit” and “Delete” button on each row. A “Create Account” button at the bottom of the table. A “Go back” button at top left corner of the screen. |
|  | 3 | User clicks “Go back” button on the manage user accounts screen. |
|  | 4 | System displays wellness system admin screen. |
| **Alternate Flow – Edit Account** | 1 | User clicks an “Edit” button in the table . |
|  | 2 | System displays user account edit screen with a “Save” and “Cancel” button at the bottom of the screen. |
|  | 3 | User edits the user account details, roles, and permissions, and clicks “Save” button. |
|  | 4 | System displays a dialog box to the user if they want to save the changes. |
|  | 5 | User clicks “Yes” button in the dialog box. |
|  | 6 | System displays a confirmation message that the user account data is updated. |
|  | 7 | Back to Main Flow Step 2. |
| **Alternate Flow – User clicks “No” in the save changes dialog box** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | Back to Alternate Flow – Edit Account Step 2. |
| **Alternate Flow – User clicks “Cancel” button in the user account edit screen** | 1 | System displays a dialog box to the user if they want to go back to previous screen without savng. |
|  | 2 | User clicks “Yes” button in the dialog box. |
|  | 3 | Back to Main Flow Step 2 without the changes saved. |
| **Alternate Flow – User clicks “No” button in the go back without saving dialog box** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | Back to Alternate Flow – Edit Account Step 2. |
| **Alternate Flow – Delete User Account** | 1 | User clicks a “Delete” button in the user accounts table. |
|  | 2 | System displays a dialog box to the user if they want to delete the selected account. |
|  | 3 | User clicks “Yes” button in the dialog box. |
|  | 5 | System displays a confirmation message that the selected user account is deleted. |
|  | 6 | Back to Main Flow Step 2. |
| **Alternate Flow – User clicks “No” button in the account deletion dialog box.** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | Back to Main Flow Step 2 without the selected user account removed from the table. |
| **Alternate Flow – Create User Account** | 1 | User clicks “Create Account” button on the manage user account screen. |
|  | 2 | System displays create user account screen with “Create” and “Cancel” button at the bottom of the screen. |
|  | 3 | User inputs the user account details, roles, and permissions, and clicks “Create” button. |
|  | 4 | System displays a dialog box to the user if they want to create the account. |
|  | 5 | User clicks “Yes” button in the dialog box. |
|  | 6 | System displays a confirmation message that a user account has been created. |
|  | 7 | Back to Main Flow Step 2. |
| **Alternate Flow – User clicks “No” button in the account creation dialog box.** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | Back to Alternate Flow – Create User Account Step 2. |
| **Alternate Flow – User clicks “Cancel” button in the account creation screen.** | 1 | User clicks “Cancel” button in the account creation screen. |
|  | 2 | System displays a dialog box to the user if they want to cancel the account creation. |
|  | 3 | User clicks “Yes” button in the dialog box. |
|  | 4 | Back to Main Flow Step 2 without the user account added in the table. |
| **Alternate Flow – User clicks “No” button in the cancel account creation screen.** | 1 | User clicks “No” button in the dialog box. |
|  | 2 | Back to Alternate Flow – Create User Account Step 2. |
| **Rules** |  | |
| **Author** | Fikrul Amsyar Azmin | |

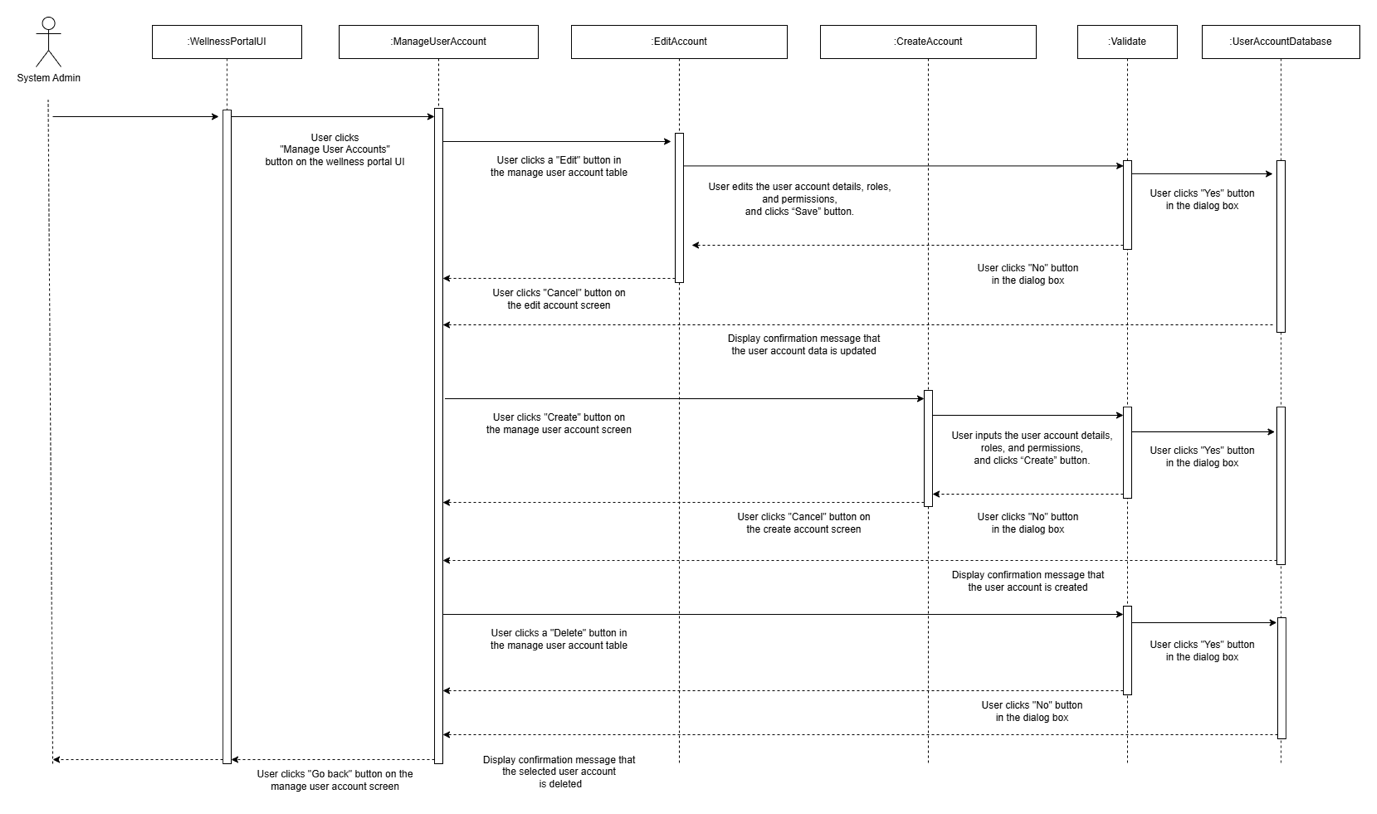


Figure 13: Sequence Diagram for Managing Use Access

### 3.1.12 View system logs

Functional requirements for view system logs

|  |  |
| --- | --- |
| Requirement ID | REQ\_1201 |
| Description | System shall allow administrators to filter logs by date range, severity level, and source module. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1202 |
| Description | System shall display a paginated list of log entries matching the selected filter criteria. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1203 |
| Description | System shall show full details (timestamp, actor, action, message, stack trace) for any log entry. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1204 |
| Description | System shall allow administrators to export filtered log entries in CSV or JSON format. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1205 |
| Description | System shall restrict access to the View Logs feature to users with the “Wellness System Admin” role. |
| Author | Nicholas Thong Meng Shui |

|  |  |
| --- | --- |
| Requirement ID | REQ\_1206 |
| Description | System shall log each “view” or “export” action performed by administrators for audit purposes. |
| Author | Nicholas Thong Meng Shui |

Use Case Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC\_012 | | |
| **Feature** | F012 View System Logs | | |
| **Purpose** | |  | | --- | |  |  |  | | --- | | Provide administrators with on-demand access to system logs for monitoring and troubleshooting. | | | |
| **Actor** | |  | | --- | |  |  |  | | --- | | Wellness System Admin | | | |
| **Trigger** | |  | | --- | |  |  |  | | --- | | Admin selects “View System Logs” in the admin page. | | | |
| **Precondition** | |  | | --- | |  |  |  | | --- | | Admin is authenticated and has “Wellness System Admin” role.; Log database is reachable. | | | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow – User scheduled an appointment** | 1 | |  |  | | --- | --- | |  | Admin navigates to the System Logs page. | |
| 2 | |  | | --- | | System displays filter controls (date range, severity, source). (*REQ-1201)* | |
| 3 | |  |  |  |  | | --- | --- | --- | --- | |  | |  | | --- | |  |  |  | | --- | | Admin enters filter criteria and clicks “Search.” | | |
| 4 | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | System retrieves matching log entries, paginates them, and displays a summary list. (*REQ-1202)* | | |
| 5 | Admin clicks on a specific log entry. |
| 6 | System shows full details of that entry (timestamp, actor, action, message, stack trace). (*REQ-1203)* |
| **Alternate Flow – No Matching Logs** | 4a | If no log entries match the filter, System displays “No records found” and offers to clear filters. |
| **Alternate Flow – Export Logs** | 7a | Integration Service receives data with unexpected format from one portal. |
| 7b | System generates the export file and makes it available for download. (*REQ-1204)* |
| 7c | System records the “view” and/or “export” action in an audit log. (*REQ-1206)* |
| **Rules** | - Only one export operation can run at a time per admin session.  - Log entries must be retained for at least 90 days. | | |
| **Author** | Nicholas Thong Meng Shui | |

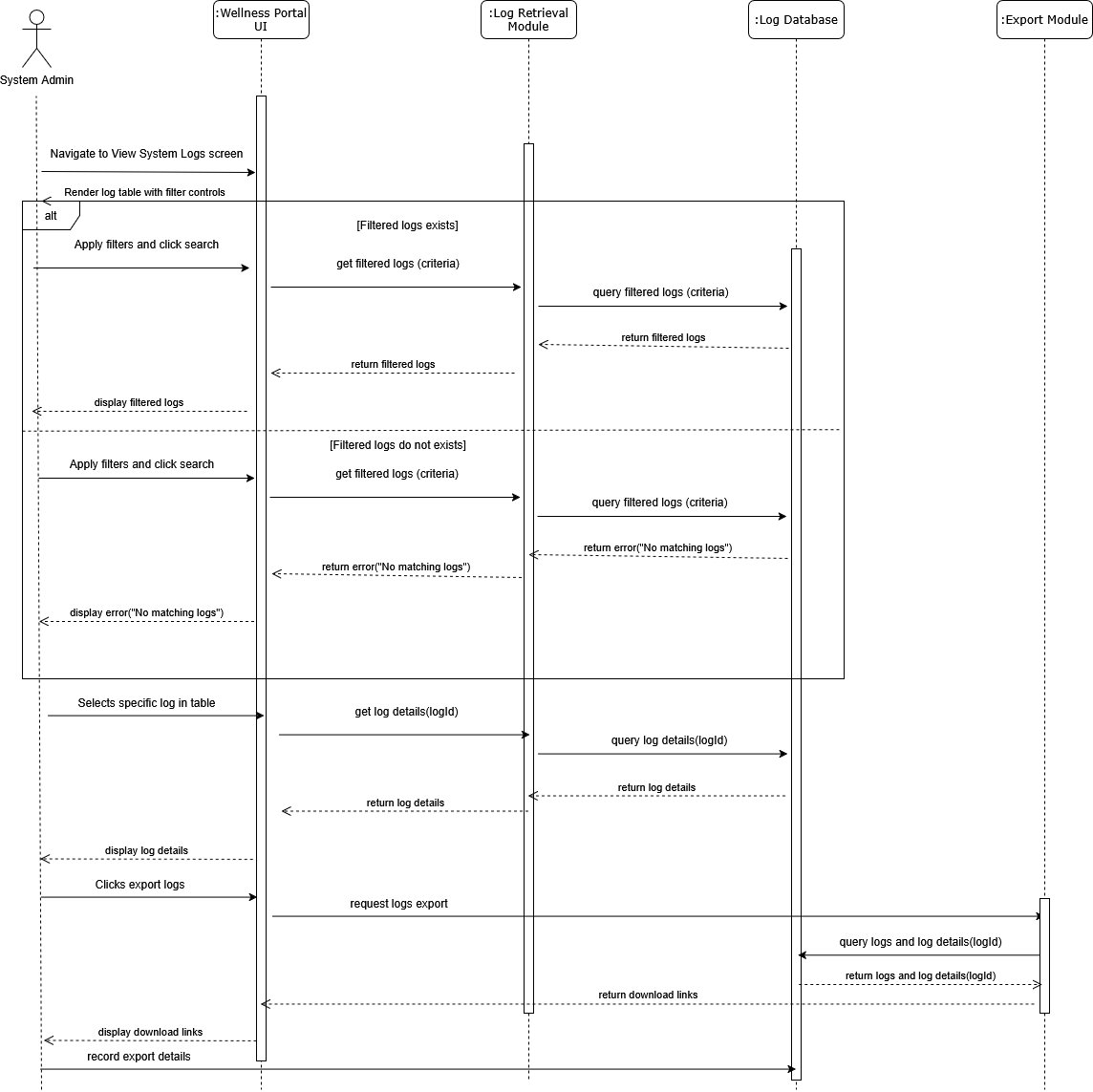
Sequence diagram for viewing system logs:

Figure 14: Sequence Diagram for Viewing System Logs

## 3.2 Performance Requirements

The performance for the Campus Wellness Portal are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_P001 | The average response time of Wellness Portal functions shall be less than 2 seconds. | High | Muhammad Anas |

## 3.3 Usability Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_UR001 | System shall be intuitive enough for first-time users to perform basic tasks without extensive training. | High | Fikrul Amsyar Azmin |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_UR002 | Consistent design patterns and navigation shall be used across pages. | High | Fikrul Amsyar Azmin |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_UR003 | System shall prevent user errors through input validation, tooltips, and disabling invalid options. | High | Fikrul Amsyar Azmin |

## 3.4 Interface Requirements

This section covers the external interface requirements of the Campus Wellness Portal, separated by the features of the system.

### 3.4.1 System Interfaces

#### 3.4.1.1 University Health Portal Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_SI101 | **Version** | 1.0 |
| **Item** | Health Portal Interface | | |
| **Description** | The campus wellness portal interacts with the university’s health portal to manage appointments with the university health centre. | | |
| **Purpose** | Allows the management of user’s health centre appointment from the wellness portal. | | |
| **Author** | Muhammad Anas | | |

#### 3.4.1.2 University Recreation Portal Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_SI102 | **Version** | 1.0 |
| **Item** | Recreation Portal Interface | | |
| **Description** | The campus wellness portal interacts with the university’s recreation portal to manage fitness class bookings. | | |
| **Purpose** | Allows the management of user’s fitness class bookings from the wellness portal. | | |
| **Author** | Muhammad Anas | | |

### 3.4.2 User Interface

#### 3.4.2.1 Login Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI101 | **Version** | 1.0 |
| **Item** | Student ID (input) | | |
| **Description** | Input student ID on the portal login | | |
| **Purpose** | To allow the student to log into the portal | | |
| **Format** | String/Text | **Valid Range** | Not applicable |
| **Related I/O** | None | | |
| **Author** | Muhammad Anas | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI102 | **Version** | 1.0 |
| **Item** | Student password (input) | | |
| **Description** | Input student password on the portal login | | |
| **Purpose** | To allow the student to log into the portal | | |
| **Format** | String/Text | **Valid Range** | Not applicable |
| **Related I/O** | None | | |
| **Author** | Muhammad Anas | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI103 | **Version** | 1.0 |
| **Item** | ‘Login Button’ button (input) | | |
| **Description** | Sends Student ID and Student password information to the authentication system | | |
| **Purpose** | To allow the student to log into the portal | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | None | | |
| **Author** | Nicholas Thong Meng Shui | | |

#### 3.4.2.2 Navigation Bar Interface

The navigation bar will provide easy access to students and admins to navigate to certain pages.

Navigation bar items:

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI201 | **Version** | 1.0 |
| **Item** | ‘Dashboard’ button (input) | | |
| **Description** | Button to navigate to the Student Dashboard screen | | |
| **Purpose** | To allow the student to see and interact with their dashboard | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | None | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI202 | **Version** | 1.0 |
| **Item** | ‘Fitness Class Bookings’ button (input) | | |
| **Description** | Button to navigate to the Manage Fitness Class Bookings screen | | |
| **Purpose** | To allow the student to manage their fitness classes | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: “Fitness Class Bookings” screen | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI203 | **Version** | 1.0 |
| **Item** | ‘Health Centre Appointments’ button (input) | | |
| **Description** | Button to navigate to the Manage Health Centre Appointments screen | | |
| **Purpose** | To allow the student to manage their health centre appointments | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Manage Health Care Appointments table | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI204 | **Version** | 1.0 |
| **Item** | ‘Notifications’ button (Input) | | |
| **Description** | Button to view notifications from the system | | |
| **Purpose** | To allow the student to view all notifications from the system | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Notifications Panel | | |
| **Author** | Nicholas Thong Meng Shui | | |

#### 3.4.2.3 Student Dashboard Interface

The Student Dashboard interface will use a responsive layout with a fixed top navigation bar.

Student Dashboard Items:

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI301 | **Version** | 1.0 |
| **Item** | “Wellness Goals” button (Input) | | |
| **Description** | |  | | --- | | Button to open the Wellness Goals dashboard | | | |
| **Purpose** | To allow the student to view their wellness-goals page | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Wellness Goals dashboard | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI302 | **Version** | 1.0 |
| **Item** | “Set new wellness goal” button (Input) | | |
| **Description** | Button to initiate creation of a new goal | | |
| **Purpose** | To allow the student to add a new wellness goal | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Input wellness goal prompt | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI303 | **Version** | 1.0 |
| **Item** | “Update” button (Input) | | |
| **Description** | Button to edit an existing wellness goal | | |
| **Purpose** | To allow the student to modify an existing goal’s values | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Update-goal prompt dialog (Input) | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI304 | **Version** | 1.0 |
| **Item** | Wellness Goals dashboard (Output) | | |
| **Description** | Screen showing the student’s current wellness goals | | |
| **Purpose** | To present all tracked goals and their status | | |
| **Format** | Screen/UI | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Wellness Goals” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI305 | **Version** | 1.0 |
| **Item** | Wellness goal prompt dialog (Input) | | |
| **Description** | Dialog prompting student to enter a new wellness goal | | |
| **Purpose** | To collect details for a new goal | | |
| **Format** | Modal dialog | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Set new wellness goal” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI306 | **Version** | 1.0 |
| **Item** | Update-goal prompt dialog (Input) | | |
| **Description** | Dialog prompting student to modify existing goal values | | |
| **Purpose** | To collect updated information for an existing goal | | |
| **Format** | Modal dialog | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Update” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI307 | **Version** | 1.0 |
| **Item** | Updated Wellness Goals dashboard | | |
| **Description** | Screen showing the refreshed list of wellness goals | | |
| **Purpose** | To inform the student that their goal update has been saved | | |
| **Format** | Toast/banner | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Update” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI308 | **Version** | 1.0 |
| **Item** | “View Health Tips” Button (Input) | | |
| **Description** | Button on the Student Dashboard to access personalized tips | | |
| **Purpose** | To allow the student to retrieve tailored health and wellness tips | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Health Tips list | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI309 | **Version** | 1.0 |
| **Item** | Health Tips list (Output) | | |
| **Description** | Scrollable list displaying brief summaries of curated tips | | |
| **Purpose** | To present all personalized tips matching the student’s profile | | |
| **Format** | List with selectable items | **Valid Range** | Not applicable |
| **Related I/O** | Input: “View Health Tips” menu item | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI310 | **Version** | 1.0 |
| **Item** | Tip Details panel (Output) | | |
| **Description** | Panel or modal showing full details of a selected tip | | |
| **Purpose** | To allow the student to read and understand a specific health tip in-depth | | |
| **Format** | Panel / Modal dialog | **Valid Range** | Not applicable |
| **Related I/O** | Input: Health Tips list (when a tip is selected) | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI311 | **Version** | 1.0 |
| **Item** | “No Tips Available” message (Output)) | | |
| **Description** | Inline message stating “No personalized tips available.” | | |
| **Purpose** | To inform the student when no tips match their profile | | |
| **Format** | Inline text / Banner | **Valid Range** | Not applicable |
| **Related I/O** | Input: “View Health Tips” button (when list is empty) | | |
| **Author** | Nicholas Thong Meng Shui | | |

#### 3.4.2.4 Notification Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI401 | **Version** | 1.0 |
| **Item** | Notifications Panel (Output) | | |
| **Description** | Panel displaying system notifications, including fitness-class and health-centre updates | | |
| **Purpose** | To present the student with all current notifications in a consolidated view | | |
| **Format** | Panel / Sidebar UI | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Notifications” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

#### 3.4.2.5 Managing Health Care Appointments Interface

The Managing Health Care Appointments interface will use a responsive layout with a fixed top navigation bar.

Managing Health Care Appointments items:

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI501 | **Version** | 1.0 |
| **Item** | Manage Health Care Appointments table (Output) | | |
| **Description** | Table listing current appointments with both “Schedule appointment” and “Cancel Appointment” actions available | | |
| **Purpose** | To present existing bookings and allow scheduling/cancellation without leaving the page | | |
| **Format** | Table with action buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Health Centre Appointments” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI502 | **Version** | 1.0 |
| **Item** | “Schedule Health Centre Appointments” button (Input) | | |
| **Description** | Button to open the available-appointments view | | |
| **Purpose** | To allow the student to view and choose from upcoming health centre slots | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Available Appointments table | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI503 | **Version** | 1.0 |
| **Item** | Available Appointments table (Output) | | |
| **Description** | Tabular list of upcoming health centre timeslots, each with a “Schedule appointment” button | | |
| **Purpose** | To display all slots the student can book | | |
| **Format** | Table with action buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Schedule Health Centre Appointments” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI504 | **Version** | 1.0 |
| **Item** | “Schedule appointment” button (Input) | | |
| **Description** | Button on each row to initiate booking of that timeslot | | |
| **Purpose** | To let the student confirm which appointment to schedule | | |
| **Format** | Button (per row) | **Valid Range** | Not applicable |
| **Related I/O** | Output: Schedule Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI505 | **Version** | 1.0 |
| **Item** | Schedule Confirmation dialog (Output) | | |
| **Description** | Modal asking “Do you want to confirm this appointment?” | | |
| **Purpose** | To get explicit user confirmation before booking | | |
| **Format** | Modal dialog with “Yes”/“No” buttons) | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Schedule appointment” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI506 | **Version** | 1.0 |
| **Item** | Booking Confirmation message (Output) | | |
| **Description** | Inline toast/banner saying “Appointment scheduled successfully” | | |
| **Purpose** | To inform the student the booking went through | | |
| **Format** | Toast/banner notification | **Valid Range** | Not applicable |
| **Related I/O** | Input: Schedule Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI507 | **Version** | 1.0 |
| **Item** | “Cancel Appointment” button (Input) | | |
| **Description** | Button displayed next to each appointment on the Manage screen | | |
| **Purpose** | To allow the student to initiate cancellation of a booked slot | | |
| **Format** | Button (per row) | **Valid Range** | Not applicable |
| **Related I/O** | Output: Cancel Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI508 | **Version** | 1.0 |
| **Item** | Cancel Confirmation dialog (Output) | | |
| **Description** | Modal asking “Are you sure you want to cancel this appointment?” | | |
| **Purpose** | To verify the student’s intent before removing the booking | | |
| **Format** | Modal dialog with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Cancel Appointment” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI509 | **Version** | 1.0 |
| **Item** | Cancellation Confirmation message (Output) | | |
| **Description** | Inline toast/banner saying “Appointment cancelled successfully” | | |
| **Purpose** | To inform the student the cancellation was processed | | |
| **Format** | Toast/banner notification | **Valid Range** | Not applicable |
| **Related I/O** | Input: Cancel Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

#### 3.4.2.6 Managing Fitness Class Bookings interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI601 | **Version** | 1.0 |
| **Item** | “Fitness Class Bookings” screen (Output) | | |
| **Description** | Screen showing current class reservations in a table with Book/Cancel actions | | |
| **Purpose** | To provide a unified view for viewing, booking, and cancelling classes | | |
| **Format** | Screen/UI with action buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: ‘Fitness Class Bookings’ button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI602 | **Version** | 1.0 |
| **Item** | Refresh Bookings action (Input) | | |
| **Description** | Implicit or visible control to reload the bookings list after an operation | | |
| **Purpose** | To update the display with the latest reservation state | | |
| **Format** | Automatic Refresh | **Valid Range** | Not applicable |
| **Related I/O** | Output: Updated Bookings table | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI603 | **Version** | 1.0 |
| **Item** | Error display area (Output) | | |
| **Description** | Inline message area for showing API errors when loading or refreshing | | |
| **Purpose** | To inform the student when bookings cannot be retrieved or updated | | |
| **Format** | Inline text / Banner | **Valid Range** | Not applicable |
| **Related I/O** | Input: system error responses | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI604 | **Version** | 1.0 |
| **Item** | “Book Fitness Class” button (Input) | | |
| **Description** | Button on the wellness portal to initiate booking flow | | |
| **Purpose** | To allow the student to view available classes and start a booking | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Available Classes table | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI605 | **Version** | 1.0 |
| **Item** | Available Classes table (Output) | | |
| **Description** | Table listing classes with a “Book” button on each row and a “Go back” button | | |
| **Purpose** | To display all classes the student can book | | |
| **Format** | Table with action buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Book Fitness Class” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI606 | **Version** | 1.0 |
| **Item** | “Book” button (Input) | | |
| **Description** | Button on each class row to select that class for booking | | |
| **Purpose** | To let the student choose which class to reserve | | |
| **Format** | Button (per row) | **Valid Range** | Not applicable |
| **Related I/O** | Output: Booking Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI607 | **Version** | 1.0 |
| **Item** | Booking Confirmation dialog (Output) | | |
| **Description** | Modal asking “Do you want to book this class?” | | |
| **Purpose** | To confirm the student’s intent before placing the booking | | |
| **Format** | Modal dialog with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Book” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI608 | **Version** | 1.0 |
| **Item** | Booking Success message (Output) | | |
| **Description** | Inline toast/banner reading “Booking successful” | | |
| **Purpose** | To inform the student that their class has been reserved | | |
| **Format** | Toast/banner notification | **Valid Range** | Not applicable |
| **Related I/O** | Input: Booking Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI609 | **Version** | 1.0 |
| **Item** | “Cancel” button (Input) | | |
| **Description** | Button next to each booked class on the Manage screen | | |
| **Purpose** | To allow the student to start a cancellation | | |
| **Format** | Button (per row) | **Valid Range** | Not applicable |
| **Related I/O** | Output: Cancellation Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI610 | **Version** | 1.0 |
| **Item** | Cancellation Confirmation dialog (Output) | | |
| **Description** | Modal asking “Are you sure you want to cancel this booking?” | | |
| **Purpose** | To verify the student’s intent before removing the reservation | | |
| **Format** | Modal dialog with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Cancel” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI611 | **Version** | 1.0 |
| **Item** | Cancellation Success message (Output) | | |
| **Description** | Inline toast/banner reading “Booking cancelled successfully” | | |
| **Purpose** | To inform the student that the cancellation was processed | | |
| **Format** | Toast/banner notification | **Valid Range** | Not applicable |
| **Related I/O** | Input: Cancellation Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

#### 3.4.2.7 System Admin Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI701 | **Version** | 1.0 |
| **Item** | “Manage User Accounts” button (Input) | | |
| **Description** | Button on the admin console to open the user‐accounts management screen | | |
| **Purpose** | To allow the admin to view, edit, delete, or create user accounts | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: User Accounts table | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI702 | **Version** | 1.0 |
| **Item** | User Accounts table (Output) | | |
| **Description** | Table listing existing user accounts with “Edit”, “Delete” buttons per row, plus “Create Account” and “Go back” buttons | | |
| **Purpose** | To present all user records and available actions in one view | | |
| **Format** | Table with action buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Manage User Accounts” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI703 | **Version** | 1.0 |
| **Item** | “Edit” button (Input) | | |
| **Description** | Button on each user‐row to open the edit‐account screen | | |
| **Purpose** | To allow the admin to modify a specific user’s details and roles | | |
| **Format** | Button (per row) | **Valid Range** | Not applicable |
| **Related I/O** | Output: Edit Account screen | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI704 | **Version** | 1.0 |
| **Item** | Edit Account screen (Output) | | |
| **Description** | Form displaying user fields, roles/permissions checkboxes, with “Save” and “Cancel” buttons | | |
| **Purpose** | To allow the admin to update account details and manage permissions | | |
| **Format** | Screen/Form with buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Edit” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI705 | **Version** | 1.0 |
| **Item** | “Save” button (Input) | | |
| **Description** | Button on the Edit Account screen to submit changes | | |
| **Purpose** | To finalize and persist modifications to the user record | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Save Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI706 | **Version** | 1.0 |
| **Item** | Save Confirmation dialog (Output) | | |
| **Description** | Modal asking “Do you want to save changes to this account?” | | |
| **Purpose** | To confirm the admin’s intent before updating | | |
| **Format** | Modal with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Save” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI707 | **Version** | 1.0 |
| **Item** | “Cancel” button (Input) | | |
| **Description** | Button on the Edit Account screen to abandon edits | | |
| **Purpose** | To allow the admin to exit without saving changes | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Cancel‐Without‐Save dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |
| **Requirement ID** | REQ\_UI708 | **Version** | 1.0 |
| **Item** | Cancel‐Without‐Save dialog (Output) | | |
| **Description** | Modal asking “Discard changes and return to account list?” | | |
| **Purpose** | To verify the admin’s choice before losing edits | | |
| **Format** | Modal with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Cancel” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI709 | **Version** | 1.0 |
| **Item** | “Delete” button (Input) | | |
| **Description** | Button on each user‐row to initiate account removal | | |
| **Purpose** | To allow the admin to remove a user record | | |
| **Format** | Button (per row) | **Valid Range** | Not applicable |
| **Related I/O** | Output: Delete Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI710 | **Version** | 1.0 |
| **Item** | Delete Confirmation dialog (Output) | | |
| **Description** | Modal asking “Are you sure you want to delete this user?” | | |
| **Purpose** | To confirm the admin’s intent before deleting | | |
| **Format** | Modal with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Delete” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI711 | **Version** | 1.0 |
| **Item** | Deletion Success message (Output) | | |
| **Description** | Inline toast/banner reading “User account deleted successfully” | | |
| **Purpose** | To inform the admin that the removal was processed | | |
| **Format** | Toast/banner notification | **Valid Range** | Not applicable |
| **Related I/O** | Input: Delete Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI712 | **Version** | 1.0 |
| **Item** | “Create Account” button (Input) | | |
| **Description** | Button at bottom of the User Accounts table to open the account‐creation form | | |
| **Purpose** | To allow the admin to add a new user record | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Create Account screen | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI713 | **Version** | 1.0 |
| **Item** | Create Account screen (Output) | | |
| **Description** | Form to enter new user details, roles, and permissions, with “Create” and “Cancel” buttons | | |
| **Purpose** | To collect and submit new account information | | |
| **Format** | Screen/Form with buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Create Account” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI714 | **Version** | 1.0 |
| **Item** | “Create” button (Input) | | |
| **Description** | Button on the Create Account screen to finalize new account | | |
| **Purpose** | To submit and persist the new user record | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Create Confirmation dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI715 | **Version** | 1.0 |
| **Item** | Create Confirmation dialog (Output) | | |
| **Description** | Modal asking “Confirm creation of new user account?” | | |
| **Purpose** | To verify the admin’s intent before adding | | |
| **Format** | Modal with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Create” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI716 | **Version** | 1.0 |
| **Item** | “Cancel” button (Input) on Create screen | | |
| **Description** | Button to abandon new‐account form | | |
| **Purpose** | To exit account creation without saving | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Cancel‐Create dialog | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI717 | **Version** | 1.0 |
| **Item** | Cancel‐Create dialog (Output) | | |
| **Description** | Modal asking “Discard new account and return to list?” | | |
| **Purpose** | To confirm before losing entered data | | |
| **Format** | Modal with “Yes”/“No” buttons | **Valid Range** | Not applicable |
| **Related I/O** | Input: Cancel button on Create screen | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI718 | **Version** | 1.0 |
| **Item** | “View System Logs” menu item (Input) | | |
| **Description** | Menu item or button on the admin page to open logs view | | |
| **Purpose** | To allow the admin to access log‐viewing functionality | | |
| **Format** | Menu item / Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Filter controls | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI719 | **Version** | 1.0 |
| **Item** | Filter controls panel (Output) | | |
| **Description** | UI elements for date range, severity, and source filtering | | |
| **Purpose** | To enable precise log queries by the admin | | |
| **Format** | Panel with date‐pickers, dropdowns, checkboxes | **Valid Range** | Not applicable |
| **Related I/O** | Input: “View System Logs” menu item | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI720 | **Version** | 1.0 |
| **Item** | “Search” button (Input) | | |
| **Description** | Button to submit the filter criteria | | |
| **Purpose** | To trigger retrieval of matching log entries | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Log Entries table | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI721 | **Version** | 1.0 |
| **Item** | Log Entries table (Output) | | |
| **Description** | Paginated table showing timestamp, actor, action, severity, and message summary | | |
| **Purpose** | To display the filtered log results to the admin | | |
| **Format** | Table with pagination controls | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Search” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI722 | **Version** | 1.0 |
| **Item** | Log Details panel (Output) | | |
| **Description** | Panel or modal showing full log entry details, including stack trace | | |
| **Purpose** | To allow the admin to drill down into a specific log event | | |
| **Format** | Panel / Modal dialog | **Valid Range** | Not applicable |
| **Related I/O** | Input: row selection in Log Entries table | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI723 | **Version** | 1.0 |
| **Item** | “Export” button (Input) | | |
| **Description** | Button to export current filtered logs to CSV or JSON | | |
| **Purpose** | To allow the admin to download logs for offline analysis | | |
| **Format** | Button | **Valid Range** | Not applicable |
| **Related I/O** | Output: Export file link | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI724 | **Version** | 1.0 |
| **Item** | Export file link (Output) | | |
| **Description** | Download link to the generated CSV or JSON log file | | |
| **Purpose** | To provide the admin with the exported data | | |
| **Format** | Hyperlink / Button | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Export” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_UI725 | **Version** | 1.0 |
| **Item** | “No records found” message (Output) | | |
| **Description** | Inline text shown when no logs match the filter | | |
| **Purpose** | To inform the admin that the query returned no results | | |
| **Format** | Inline text / Banner | **Valid Range** | Not applicable |
| **Related I/O** | Input: “Search” button | | |
| **Author** | Nicholas Thong Meng Shui | | |

### 3.4.3 Hardware Interfaces

The system should support the following hardware-related considerations:

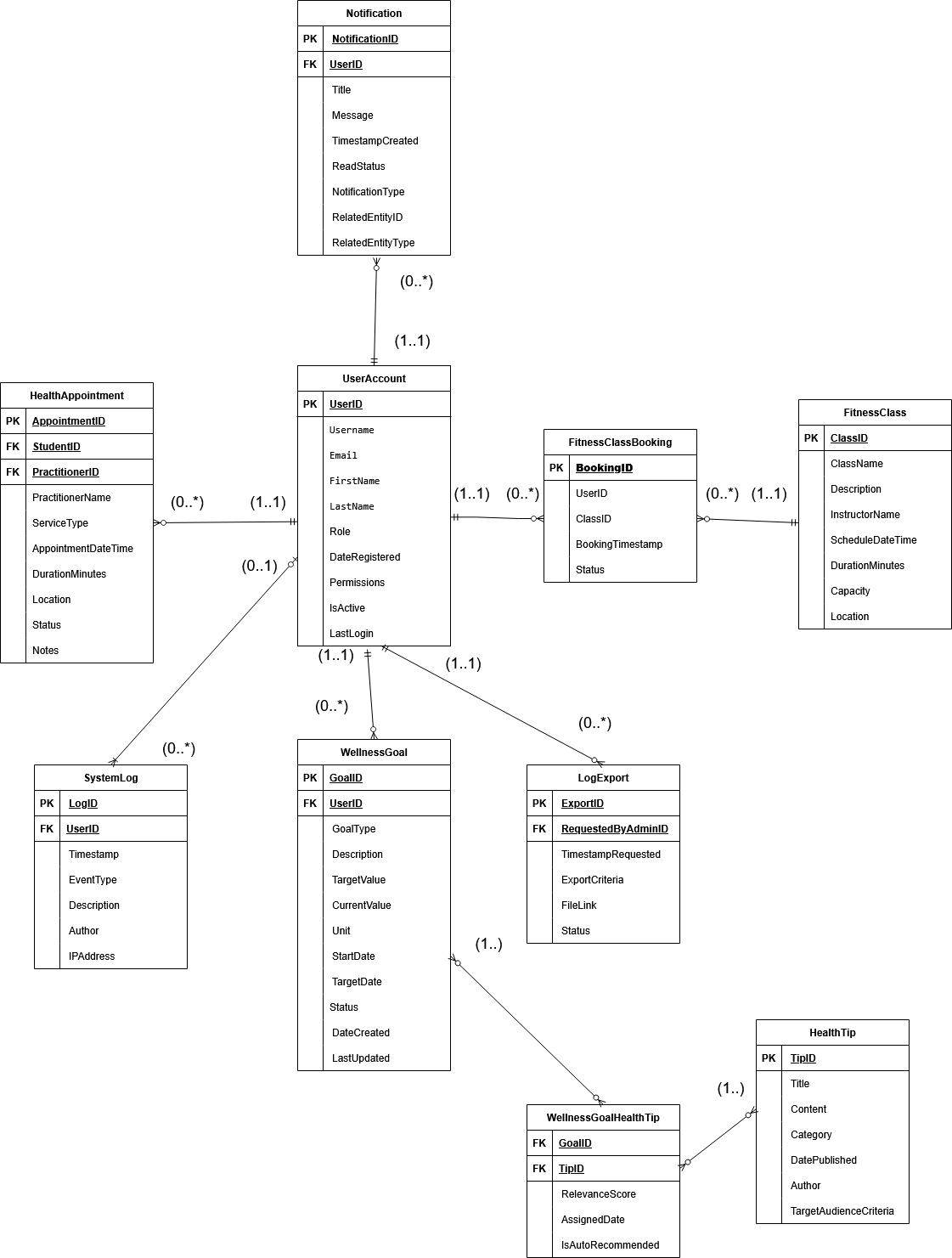
#### 3.4.3.1 Client Devices:

* Personal computers (PC) both laptops and desktops (Windows, macOS and Linux)
* Mobile devices (iOS and Android)

#### 3.4.3.2 Server Infrastructure:

* Campus Wellness Portal shall be deployed on university hosted servers.
* Supports standard networking, monitoring and backup operations.

## 3.5 Logical Database Requirements

Below is the ERD diagram:

*Figure 15: ERD diagram*

Below is the data dictionary:

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Field Name** | **Data Type** | **Description** |
| **UserAccount** | UserID | INT (PK) | Unique identifier for the user |
|  | Username | VARCHAR | User's login name |
|  | PasswordHash | VARCHAR | Hashed password |
|  | Email | VARCHAR | User's email address |
|  | FirstName | VARCHAR | User's first name |
|  | LastName | VARCHAR | User's last name |
|  | Role | ENUM('Student', 'Admin', 'Practitioner') | Role of the user |
|  | DateRegistered | TIMESTAMP | Account creation timestamp |
|  | Permissions | TEXT/JSON (Optional) | Custom permissions or overrides |
| **FitnessClass** | ClassID | INT (PK) | Unique class identifier |
|  | ClassName | VARCHAR | Name of the class |
|  | Description | TEXT | Description of the class |
|  | InstructorName | VARCHAR | Name of the instructor |
|  | ScheduleDateTime | TIMESTAMP | Scheduled date and time |
|  | DurationMinutes | INT | Duration in minutes |
|  | Capacity | INT | Maximum number of participants |
|  | Location | VARCHAR | Class location |
|  | DifficultyLevel | VARCHAR | Difficulty level (e.g., Beginner, Intermediate) |
| **FitnessClassBooking** | BookingID | INT (PK) | Unique booking identifier |
|  | UserID | INT (FK → UserAccount.UserID) | Student who made the booking |
|  | ClassID | INT (FK → FitnessClass.ClassID) | Booked class |
|  | BookingTimestamp | TIMESTAMP | Booking creation time |
|  | Status | ENUM('Booked', 'Cancelled', 'Attended') | Booking status |
| **HealthAppointment** | AppointmentID | INT (PK) | Unique appointment ID |
|  | StudentID | INT (FK → UserAccount.UserID) | Student who booked the appointment |
|  | PractitionerID | INT (FK → UserAccount.UserID, Nullable) | Practitioner assigned (nullable) |
|  | PractitionerName | VARCHAR | Name of external practitioner (optional) |
|  | ServiceType | VARCHAR | Type of service (e.g., Consultation, Check-up) |
|  | AppointmentDateTime | TIMESTAMP | Appointment date and time |
|  | DurationMinutes | INT | Appointment duration in minutes |
|  | Location | VARCHAR | Appointment location |
|  | Status | ENUM('Scheduled', 'Cancelled', 'Completed', 'NoShow') | Status of the appointment |
|  | Notes | TEXT (Nullable) | Additional comments or notes |
| **WellnessGoal** | GoalID | INT (PK) | Unique goal ID |
|  | UserID | INT (FK → UserAccount.UserID) | Owner of the goal |
|  | GoalType | VARCHAR | Type of goal (e.g., Weight, Steps) |
|  | Description | TEXT (Nullable) | Goal description |
|  | TargetValue | DECIMAL or VARCHAR | Target value to achieve |
|  | CurrentValue | DECIMAL or VARCHAR (Nullable) | Progress made |
|  | Unit | VARCHAR | Unit of measurement |
|  | StartDate | DATE | When the goal begins |
|  | TargetDate | DATE | Deadline to achieve the goal |
|  | Status | ENUM('Active', 'Achieved', 'Abandoned') | Goal status |
|  | DateCreated | TIMESTAMP | Goal creation date |
|  | LastUpdated | TIMESTAMP | Last modified date |
| **HealthTip** | TipID | INT (PK) | Unique tip ID |
|  | Title | VARCHAR | Title of the tip |
|  | Content | TEXT | Description or guidance |
|  | Category | VARCHAR (Nullable) | Category (e.g., Nutrition, Exercise) |
|  | DatePublished | TIMESTAMP | Date the tip was published |
|  | Author | VARCHAR (Nullable) | Source or author of the tip |
|  | TargetAudienceCriteria | TEXT/JSON (Nullable) | Optional filters for personalization |
| **Notification** | NotificationID | INT (PK) | Unique notification ID |
|  | UserID | INT (FK → UserAccount.UserID) | Notification recipient |
|  | Title | VARCHAR | Notification title |
|  | Message | TEXT | Content of the message |
|  | TimestampCreated | TIMESTAMP | Time the notification was created |
|  | ReadStatus | BOOLEAN | Whether the notification is read |
|  | NotificationType | VARCHAR (Nullable) | Type of notification |
|  | RelatedEntityID | INT (Nullable) | Related entity (e.g., Class ID) |
|  | RelatedEntityType | VARCHAR (Nullable) | Type of the related entity |
| **SystemLog** | LogID | INT (PK) | Unique log ID |
|  | Timestamp | TIMESTAMP | Log timestamp |
|  | EventType | VARCHAR | Type of event (e.g., Login, Update) |
|  | Description | TEXT | Event description |
|  | UserID | INT (FK → UserAccount.UserID, Nullable) | User who triggered the event |
|  | IPAddress | VARCHAR (Nullable) | Source IP address |
| **LogExport** | ExportID | INT (PK) | Unique export identifier |
|  | RequestedByAdminID | INT (FK → UserAccount.UserID) | Admin who requested the export |
|  | TimestampRequested | TIMESTAMP | Export request time |
|  | ExportCriteria | TEXT/JSON (Nullable) | Filter criteria used |
|  | FileLink | VARCHAR (Nullable) | Path to exported file |
|  | Status | ENUM('Pending', 'Processing', 'Completed', 'Failed') | Status of export |
| **WellnessGoalHealthTip** | GoalID | INT (FK → WellnessGoal.GoalID) | Related goal |
|  | TipID | INT (FK → HealthTip.TipID) | Related health tip |
|  | RelevanceScore | DECIMAL | How relevant the tip is to the goal |
|  | AssignedDate | TIMESTAMP | When the tip was linked to the goal |
|  | IsAutoRecommended | BOOLEAN | Whether the tip was auto-assigned or manually chosen |

## Design Constraints

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_DC001 | System shall be compatible with latest versions of major browsers. | High | Fikrul Amsyar Azmin |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_DC002 | The interface design shall comply with university’s branding guidelines. (fonts, color palette, logo placement, and illustrations) | High | Fikrul Amsyar Azmin |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_DC003 | The system shall be responsive and usable on desktops, laptops, and mobile devices. | High | Fikrul Amsyar Azmin |

## 3.7 Software System Attributes

This section specifies the attributes that are expected from the Wellness Portal and its implementations.

### 3.7.1 Security

The security requirements for the Wellness Portal are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_Q001 | The Wellness Portal may only be accessed by students of the university. | High | Muhammad Anas |

## Supporting Information

### 3.8.1 Sample Input/Output Formats

For exporting system log files, data can be exported in either JSON or CSV format.

### 3.8.2 Problems to Be Solved by the Software

* Organize the university’s many health-related facilities into a universal software (Wellness Portal).

# 4.0 Verification

This section describes the approach and criteria that will be used to verify that the Campus Wellness Portal system meets its functional and non-functional requirements. This is done to ensure that the stakeholder’s expectations are met, and that the system performs in accordance to the specification.

## 4.1 Verification Approach

Verification of the Campus Wellness Portal will be conducted using various software testing techniques throughout the development cycle. The table below lists out the verification method, responsible parties, timeline and location for the techniques.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Verification Technique** | **Description** | **Responsible Party(s)** | **Timeline** | **Location** |
| **Unit Testing** | Tests that individual modules function as intended. | Developer | Week 3-6 | Developer environment |
| **Integration Testing** | Tests that integration/interaction between modules functions as intended. | Developer, QA Tester | Week 7-9 | Staging/test environment |
| **Functional Testing** | Tests that the functions (F001-F011) meet requirements. | QA Team, Project Supervisor | Week 9-10 | University Test Lab |
| **System Testing** | Assess the entire system behaviour alongside performance and security. | QA Team | Week 10-11 | Controlled Lab/Test Server |
| **User Acceptance Testing (UAT)** | Tests with real users to confirm that it meets real-world needs. | End-user, QA Team | Week 11-12 | Campus Lab or Virtual Environment |

## 4.2 Verification Criteria

### 4.2.1 Unit testing verification criteria

|  |  |  |
| --- | --- | --- |
| **Module** | **Test Case** | **Expected Output** |
| Login | Users login to their own accounts. | Returns the campus wellness dashboard upon successful login. |
| Book Fitness Class | Book fitness classes. | Return successful booking message to the user. |
| Cancel Fitness Class | Cancel previously booked fitness classes. | Return successful booking cancellation to the user. |
| Schedule Health Centre Appointment | Schedule new health centre appointment. | Updates user appointments database and returns successful appointment booking to the user. |
| Cancel Health Centre Appointment | Cancel previously scheduled health centre appointments. | Removes selected appointment from user’s scheduled appointments table in database. |
| Track Wellness Goals | Track user wellness goals. | Return updated wellness goals in student dashboard to the user. |
| View System Notifications | View pending notifications. | Returns notifications regarding health centre/fitness class bookings to the user. |
| View Personalized Health | View personalized health tips. | Returns personalized health tips to the user. |
| Manage Health Centre Appointments | View user’s health centre appointments. | Returns the health centre appointment dashboard to the user. |
| Manage Fitness Class Booking | View booked fitness class booking. | Returns the fitness class dashboard to the user. |
| Manage User Access Control | Create, modify, and delete user accounts and assign or revoke roles/permissions. | Return the users access dashboard to the user. |
| View System Logs | View system logs | Return the system logs. |

### 4.2.2 Integration testing verification criteria

|  |  |  |
| --- | --- | --- |
| **Integrated Modules** | **Test Case** | **Expected Outcome** |
| Book Fitness Class + Cancel Fitness Class + Manage Fitness Class Bookings | Users view the fitness class bookings and can book new fitness class sessions and cancel previously booked classes. | Fitness class dashboard displays correctly and success message pops up for successful booking and cancellation of fitness class. |
| Schedule Health Centre Appointment + Cancel Health Centre Appointment + Manage Health Centre Appointment | User views their scheduled health centre appointments and can schedule a new appointment or cancel a scheduled appointment. | Health centre appointment dashboard displays correctly and success message pops up for successful booking and cancellation of health centre appointment. |

### 4.2.3 Functional testing verification criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Code** | **Feature** | **Test Case** | **Expected Outcome** |
| F001 | Login | Login with valid credentials. | Redirected to campus wellness portal dashboard |
| F002 | Book Fitness Class | Book a fitness class. | Fitness class successfully booked |
| F003 | Cancel Fitness Class | Cancel a booked fitness class. | Fitness class successfully cancelled |
| F004 | Schedule Health Centre Appointment | Schedule a new health centre appointment. | Health centre appointment successfully scheduled. |
| F005 | Cancel Health Centre Appointment | Cancel a scheduled health centre appointment. | Health centre appointment successfully cancelled |
| F006 | Track Personal Wellness Goals | View personal wellness goals. | Redirect to personal wellness goals dashboard |
| F007 | View System Notifications | View system notifications (fitness class and health centre appointments) | Redirect to notifications dashboard. |
| F008 | View Personalized Health Tips | View personalized health tips | Redirect to personalized health tips dashboard. |
| F009 | Manage Health Centre Appointments | View scheduled health centre appointments. | Redirect to health centre appointments dashboard. |
| F010 | Manage Fitness Class Bookings | View fitness class bookings. | Redirect to fitness class bookings dashboard. |
| F011 | Manage User Access Control | Manage users access to the system. | Redirect to user access dashboard. |
| F012 | View System Logs | View past system logs. | Shows the past system logs. |

### 4.2.4 User acceptance testing (UAT) verification criteria

|  |  |  |
| --- | --- | --- |
| User Role | UAT Scenario | Acceptance Criteria |
| Student | Login, book fitness class, cancel fitness class, manage fitness class bookings, manage health centre appointments, schedule health centre appointment, cancel health centre appointment, view health tips | All actions complete without errors and are easy to navigate. |
| Admin | Manage user access control, view system logs | User access can be changed and system logs can be viewed. |

### 4.2.5 Performance Verification Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Verification Criteria | Priority |
| REQ\_P001 | The average response time of Wellness Portal functions shall be less than 2 seconds. | Wellness Portal functions must load and complete within <= 2 seconds. | High |

### 4.2.6 Usability Verification Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Verification Criteria | Priority |
| REQ\_UR001 | System shall be intuitive enough for first-time users to perform basic tasks without extensive training. | Users are able to navigate the system and interact with it and use its functions without much instructions | High |
| REQ\_UR002 | Consistent design patterns and navigation shall be used across pages. | Design language of the system UI must be uniform and consistent across the whole system. | Medium |
| REQ\_UR003 | System shall prevent user errors through input validation, tooltips, and disabling invalid options. | Tooltips show up on buttons where applicable, input validation is done for input boxes and invalid options are disabled for selection boxes. | High |

# 5.0 Appendices

## 5.1 Assumptions and Dependencies

### 5.1.1 University Authentication System

Type: OAuth 2.0 Single Sign-on (SSO)

Role: Secure login and access control

Dependency: The Wellness Portal depends on this framework to validate user login and assign permissions. Any downtime in the SSO will affect all user access.

Type: JSON Web Token Standard (JWT)

Role: Verify user identities

Dependency: The Wellness Portal depends on this system to differentiate between normal users and system administrators. It works in conjunction with OAuth 2.0 for handling user authentication.

### 5.1.2 Fitness Class Booking

Type: REST API integration

Role: Book fitness classes and cancel booked fitness classes.

Dependency: The Wellness Portal depends on this system to handle fitness class availability.

### 5.1.3 Health Centre Appointment Booking

Type: REST API integration

Role: Schedule health centre appointments and cancel health centre appointments.

Dependency: The Wellness Portal depends on this system to handle health centre appointment availability.

### 5.1.4 Browser Standards

Type: Client-side software

Role: Interface in which users will access the Wellness Portal

Dependency: The Wellness Portal will assume compliance with ECMAScript and HTML5 standards. Use of unsupported browsers may cause UI/UX issues.

### 5.1.5 Stakeholder Availability

Type: Organizational

Role: Provide input, feedback and UAT participation

Dependency: Lack of participation from students and staff may put requirements validation and UAT at risk.

## 5.2 Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Acronym/Abbreviation** | **Meaning** |
| **JWT** | JSON Web Token |
| **RGB** | Red, Green, Blue (values for colour) |
| **JSON** | JavaScript Object Notation |
| **HTTP** | Hyper Text Transfer Protocol |
| **UAT** | User Acceptance Testing |
| **HIPAA** | Health Insurance Portability and Accountability Act |
| **SaaS** | Software as a Service |
| **JWT** | JSON Web Token |
| **CSV** | Comma Separated Value (A file type) |
| **UI** | User Interface |