**Requirements**

Ali Suhail 21072712  
Last updated: March 22, 2024

Contents

[1. Project Aim 3](#_Toc162750042)

[2. Project Objectives 3](#_Toc162750043)

[3. Pictures and Scenarios 4](#_Toc162750044)

[3.1. Alistana Fitness and Nutrition Tracker Application 4](#_Toc162750045)

[3.2. Admin Management Website 5](#_Toc162750046)

[4. Usecase Diagrams 6](#_Toc162750047)

[4.1. AFNT Application 6](#_Toc162750048)

[4.2. Admin Management Website 12](#_Toc162750049)

[4.3. Arduino Watch 13](#_Toc162750050)

[5. Functional Requirements 15](#_Toc162750051)

[5.1. Database 15](#_Toc162750052)

[5.2. Website 17](#_Toc162750053)

[5.3. Application 18](#_Toc162750054)

[5.4. Arduino Watch: 22](#_Toc162750055)

[6. Non-Functional Requirements 23](#_Toc162750056)

[6.1. Application: 23](#_Toc162750057)

[6.2. Performance: 25](#_Toc162750058)

[6.3. Efficiency & Sustainability: 26](#_Toc162750059)

[6.4. Data Storage Optimization: 26](#_Toc162750060)

[6.5. Privacy & Security: 27](#_Toc162750061)

[6.6. Reliability: 28](#_Toc162750062)

[6.7. Usability: 29](#_Toc162750063)

[6.8. Data Backup & Recovery: 30](#_Toc162750064)

[6.9. Third-Party Service Integration: 30](#_Toc162750065)

Table of Contents

[Table 1: Usecase Manage Workout and the <<extend>>. 7](#_Toc162750030)

[Table 2: Usecase Manage Meal and the <<extend>>. 7](#_Toc162750031)

[Table 3: Usecase Manage Water Intake and the <<extend>>. 8](#_Toc162750032)

[Table 4: Usecase Gym Finder. 8](#_Toc162750033)

[Table 5: Usecase Create Account. 9](#_Toc162750034)

[Table 6: Usecase Update Profile. 9](#_Toc162750035)

[Table 7: Usecase Delete Profile. 10](#_Toc162750036)

[Table 8: Usecase Track Workout Progress and <<extend>>. 10](#_Toc162750037)

[Table 9: Usecase Generate Progress Graphs and the <<extend>>. 11](#_Toc162750038)

[Table 10: Usecase Manage Website and the <<extend>>. 12](#_Toc162750039)

[Table 11: Usecase Store Measured Data and the <<extend>>. 14](#_Toc162750040)

[Table 12: Usecase Sync Watch Data to Application. 14](#_Toc162750041)

1. Project Aim

To design and implement a software solution focused on helping users achieve their fitness objectives. This involves developing a user-friendly platform for creating and customising workout routines, monitoring nutrition and workout progress, and offering health and fitness guidance. The main aim is to build an engaging and efficient fitness tool that encourages users to live healthier, more active lives.

1. Project Objectives

Please note all requirements will be marked by MoSCow Prioritization technique.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Objective** | **Description** | **Created** |
| O1 | Develop a Scalable and Secure Database Management System (DBMS) | Develop a scalable MySQL DBMS, including a central database for 2,000+ records and a local database for user data and watch data. Improve database performance and security and ensure GDPR compliance. | 09/10/2023 |
| O2 | Build the Alistana Fitness & Nutrition Tracker (AFNT) Application | Develop the AFNT application with features for workout and nutrition tracking, body progress and measurement tracking. Should be connected to DBMS, Admin Management website (only for Admins), and Arduino watch. | 09/10/2023 |
| O3 | Design a User-Centric Admin Management (AM) Website | Develop a responsive website with secure login and allows Admins to edit the central database (user login data and preset workout and meal data) and app push updates to the AFNT application. The website will prioritize a user-friendly design, encrypted communication, and security measures. | 09/10/2023 |
| O4 | Design and develop a Fitness Watch using Arduino | Create an Arduino-based Fitness watch to measure blood oxygen level, heart rate and step count, and investigate ways of connecting the Arduino watch to the AFNT and store body data in the local database via AFNT. | 16/10/2023 |
| O5 | Enhance Code Quality and Performance | Implement clean, maintainable code with 80% code coverage. Optimize application and website response times to under 2 and 3 seconds, respectively. | 25/10/2023 |

1. Pictures and Scenarios
   1. Alistana Fitness and Nutrition Tracker Application

A computer screen shot of a diagram

Description automatically generated

Figure 1: AFNT UI Sketches.

* 1. Admin Management Website

A computer screens with different colored lines

Description automatically generated with medium confidence

Figure 2: AM Website UI Sketches.

1. Usecase Diagrams
   1. AFNT Application

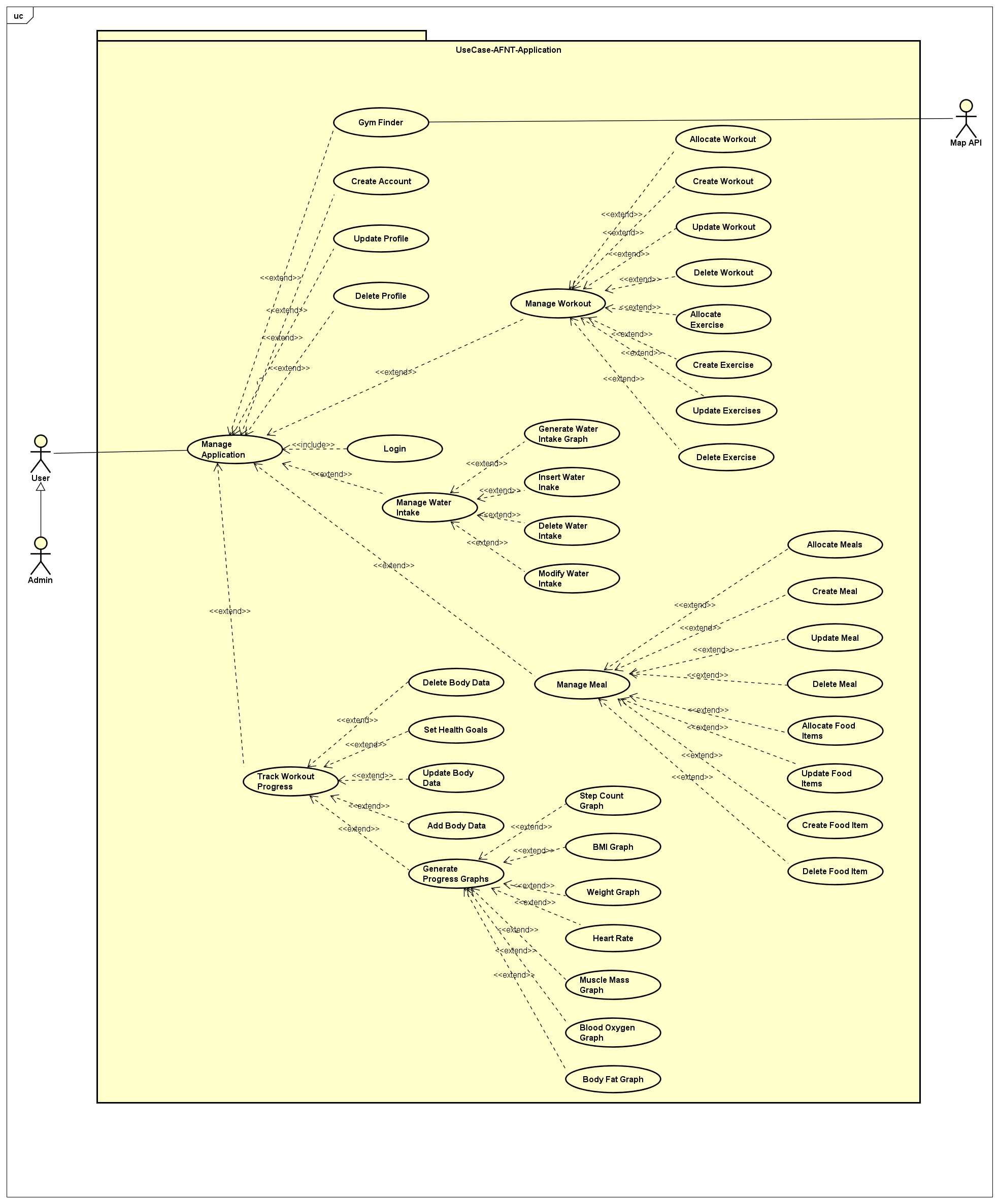


Figure 3: AFNT App UseCase Diagram.

Table 1: Usecase Manage Workout and the <<extend>>.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Manage Workout and the <<extend>> |
| **Goal** | Manage Workouts and Exercises (Add/Update/Delete) |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | Details of Workouts and Exercises are stored in the LocalDB database and to access its contents and modify data, the User must be in the "Workout" tab. |
| **Post-conditions** | Details of Workouts and Exercises are displayed to the User/Admin |
| **Basic Flow of Events** | 1. User navigates to the 'Workout' tab. 2. App displays all workouts planned for the current week. 3. User can change the date range to see workouts in the past and future 4. User can select the workout and view the exercises allocated in the selected workout.  5. The page includes buttons that allow user to allocate, create, edit, and delete workouts and exercises. |
| **Alternative Flow** | None |

Table 2: Usecase Manage Meal and the <<extend>>.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Manage Meal and the <<extend>> |
| **Goal** | Manage Meals and Food Items (Add/Update/Delete) |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | Details of Meals and Food Items are stored in the LocalDB database and to access its contents and modify data, the User must be in the "Meal" tab. |
| **Post-conditions** | Details of Meals and Food Items are displayed to the User/Admin |
| **Basic Flow of Events** | 1. User navigates to the 'Meal' tab. 2. App displays all meals planned for the current week. 3. User can change the date range to see meals in the past and future 4. User can select the meal and view the food items allocated in the selected meal.  5. The page includes buttons that allow user to allocate, create, edit, and delete meals and food items. |
| **Alternative Flow** | None |

Table 3: Usecase Manage Water Intake and the <<extend>>.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Manage Water Intake and the <<extend>> |
| **Goal** | Manage Water Intake (Add/Update/Delete) |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | Details of Water Intake (ml) are stored in the LocalDB database and to access its contents and modify data, the User must be in the "Water Intake" tab. |
| **Post-conditions** | The screen contains a foam where the User can select the date and the amount of water drank. The use can also generate graphs of water intake per selected month/selected year |
| **Basic Flow of Events** | 1. User navigates to the 'Water Intake' tab. 2. App displays a form for adding water intake in the selected date (default date is the current date). 3. User can change the date to update/add the water intake and click 'Submit' to add/update records. 4. The page also contains a button to delete all water intake data. 5. The page also contains a button to generate two water intake graphs, one for per selected month, and one for the selected year |
| **Alternative Flow** | None |

Table 4: Usecase Gym Finder.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Gym Finder |
| **Goal** | Find nearest gym relative using user's location data. |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin MapAPI |
| **Pre-conditions** | User must consent to share their location |
| **Post-conditions** | The screen displays the nearest gym to users’ location around a 10-mile radius |
| **Basic Flow of Events** | 1. User navigates to the 'Gym Finder' tab. 2. App uses user's location data and fetches the nearest gyms to the user’s location. 3. User can generate a path to the selected gym and can download/save the location and directions. |
| **Alternative Flow** | None |

Table 5: Usecase Create Account.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Create Account |
| **Goal** | Users can create a new account. |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | User must navigate to the 'Register' screen |
| **Post-conditions** | The user must enter their details like username, password, gender, dob email, phone number and postcode (optional) |
| **Basic Flow of Events** | 1. User launches the AFNT app. 2. The app displays a login page, user needs to select the 'Register' button. 3. User needs to provide the details stated in post-conditions. 4. After entering valid data, then press 'Register'. 5. A success popup will be displayed, and the user can now login using the new account. |
| **Alternative Flow** | None |

Table 6: Usecase Update Profile.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Update Profile |
| **Goal** | Users can update their profile (Add/Update) |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | User must navigate to the 'Profile' screen by clicking the profile icon in the 'Dashboard' screen |
| **Post-conditions** | The user can modify details like username, password, gender, dob email, phone and postcode (optional) |
| **Basic Flow of Events** | 1. User opens the AFNT app 2. User logs in. 3. User is sent to the dashboard screen. 4. User clicks on the profile icon and clicks 'Profile'. 5. User can now modify the account details stated in post-conditions |
| **Alternative Flow** | None |

Table 7: Usecase Delete Profile.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Delete Profile |
| **Goal** | Users can delete their profile |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | User must navigate to the 'Profile' screen by clicking the profile icon in the 'Dashboard' screen |
| **Post-conditions** | The user can delete their profile (and all the data associated with it) |
| **Basic Flow of Events** | 1. User opens the AFNT app 2. User logs in. 3. User is sent to the dashboard screen. 4. User clicks on the profile icon and clicks 'Profile'. 5. There is a delete button in the bottom of the screen and clicking that will delete the profile and all the data  associated with it. The user is then sent back to the login screen. |
| **Alternative Flow** | None |

Table 8: Usecase Track Workout Progress and the <<extend>>.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Track Workout Progress and the <<extend>> |
| **Goal** | Users can track their workout progress |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | User must navigate to the 'Body Stats' screen in the 'Dashboard' screen |
| **Post-conditions** | The user can monitor and modify their workout progress data like step count, BMI, weight, heart rate, muscle mass, blood oxygen level and body fat data. |
| **Basic Flow of Events** | 1. User navigates to the dashboard screen. 2. User clicks on the 'Body Stats' tab. 3. Users can input their current body stat (weight, height, BMI etc.). User can also update and delete data. 4. User can also set health goals (i.e. set weight goal, step goal etc.) |
| **Alternative Flow** | None |

Table 9: Usecase Generate Progress Graphs and the <<extend>>.

|  |  |
| --- | --- |
| **Use Case Diagram** | AFNT Application |
| **Use Case Identifier** | Generate Progress Graphs and the <<extend>> |
| **Goal** | Users can generate graphs to track their workout progress |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | User must navigate to the 'Body Stats' screen in the 'Dashboard' screen. Then select one of the generate graphs button. |
| **Post-conditions** | The user can generate graph data for step count, BMI, weight, heart rate, muscle mass, blood oxygen level and body fat data. |
| **Basic Flow of Events** | 1. User navigates to the dashboard screen. 2. User clicks on the 'Body Stats' tab. 3. User needs to input the month and year and then select which body stat to generate a graph for. 4. The app will then display a monthly and yearly graph for the selected body stat. |
| **Alternative Flow** | None |

* 1. Admin Management Website

A diagram of a diagram

Description automatically generated

Figure 4: AM Website UseCase Diagram

Table 10: Usecase Manage Website and the <<extend>>.

|  |  |
| --- | --- |
| **Use Case Diagram** | Admin Management Website |
| **Use Case Identifier** | Manage Website and the <<extend>> |
| **Goal** | Admins and modify the Central Database and push updates to AFNT app |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | Admin |
| **Pre-conditions** | Admins needs to login, upon a successful login Admins are directed to the admin dashboard |
| **Post-conditions** | Admins need to navigate to the 'Edit Central DB' tab |
| **Basic Flow of Events** | 1. Admin logs on the website 2. Upon a successful login, admins are redirected to the Admin Dashboard. 3. Admins can navigate to 'Edit Central DB' and can modify its contents (consists of presets meals and workout data, and user login data). 4. After implementing successful changes, admins can push this and the LocalDB will import the new centralDB updates when user logins successfully. |
| **Alternative Flow** | None |

* 1. Arduino Watch

A diagram of a diagram

Description automatically generated

Figure 5: Arduino Watch UseCase Diagram

Table 11: Usecase Store Measured Data and the <<extend>>.

|  |  |
| --- | --- |
| **Use Case Diagram** | Arduino Watch |
| **Use Case Identifier** | Store Measured Data and the <<extend>> |
| **Goal** | The Arduino Watch can store and measure body data (Step Count, Heart Rate, Blood Oxygen level, Count Reps, and current Date and Time) |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | User must have an Arduino Watch which is turned on and straped properly on the wrist. |
| **Post-conditions** |  |
| **Basic Flow of Events** | 1. Turn on the Arduino Watch 2. Properly wear the Arduino Watch around the wrist. 3. Arduino Watch should start measuring real time heart rate, blood oxygen, step count and the time data and store this data in the microSD card, which can be accessed to view the data collected. |
| **Alternative Flow** | None |

Table 12: Usecase Sync Watch Data to Application.

|  |  |
| --- | --- |
| **Use Case Diagram** | Arduino Watch |
| **Use Case Identifier** | Sync Watch Data to Application |
| **Goal** | The Arduino Watch shall send the collected user data to the AFNT App via Bluetooth, which can be accessed and viewed on the AFNT App live. |
| **Priority** | M |
| **Updated** | 21/03/2024 |
| **Participating Actors** | User Admin |
| **Pre-conditions** | User must have an Arduino Watch which is turned on and straped properly on the wrist. |
| **Post-conditions** | The Arduino Watch shall be linked to AFNT App via Bluetooth. |
| **Basic Flow of Events** | 1. Turn on the Arduino Watch 2. Properly wear the Arduino Watch around the wrist. 3. On the AFNT App, go to 'Arduino Watch' tab and press sync. 4. The app should link with the Arduino Watch after successfully finding it, and the Arduino Watch shall start transmitting data to the AFNT App. |
| **Alternative Flow** | None |

1. Functional Requirements

The functional requirements are divided into 4 categories: Database, Website, Application and Arduino Watch. These categories may be divided into further sub-categories to make them more readable, making it easier to track progress.

* 1. Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database Management System (DBMS)** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| **Central Database (CDB)** | | | | |
| FD1 | The central database should be stored in a MySQL Server. | M | Partially Complete | 25/10/2023 |
| FD2 | Should store preset meals, food items, exercises, and workouts with attributes such as meal\_id, food\_item\_id, exercise\_id, and workout\_id. | M | Complete | 11/10/2023 |
| FD3 | Store user information in CDB with attributes such as user, email, password hash, type (User or Admin), gender, phone, address, and date created. | M | Complete | 25/10/2023 |
| FD4 | Support queries to retrieve preset meals, food items, exercises, and workouts based on various filters such as meal type, food category, exercise category, and workout difficulty. | M | Partially Complete | 25/10/2023 |
| FD5 | Provide data synchronization capabilities to update the local database with the latest preset data by comparing the timestamps of the local and central databases. | M | Partially Complete | 25/10/2023 |
| **Local Database (LDB)** | | | | |
| FD6 | The local database should be stored in the user drive using SQLite. | M | Complete | 25/10/2023 |
| FD7 | Store both preset and custom (combined as one) in tables: meals, meal\_logs food\_items, food\_item\_logs. exercises, exercise\_logs workouts, workout\_logs, heart rate, blood oxygen level, and step count. | M | Complete | 16/10/2023 |
| FD8 | Store user information in LDB with attributes such as user, email, password hash, type (User or Admin), gender, phone, address, and date created. | M | Complete | 16/10/2023 |
| FD9 | Support updates to custom meals, workouts, exercises, and health metrics by allowing users to add, modify, and delete records. | M | Complete | 09/10/2023 |
| FD10 | Store health metrics from the Arduino watch including step count, heart rate and blood oxygen level. | M | Complete | 17/10/2023 |
| FD11 | Store other metrics such as height (meters), weight (kg), BMI, skeletal muscle (kg), body fat (%) and water intake (ml), | M | Complete | 26/10/2023 |

* 1. Website

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Admin Management Website (AM)** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| FW1 | The AFNT App shall link to the AM website for Admin login. | S | Not Started | 09/10/2023 |
| FW2 | The website should only be accessible to Admins and can be accessed Online or via AFNT App. | S | Partially Complete | 09/10/2023 |
| FW3 | The website shall allow Admin users to log in securely with their credentials. | S | Complete | 09/10/2023 |
| FW6 | The website shall allow admins to modify user personal details (i.e., profile picture, name, age, gender, dob, email, password, phone, address, and postcode. | C | Not Implementing | 09/10/2023 |
| FW7 | The website shall allow admins to edit their own profiles (profile picture, name, age, gender, dob, email, password, phone, address, and postcode). | C | Not Implementing | 09/10/2023 |
| FW8 | The website should allow admins to add/modify/delete preset workout data | M | Not Implementing | 09/10/2023 |
| FW9 | The website should allow admins to add/modify/delete preset exercise data | M | Not Implementing | 09/10/2023 |
| FW10 | The website should allow admins to add/modify/delete preset meal data | M | Not Implementing | 09/10/2023 |
| FW11 | The website should allow admins to add/modify/delete preset food item data | M | Not Implementing | 09/10/2023 |
| FW12 | The website shall allow admins to push database updates to the AFNT application. | M | Not Implementing | 10/10/2023 |

* 1. Application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Alistana Fitness & Nutrition Tracker Application (AFNT)** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| **Login and Registration** | | | | |
| FA41 | The application shall allow users to login using their username and password credentials. | M | Complete | 17/10/2023 |
| FA42 | The application shall allow users to register, and they must provide details such as username, email, password, phone number, email address, address, date of birth and gender. | M | Complete | 17/10/2023 |
| **User Profile** | | | | |
| FA43 | The application shall allow users to modify their personal data such as profile picture, username, email, password, phone number, email address, address, date of birth and gender. | M | Partially Complete | 17/10/2023 |
| FA44 | The application shall allow users to delete their personal data. | M | Partially Complete | 17/10/2023 |
| FA45 | The application shall allow users to delete their account and all the data associated with it. | M | Partially Complete | 17/10/2023 |
| **Workout Tracker** | | | | |
| FA1 | The application shall allow users to create their own workout plans. | M | Partially Complete | 09/10/2023 |
| FA2 | The application shall allow users to customize their own workout plans. | M | Partially Complete | 09/10/2023 |
| FA3 | The application shall allow users to choose pre-designed workout plans based on their goals. | M | Partially Complete | 09/10/2023 |
| FA4 | The application shall allow users to receive suggested workouts based on fitness goals. | S | Partially Complete | 09/10/2023 |
| FA5 | The application shall allow users to track their workouts by recording exercises, sets, reps, and weights (For custom exercises). | M | Partially Complete | 09/10/2023 |
| FA6 | The application shall allow users to rate their performance for each exercise (For both preset and custom). | M | Partially Complete | 09/10/2023 |
| FA7 | The application shall allow users to rate their performance for each workout (For both preset and custom). | M | Partially Complete | 09/10/2023 |
| FA8 | The application shall allow users to view their workout logs (For both preset and custom). | M | Partially Complete | 09/10/2023 |
| FA9 | The application shall allow users to view their exercise logs (For both preset and custom). | M | Partially Complete | 09/10/2023 |
| FA10 | The application shall allow users to display graphs of each exercise progress (i.e., Bench press max weight every month). | M | Partially Complete | 09/10/2023 |
| FA11 | The application shall allow users to display graphs of cardio-related exercises (i.e., Step/Distance for treadmill session) by manually inputting the information. | M | Partially Complete | 23/10/2023 |
| FA12 | The application shall allow users to display graphs of cardio-related exercises (i.e., Step/Distance for treadmill session) provided by the Arduino watch. | M | Not Started | 23/10/2023 |
| FA13 | The application shall produce charts for weight in kilogram (per day). | M | Complete | 26/10/2023 |
| FA14 | The application shall allow users to set goals for workouts. | C | Not Started | 24/10/2023 |
| **Nutrition Tracker** | | | | |
| FA15 | The application shall allow users to select preset food items. | M | Partially Complete | 09/10/2023 |
| FA16 | The application shall allow users to select preset meals. | M | Partially Complete | 09/10/2023 |
| FA17 | The application shall allow users to create custom food items and define their nutritional contents (i.e., calories, protein, carbs, fats etc.) | M | Partially Complete | 09/10/2023 |
| FA18 | The application shall allow users to create custom meals and add food items (Both custom and preset) to the custom meal. | M | Partially Complete | 26/10/2023 |
| FA19 | The application shall allow users to modify custom food items. | M | Partially Complete | 09/10/2023 |
| FA20 | The application shall allow users to modify custom meals. | M | Partially Complete | 09/10/2023 |
| FA21 | The application shall allow users to input daily water intake in milliliters (per day). | M | Partially Complete | 09/10/2023 |
| FA22 | The application shall allow users to generate a daily calorie intake graph based on a selected date range. | M | Not Started | 24/10/2023 |
| FA23 | The application shall allow users to generate a daily nutritional content intake (i.e., Carbs, fat, protein intake per day) graph based on a selected date range. | M | Not Started | 24/10/2023 |
| FA24 | The application shall allow users to generate a daily water intake graph based on a selected date range. | M | Not Started | 24/10/2023 |
| FA25 | The application shall allow users to set goals for nutritional intake. | C | Not Started | 25/10/2023 |
| FA26 | The application shall allow users to set goals for their water intake | C | Not Started | 25/10/2023 |
| **Body Measurements Tracker** | | | | |
| FA27 | The application shall allow users to input their daily weight in kilograms (per day). | M | Complete | 26/10/2023 |
| FA28 | The application shall allow users to input their daily height in meters (per day). | M | Complete | 26/10/2023 |
| FA29 | The application shall allow users to input Skeletal muscle data in kilograms (per day). | M | Complete | 26/10/2023 |
| FA30 | The application shall allow users to input Body fat data in percentage (per day). | M | Complete | 26/10/2023 |
| FA31 | The application shall use weight and height data and calculate user’s BMI (per day). | M | Complete | 26/10/2023 |
| FA32 | The application shall produce charts for weight in kilogram (per day). | M | Complete | 26/10/2023 |
| FA33 | The application shall produce graphs for height change. | M | Complete | 26/10/2023 |
| FA34 | The application shall produce charts for BMI change. | M | Complete | 26/10/2023 |
| FA35 | The application shall allow users to set goals for step counts/distance covered per day. | M | Complete | 26/10/2023 |
| FA36 | The application shall allow users to generate a heart rate graph provided by the Arduino watch. | M | Partially Complete | 26/10/2023 |
| FA37 | The application shall allow users to generate a blood oxygen chart provided by the Arduino watch. | M | Partially Complete | 26/10/2023 |
| **Gym Locator** | | | | |
| FA38 | The application shall allow users to get a list of the nearest gyms based on a mapping API. | S | Partially Complete | 24/10/2023 |
| **Health & Nutrition Advice** | | | | |
| FA39 | The application shall allow users to access health-related advice. | W | Not Implementing | 09/10/2023 |
| FA40 | The application shall allow users to access fitness-related advice (i.e., correct exercise forms etc.). | W | Not Implementing | 09/10/2023 |

* 1. Arduino Watch:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arduino Fitness Watch** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| FAW1 | The watch shall display the current date and time. | M | Partially Complete | 16/10/2023 |
| FAW1 | The watch shall have an oximeter module attached to measure heart rate and blood oxygen levels in real-time. | M | Not Started | 16/10/2023 |
| FAW2 | The watch shall have an accelerometer module attached to measure steps in real-time. | M | Partially Complete | 16/10/2023 |
| FAW3 | The watch shall also use the accelerometer to count reps for the bench press in real-time. | M | Partially Complete | 18/10/2023 |
| FAW4 | The watch shall have a microSD storage unit to store body measurements in real-time. | M | Partially Complete | 18/10/2023 |
| FAW5 | The watch shall compile, and store measured data in a CSV file type. | M | Not Started | 18/10/2023 |
| FAW6 | The watch shall use a Bluetooth (low energy) module to transfer data to the application. | M | Not Started | 18/10/2023 |
| FAW7 | The watch shall also use a wired connection via a micro-USB port to transfer watch data to the application. | M | Partially Complete | 18/10/2023 |
| FAW8 | The application shall process and store watch data in the local database. | M | Not Started | 18/10/2023 |

1. Non-Functional Requirements

The non-functional requirements are divided into 9 categories: Application, Performance, Efficiency & Sustainability, Data Storage Optimization, Privacy & Security, Reliability, Usability, Data Backup & Recovery, and Third-Party Service Integration. These categories may be divided into further sub-categories to make them more readable, making it easier to track progress.

* 1. Application:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Alistana Fitness & Nutrition Tracker Application (AFNT)** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| **Workout Tracker** | | | | |
| NFA1 | Users can add an unlimited number of custom exercises | M | Complete | 25/10/2023 |
| NFA2 | Users can add an unlimited number of custom workouts | M | Complete | 25/10/2023 |
| NFA3 | Users cannot modify preset workouts and exercises. | M | Complete | 25/10/2023 |
| NFA4 | Users can allocate three workouts per day. | M | Complete | 25/10/2023 |
| NFA5 | Users can choose up to 20 exercises (incl. custom exercises) per workout. | M | Complete | 25/10/2023 |
| NFA6 | Users cannot modify preset workouts and exercises. | M | Complete | 25/10/2023 |
| **Nutrition Tracker** | | | | |
| NFA8 | Users can add an unlimited number of custom food items | M | Partially Complete | 25/10/2023 |
| NFA9 | Users can add an unlimited number of custom meals | M | Partially Complete | 25/10/2023 |
| NFA10 | Users cannot modify preset food items and meals. | M | Partially Complete | 25/10/2023 |
| NFA11 | Users can modify meal logs and food item serving size in that meal (Can be both custom or preset meal and food item.) | M | Partially Complete | 26/10/2023 |
| NFA12 | Users can only select up to 4 meals (Morning, Afternoon, Evening, and Dinner) per day. | M | Not Started | 25/10/2023 |
| NFA13 | Users can choose up to 8 food items (incl. custom food items) per meal. | M | Not Started | 25/10/2023 |
| NFA14 | Users cannot modify the nutrition content of preset meals and food items. | M | Partially Complete | 25/10/2023 |

* 1. Performance:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Performance** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFP1 | The AM website should load in under 5 seconds, ensuring optimal user experience | S | Complete | 09/10/2023 |
| NFP2 | The AM website shall respond to user input within 200 milliseconds, providing a smooth and responsive interaction. | S | Partially Complete | 09/10/2023 |
| NFP3 | The application shall load within a 5-second time frame, ensuring users can access the features promptly. | S | Complete | 09/10/2023 |
| NFP4 | The Arduino watch should display heart rate, blood oxygen level, steps count in real time on the watch display. | M | Not Started | 17/10/2023 |

* 1. Efficiency & Sustainability:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Efficiency & Sustainability** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFES1 | The AM website should have a memory usage of no more than 500 MB and a CPU usage of no more than 10% under normal operating conditions. | C | Complete | 09/10/2023 |
| NFES2 | The application should have a memory usage of no more than 250 MB and a CPU usage of no more than 5% under normal operating conditions. | C | Complete | 25/10/2023 |
| NFES3 | The watch should collect and store data with a latency of no more than 100 milliseconds and a storage usage of no more than 50 MB. | C | Not Started | 24/10/2023 |
| NFES4 | The watch should transfer data to the application with a latency of no more than 200 milliseconds and a data transfer rate of at least 1 Mbps. | C | Not Started | 24/10/2023 |
| NFES5 | Ensure database consistency and integrity by implementing ACID (Atomicity, Consistency, Isolation, Durability) properties in database transactions. | C | Complete | 25/10/2023 |

* 1. Data Storage Optimization:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Storage & Optimization** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFDS1 | Implement efficient data storage techniques such as indexing, partitioning, and normalization to ensure that the average database query time is no more than 200 milliseconds for read queries and no more than 500 milliseconds for write queries. | S | Complete | 09/10/2023 |

* 1. Privacy & Security:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Privacy & Security** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFPS1 | The system shall implement features such as data minimization, purpose limitation, data portability, and the right to be forgotten, to comply with GDPR guidelines. | C | Partially Complete | 09/10/2023 |
| NFPS2 | The system shall implement role-based access control, secure authentication and authorization mechanisms, and regular security audits to ensure data privacy and security for user personal and health data. | W | Partially Complete | 09/10/2023 |
| NFPS3 | User data, both at rest and in transit, shall be encrypted using industry-standard encryption algorithms (e.g., AES-256) to ensure data is stored and transmitted safely. | C | Partially Complete | 24/10/2023 |
| NFPS4 | The Bluetooth or wired data transfer from the Arduino watch to the app shall use secure protocols (e.g., TLS/SSL) and industry-standard encryption algorithms (e.g., AES-256) to ensure the data is transferred securely and encrypted. | S | Not Started | 24/10/2023 |

* 1. Reliability:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reliability** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFR1 | The AM website will have an uptime of at least 99.9% by implementing robust error handling and failover mechanisms. | S | Partially Complete | 10/10/2023 |
| NFR2 | The application shall implement robust error handling to ensure that any errors are gracefully handled, and logged for analysis and that the application recovers without crashing. | M | Partially Complete | 24/10/2023 |
| NFR3 | The database and server shall implement robust error handling to automatically recover from failures and backup mechanisms to ensure data is backed up at least once a day and can be restored within 24 hours in case of data loss. | M | Partially Complete | 09/10/2023 |
| NFR4 | The Arduino hardware shall be constructed with materials that meet industry standards for wearables. | C | Partially Complete | 16/10/2023 |
| NFR5 | The smartwatch shall have a power management system that optimizes battery usage to ensure at least 12 hours of continuous operation on a single charge under normal usage conditions. | C | Partially Complete | 16/10/2023 |

* 1. Usability:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Usability** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFU1 | The AM website shall follow WCAG 2.1 AA compliance, as verified by automated testing tools. | C | Partially Complete | 09/10/2023 |
| NFU2 | The application shall have a user-friendly interface with a System Usability Scale (SUS) score of at least 70, indicating good usability. | C | Partially Complete | 24/10/2023 |
| NFU3 | The AM website UI shall have a navigation menu that is clearly visible and accessible from all pages, and a user flow that requires no more than three clicks to reach any page. | S | Partially Complete | 25/10/2023 |
| NFU4 | The website and application shall implement features such as screen reader compatibility, keyboard navigation, and text alternatives for non-text content to enhance accessibility for users with disabilities. | W | Not Implementing | 25/10/2023 |
| NFU5 | The application shall synchronize health data from the smartwatch to the app by pressing a sync button, ensuring that the data is up-to-date and accurate. | M | Not Started | 16/10/2023 |
| NFU6 | The application shall display clear and informative error messages in case of data transfer issues from the watch and provide a troubleshooting guide in the help section of the app. | S | Not Started | 16/10/2023 |
| NFU7 | The application should be compatible with mobile platforms, providing seamless functionality and usability across various mobile devices.  The responsiveness and compatibility standards should adhere to  modern design principles, facilitating accessibility and usability for  users on different devices. | S | Partially Complete | 16/10/2023 |

* 1. Data Backup & Recovery:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Backup & Recovery** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFDBR1 | The system shall have an automated backup and recovery system that creates daily backups of the database and stores them in a secure location, with the ability to restore data within 24 hours in case of data loss. | M | Complete | 09/10/2023 |
| NFDBR2 | The program's source code shall be backed up regularly to GitHub, with automated daily backups to OneDrive for additional redundancy. The backup system should support version control, allowing for the recovery of specific versions of the code if needed. | M | Complete | 09/10/2023 |

* 1. Third-Party Service Integration:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Third-Party Service Integration** | | | | |
| **ID** | **Summary** | **Priority** | **Status** | **Created** |
| NFTPS1 | The system shall integrate with mapping APIs to provide location-based services such as gym locator. The integration should be seamless, with real-time data synchronization and minimal latency. | C | Complete | 09/10/2023 |