11/5/2021

Functional Specification

Balance Health

Name : Diarmuid Brennan
Student No : C00133947
Supervisor : Joseph Kehoe

Abstract

This purpose of this document is to demonstrate what are the functionalities of the Balance Health project. The document will provide a detailed description of how the project will work, who the projects target audience is and illustrate a use case model for each of the user's interactions with the application to describe the functional operations of the project.

The document will also describe the non-functional requirements of the project using the FURPS+ model. This will also provide the metrics that will used to measure the success of the project. Finally, it will describe the testing procedures carried out to verify and validate the project.

Table of Contents

Table of Figures

Introduction

The purpose of this project is to create a mobile and web application that can be used by medical personnel to monitor the balance performance of their patients remotely from their web application. The balance activities to be carried out will follow the 4-Stage-Balance test, a tool developed by Physiopedia to assess a patient's mobility and risk of falls through monitoring the patients balance performance while carrying out the activities.

A medical personal will to set activities for the patient to carry out on the web application. The patient will carry out the activities while wearing a Movesense sensor device and the data will be transmitted to the mobile application on the patient's mobile device. The data gathered will be stored to the cloud database from which the medical personal can monitor and evaluate the patient's performance.

This document will describe who the users of the application will be and outline the functionalities and non-functionalities of the project. The functionalities will be communicated using a use case model. The main interactions between the application and the actors will be displayed in a use case diagram while use case stories and detailed use cases will be used to describe each of the functionalities in greater detail. The non-functionalities will also be outlined using the FURPS+ model and the document will provide a metric for the usability, reliability, performance, and security of the application.

Project Description

The project will consist of four main components, a web application, and a mobile application, a cloud database and a Movesense sensor device. The web application will be used by medical personnel to set activities and monitor and analyse a patient's performance. The mobile application will be used by the patient in conjunction with the Movesense device to gather sensor data while the patient performs a balance activity. The web and mobile application will then interact using a cloud database.

The web application can be accessed by medical personnel from which balance activities can be set and monitored for a patient. The web application will allow the medical personal to perform CRUD operations for patients and allow them to monitor and analyse the patient's performance while performing the balance exercises.

The patient will be able to download the activities to their mobile application and will carry out the activities while wearing a Movesense sensor device. The patient will perform the balance activities set by the medical staff and will also be able to track their own performance overtime. The data gathered during an activity will be stored to the database once the balance test has been performed.

The medical personnel will then be able to retrieve the patient's data for each of the activities carried out. They will use the data gathered to remotely monitor and analyse the patient's balance allowing them to view any improvements or deterioration in the patient's performance.

Users

This Balance Health project will be developed to target two main groups of users. The web application will be developed to target medical personnel, which can include, doctors, physiotherapists, or any medical staff that wishes to track and monitor a patient's balance performance. The mobile application is aimed at the patients. This can include elderly individuals and patients suffering or recovering from medical conditions where monitoring the patients balance can be of assistance in determining any deterioration in the patient's balance performance overtime.

Medical Personnel

Medical personnel will be able to register with and log onto the project's web application. They will be able to perform CRUD operations that allows them to manage and create patients on the application. They 4-Stage-Balance test activities will be set for each patient created and the patient will download the activities to their mobile application.

Once a patient carries out an activity their results are uploaded to the database. The data gathered can be then accessed by a medical person from the web application for analysis. The data will be displayed in graph format which can be analysed to determine improvements or deterioration in the patient's performance whilst carrying out each activity. This will provide medical personnel with data that can be used as a preventive measure to determine early diagnosis for risk of falls in the patient. Further recommendations can then be made to the patient based on the analysis carried out.

Patient

The patients will be the target users of the mobile application. Patients can range from elderly people to individuals suffering from various medical conditions who are either going though rehabilitation or are being monitored for deterioration in balance performance. Patients will be able to register an account and log on to the mobile application, entering their email and password.

Using the patients email address the application will download activities set by a medical personnel. The patient will carry out the activities while wearing a Movesense sensor device. The data will be gathered on the sensor device from which it is sent to the patient's mobile device. The mobile application will then send to the database and the data can be analysed and monitored by the medical personal. The patient will also be able to view their progress on the mobile application.

User Examples

Medical Personal

Anne is a practising doctor with over 20 years' experience. During that time, she has worked with many patients who have suffered from various medical conditions. In recent times Anne has seen a huge increase in the number of patients visiting the practice, especially elderly patients looking for health care. As the number of patients increases, the less time Anne must spend with individual cases which affects her ability to monitor and analyse a patient's condition accurately.

In Annes 20 years of experience she has realised how deterioration or improvements in a patient's balance can be used as a valuable metric in helping to gauge a patient's current condition or to help in monitoring deterioration in the patient's condition and predicting future issues. Due to the overburden placed on the practice is not possible to carry out a detailed analysis of a patient's balance performance.

The Balance Health application provides a method that will allow Anne to gather vast amounts of a patient's balance data directly from her own computer. This will reduce the number of patients visiting the practice and can help to act as a preventive measure for future conditions.

When a patient meets the criteria where monitoring of their balance will be of benefit, Anne can log on to the web application a create a patient account, entering the patient's condition and personal details. She can then set activities for the patient to carry out and upload them to the database. Anne will be able select what activities she wants the patient to carry out from an existing list or create a new activity. Once set up she will inform the patient and the patient will purchase/be supplied with a Movesense device and told to set up an account on the mobile application.

Once a patient begins carrying out the activities, Anne will be able to access the patient's data from the database and monitor their performance. The data will be displayed as part of graph from which Anne can analyse the data and determine what steps, if any, need to be taken. As the patient's progress improves or deteriorates, Anne can set new targets or activities for the patient or request a meeting in person.

The Balance Health provides a method that allows Anne to quickly monitor and analyse a patient's condition without needing to meet directly with the patient. This greatly reduces some of burden place upon Annes practice and allows more time to be spend with other patients.

Patient

John is a 60-year-old male who has recently suffered from stroke. As part of his recovery, he has been visiting Anne on a regular basis. John lives out the country which is over an hour away from Anne's practice where he often spends two hours travel and long waiting times for what often can be very quick sessions with Anne. Sometimes he would miss meetings with Anne and would have to arrange again for another session which could often be a week later.

On his last visit, happy with how John's recovery was progressing, Anne suggested the Balance Health application. This was a form of remote monitoring that could be carried out using a Movesense sensor device and a mobile application that would allow Anne to monitor John's condition from the practice, while John carried out a series of balance activities. In doing this, it would dramatically reduce the number of visits John would make to the practice and save on a lot of expense. John was happy to agree to try it out.

On his last visit to Anne, she set up his account on the web application and set several activities for John to carry out. Anne showed John the Movesense device and the mobile application that would be needed to carry out the activities and explained what exactly he had to do.

John created an account on the mobile application entering his personal details and email. On entering the application, he first selected the Movesense device to connect to the sensor. He was then displayed the list of balance activities set by Anne. Each activity contained a description of the activity and how to carry it out. Wearing the Movesense device strapped to his body, John selected the first activity to carry out. A countdown began and John began carrying out the activity and the data was transferred to the mobile device. After 30 seconds the beeper sounded to alert the end of the activity.

John could carry each of the activities set and view how is performance was going in the last number of attempts. John found this to be of huge benefit as it reduced the need for him to travel and spend time in a waiting room instead carrying out a few simple activities from which his condition could be monitored by his doctor.

Main Functionalities

This project will be separated into two separate components, a web application and a mobile application that will interact through a backend cloud database. The web application is aimed at a medical personal and will allow them to create patients and set activities for each patient to carry out. They will then be able to access and analyse the patient's data gathered from the Movesense device.

The mobile application is aimed at the patients and will allow the patient to access balance activities set by medical personnel from their mobile application. The application will connect to the Movesense sensor and gather data from the device while the patient carries out a balance activity. The data gathered will be uploaded to the database for analysis by medical personnel. The application will also allow the patient to monitor and track their own current progress. The main functionality of each application is listed below.

Web Application

- Register
- Log In
- CRUD Patient
- CRUD Activity
- Set a patient's activities
- Remove a patient's activity
- View a patient's performance
- Log out

Mobile Application

- Register
- Log In
- Connect/Disconnect to/from the Movesense device
- Display activities
- Perform exercise
- Upload Results
- View progress
- Log out

Web Application

Use Case Diagram

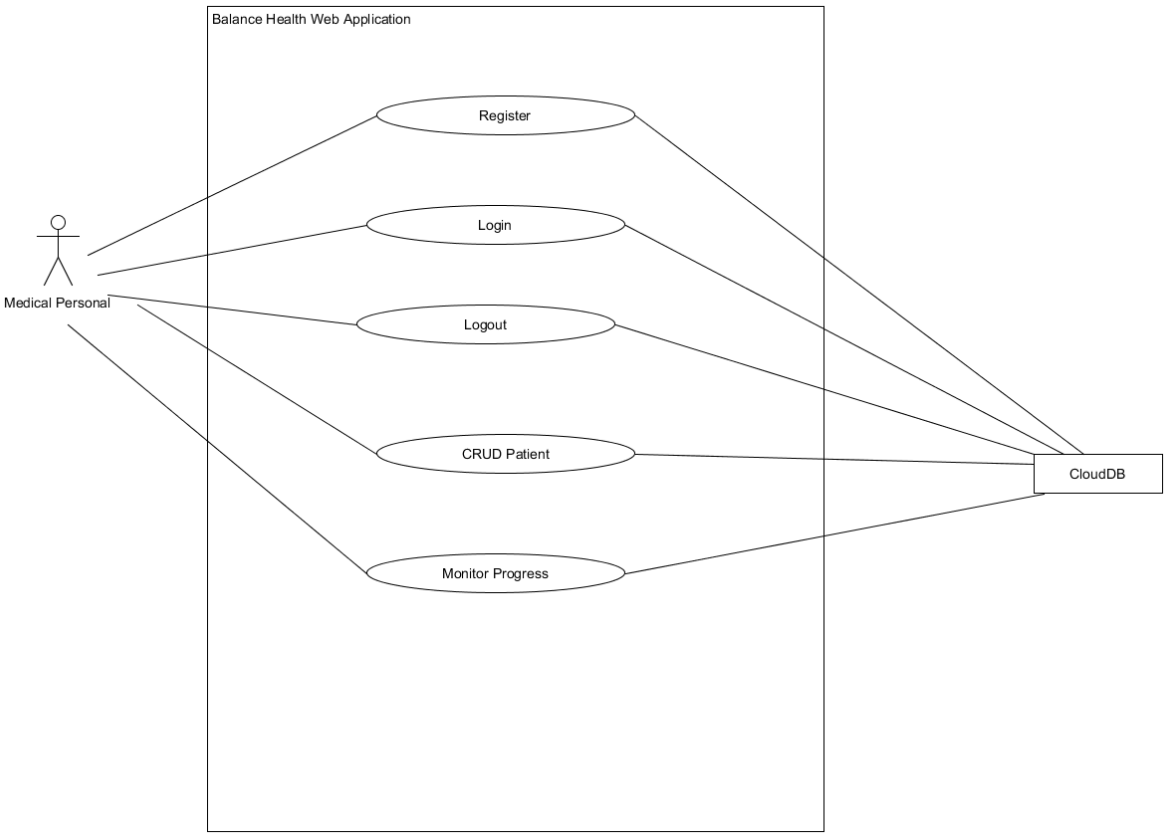

Figure 1 Web Application Use Case Diagram

Use Case Stories

Register

The register use case begins when the medical personnel start the application and is asked to log in or register a new account. The user selects register and is brought to the register page. They are then prompted to enter their details. On selecting submit, the details will be validated and if there is no existing user with the entered email the medical personals details are registered, and an account is created. The user is returned to the log in page.

Name	Register
Actors	Medical Personnel, CloudDB
Postconditions	The user's details are registered, and a new account is created
Primary Path	1. User opens application and is prompted to login or register 2. The user selects register 3. User is prompted to enter their details 4. User enters details and submits the entry 5. The input is validated, and an account is created 6. The user is directed to the log in page
Preconditions	The user has not previously registered an account.
Alternate Path	5a. User email details already exist in the database 1. User is asked to enter a different email address 2. The User details are validated, and an account is created

Login

The login use case begins when the user starts the application or has registered an account. The user will be prompted to enter their username and password. The user enters their details, and their details are validated against the data stored on the database. On validation success the user will be passed to the main screen

Name	**Login**
Actors	Medical Personnel, CloudDB
Postconditions	The user is authenticated and directed to the main screen
Primary Path	1. User opens application and is prompted to login or register 2. The user selects login 3. User is prompted to enter email and password 4. User enters details and submits the entry 5. The input is validated on the backend 6. The user is directed to the main page
Preconditions	The user has successfully registered on the application
Alternate Path	5a. User credentials do not match the database and must re-enter data 1. The user renters their details and the details are validated 2. The user is directed to the main screen

Logout

The logout use case begins when a user selects logout option from the application. The user is successfully logged out and returned to the login page of the application.

Name	**Logout**
Actors	Medical Personnel, CloudDB
Postconditions	The user has successfully logged out from the application
Primary Path	1. User selects logout option 2. The user is logged out and redirected to the login page
Preconditions	The user has successfully logged into the application
Alternate Path	N/A

CRUD Patient

The CRUD Patient use case begins when the medical personnel wishes to Create, View, Edit or Delete a new patient record. The user selects what action they would like to perform from a dropdown list. Create patient allows the user to enter a patient's details to a from and upload to the database. Edit allows the user to edit a patient's details, delete removes a patient's details from the records and view allows the user to view a patient's details. The desired patient can be selected from a list of existing patients displayed.

Name	**CRUD Patient**
Actors	Medical Personnel, CloudDB
Postconditions	1. The new patient was successfully created 2. An existing patient was viewed 3. An existing patient was successfully edited 4. An existing patient was successfully deleted
Primary Path	Create Patient 1. The user selects create patient from a dropdown list 2. The create patient form is displayed to the user 3. The user enters the patient details and selects submit 4. The data is validated, and the patient's details are stored to the database View Patient 1. The user selects view patient 2. A list of all the user's patients is displayed to the user 3. The user selects a patient and the patient's details are displayed Edit Patient 1. The user selects edit patient 2. A list of all the user's patients is displayed 3. The user selects a patient, and the patient's details are displayed 4. The user can edit the user details displayed in a form 5. The user submits the data, and the edited details are uploaded to the database Delete Patient 1. The user selects delete patient 2. A list of all the user's patients is displayed 3. The user selects the patient to be deleted 4. A pop-up displays asking to confirm the delete 5. The user selects yes, and the patient is deleted and removed from the database
Preconditions	The medical personal must have an account and be successfully logged in
Alternate Path	Create/Edit patient 3a. The user enters an invalid email 1. The user is prompted to enter a valid email address

Monitor Progress

The use case begins when the medical personal wishes to monitor a patient's performance when carrying a particular activity. The medical personnel select the patient from a list of patients. The patient's details are displayed along with the activities the patient is currently performing. The medical personnel select an activity and selects view performance. The patient's performance for the selected activity is retrieved for the database and displayed on a graph to the medical personal.

Name	**Monitor Progress**
Actors	Medical Personnel, CloudDB
Postconditions	The patient's activities data is retrieved from the database and displayed on a graph for the user
Primary Path	1. The user selects a patient from the list of patients 2. The patient's details ae displayed as well as the activities that they are currently performing 3. The user selects an activity and selects view performance 4. The activities data is retrieved from the database 5. The data is displayed to a graph for the user to analyse
Preconditions	The patient has carried out the balance activity and stored the data to the database
Alternate Path	4a. There is no data for the exercise stored on the database 1. The application displays a message stating there is currently no data store for that activity 2. The user selects a different activity

Mobile Application

Use Case Diagram

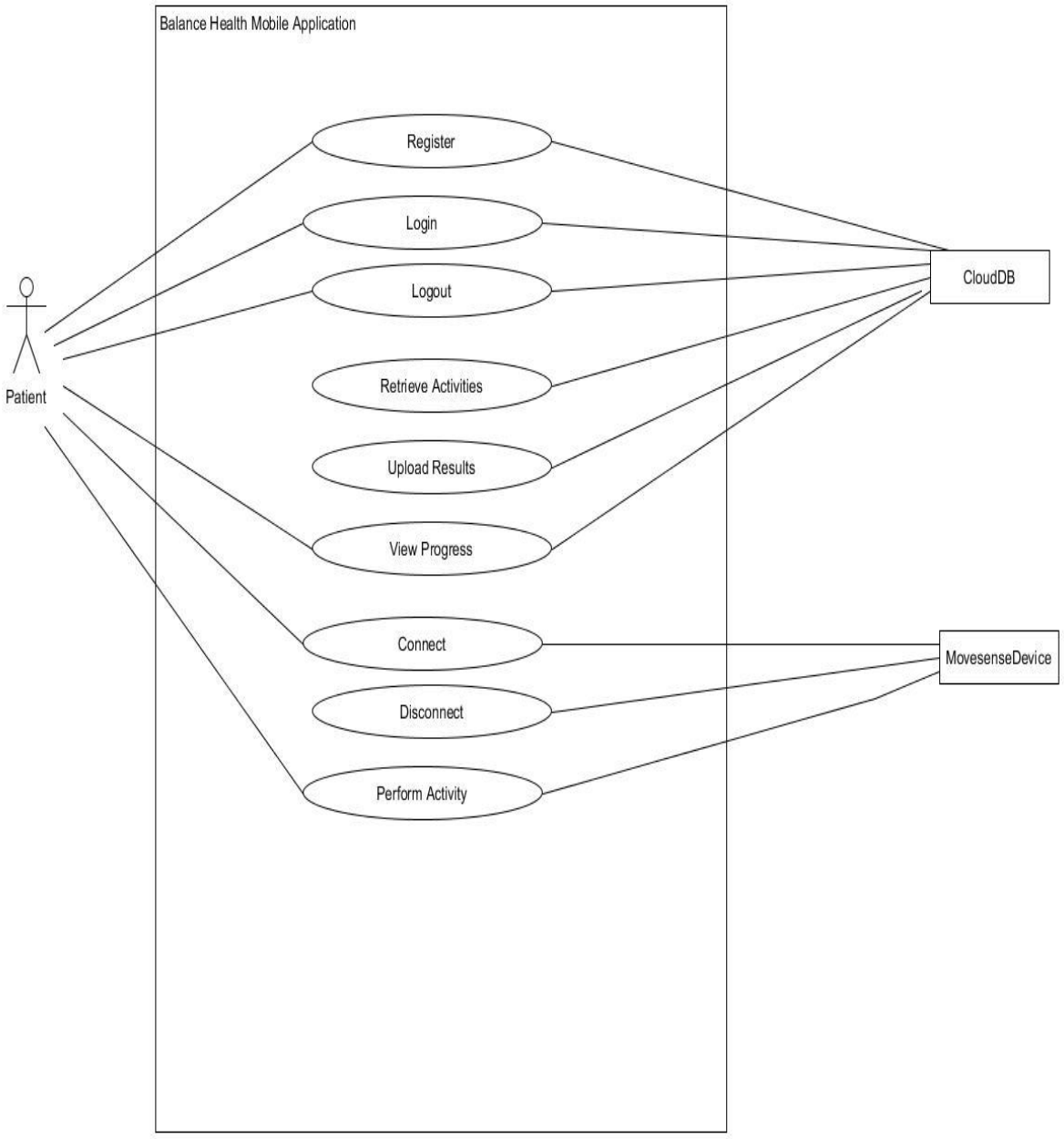

Figure 2 Mobile Application Use Case Diagram

Use Case Stories

Register

The register use case begins when the user starts the application and is asked to log in or register a new account. The user selects register and is brought to the register page. They are then prompted to enter their details. On selecting submit, the details will be validated and if there is no existing user with the entered email the patient details are registered, and an account is created. The user is returned to the log in page.

Name	**Register**
Actors	Patient, CloudDB
Postconditions	The user's details are registered, and a new account is created
Primary Path	1. User opens application and is prompted to login or register 2. The user selects register 3. User is prompted to enter their details 4. User enters details and submits the entry 5. The input is validated, and an account is created 6. The user is directed to the log in page
Preconditions	The user has not previously registered an account
Alternate Path	5a. User email details already exist in the database 1. User is asked to enter a different email address 2. The User details are validated, and an account is created

Login

The login use case begins when the user starts the application. The user will be prompted to enter their username and password. The user enters their details, and their details are validated against the data stored on the database. On validation success the user will be passed to the main screen.

Name	**Login**
Actors	Patient, CloudDB
Postconditions	The user is authenticated and directed to the main screen
Primary Path	1. User opens application and is prompted to login or register 2. The user selects login 3. User is prompted to enter email and password 4. User enters details and submits the entry 5. The input is validated on the backend 6. The user is directed to the main page
Preconditions	The user has successfully registered on the application
Alternate Path	5a. User credentials do not match the database and must re-enter data 1. The user renters their details and the details are validated

Logout

The logout use case begins when a user selects logout option from the application. The user is successfully logged out and returned to the login page of the application.

Name	**Logout**
Actors	Patient, CloudDB
Postconditions	The user has successfully logged out from the application
Primary Path	1. User selects logout option 2. The user is logged out and redirected to the login page
Preconditions	The user has successfully logged into the application
Alternate Path	N/A

Connect

The user has successfully logged into the application and has been directed to the main page. The application automatically begins searching for nearby Movesense devices over Bluetooth Low Energy (BLE). Any devices found are displayed on the screen for the user to see. The user selects the Movesense device, and the application connects to the device using BLE. Once successfully connected to the device, a message is displayed saying that the device is successfully connected, and a button is enabled allowing the user to access their balance activities.

Name	**Connect**
Actors	Patient, MovesenseDevice
Postconditions	The application has successfully connected to the device over BLE
Primary Path	1. User logs in and the application attempts to connect to the sensor over BLE 2. The application requests mobile device location permission 3. The user allows app to access device location 4. The application searches for nearby devices using Bluetooth 5. The app displays found nearby devices 6. The user selects the required device 7. The app connects to the device
Preconditions	The mobile device has its Bluetooth turned on and the user has logged in
Alternate Path	4a. The device does not find any nearby devices 1. The user must be near the device and restart the application 7a. The app fails to connect with the device 1. The app repeats attempting connection to the device every 5 seconds until connection is successful

Disconnect

Once the user has exited the application, the application disconnects its connection from the device.

Name	**Disconnect**
Actors	Patient, MovesenseDevice
Postconditions	The application has successfully disconnected from the MovesenseDevice
Primary Path	1. User exits the application 2. On exiting the application disconnects its BLE connection with the MovesenseDevice
Preconditions	The mobile device has successfully connected with the MovesenseDevice over BLE
Alternate Path	N/A

Display Activities

The application has successfully connected with the Movesense device, and the user has selected to view activities. The application retrieves the users balance activities from the database set by the medical personal. Each activity name is then displayed on a list. The user can then select an activity to view its details are carry out the activity.

Name	**Display Activities**
Actors	Patient, CloudDB
Postconditions	The activities set for the patient are retrieved and the displayed as a list
Primary Path	1. User selects view activities 2. Using the user's details, the application retrieves the user's activities from the database 3. Each activity is displayed as part of a list for the user to choose from 4. The user selects an activity and is shown a description of the activity to carry out 5. The user can then continue to carry out an activity
Preconditions	A user is successfully logged in and has activities set for them by a medical personal
Alternate Path	2a. The user has no activities set to carry out 1. The application displays a message telling the user that no activities have been currently set for the user

Perform Activity

The user has selected a balance activity. The user is brought to the perform activity page and selects start activity. A countdown begins and user performs the activity described for a set amount of time while wearing the Movesense device. The app requests balance data from the sensor for the duration of the activity, displaying the data to a graph. When the time has elapsed, the app unsubscribes from the request to the device and stores the data.

Name	**Perform Activity**
Actors	Patient, MovesenseDevice
Postconditions	The application has successfully requested data from the device, displayed it on a graph and stored it to the application
Primary Path	1. A balance activity has been selected 2. A countdown begins, letting the user know when to begin carrying out the activity 3. The user carries out the activity while wearing the Movesense device 4. The application requests the data form the sensor for the duration of the activity 5. The data contained in the response is displayed on the graph on the application 6. A buzzer informs the user the activity has completed 7. The app unsubscribes from the device and stores the results
Preconditions	A connection has been established between the application and the Movesense device over BLE.
Alternate Path	5a. The data is not returned through the request 1. An error message is displayed to the user informing that there has been an error in the request to the sensor device

Upload Results

The user has completed a balance activity. The user's movements while carrying out the activity are calculated. The application uploads the calculated results, balance data, and user details to be stored on the database.

Name	**Upload Results**
Actors	CloudDB
Postconditions	The balance activity data has been successfully uploaded and stored on the database
Primary Path	1. A balance activity has completed 2. The user's movements while carrying out the activity are calculated 3. The application uploads the data to the database, including the user's details 4. The app returns to the previous screen
Preconditions	The user has successfully carried out a balance activity.
Alternate Path	3a. The application fails to upload to the database 1. On failure, the application attempts to reconnect to the database and upload until upload is successful

View Progress

The user selects view progress button for a balance activity. The app requests the user's previous weeks calculated scores from the database. The app displays the results on a graph showing the users progression over time.

Name	**View Progress**
Actors	Patient, CloudDB
Postconditions	The app has successfully requested the user's balance scores data from the CloudDB and displayed the results on a graph.
Primary Path	1. A user selects view progress 2. The app requests the users balance result data form the database 3. The view progress screen displays the user progress from the preceding weeks results when performing the selected activity 4. The data is displayed on graph showing the user's balance progress
Preconditions	A connection has been established between the application and the CloudDB
Alternate Path	2a. The application fails to connect to the database 1. The application displays an error message to the user stating that request was unsuccessful

FURPS+

The requirements of a software development project can generally be defined under two main headings, functional and non-functional requirements (Wang, 2011). This classification of the requirements allows for different aspects of the requirements to be separated out and viewed independently.

A use case model is used to allow us to visualise the interactions between the user and application. It is an important tool that allows to clearly communicate to any stakeholders involved what are the functional requirements of the project.

There are however many other non-functionalities of the application that are not simple interactions between the user and the application, but which are vital to the efficient running of the application. The acronym FURPS+ is used to illustrate the main non-functionalities of a software project that should be addressed when developing a software project and can be described under the following headings:

- **F**unctionality
- **U**sability
- **R**eliability
- **P**erformance
- **S**upportability
- **+** Security, Implementation, and Interface constraints

"This FURPS+ technique made the requirements classification to stress on understanding the different types of non-functional requirements more" (COEPD, 2021). FURPS+ is widely used throughout the software industry as a method to outline the non-functional requirements that are common to most software projects. It provides a way for quantifying requirements which can be otherwise difficult to measure. By quantifying the non-functional requirements, it will provide us with a metric for determining how well the application is performing.

Functionality

The main functional requirements of the application are best illustrated using a use case model. The functional requirements described in the FURPS+ model do not refer to the functionalities described in the use case model. Here, the functional requirements that cannot be easily communicated using a use case model are detailed. This can include supplementary functionalities of the application that are used to support the use cases.

MDS Library

The MDS library provides the functionality to communicate with the Movesense device using the inbuilt Movesense REST API Whiteboard. The library allows us to connect with the sensor device and subscribe to a sensor. The sensor returns its response as a JSON string which can be mapped to an object class in the application.

Usability

Usability refers to the users experience while interacting with and utilizing the product. The usability requirements involve looking at, capturing, and recording the requirements of the application regarding the user interface. The usability requirements for the mobile and web application are shown below.

Web application

- The user interface design should be both engaging and easy to use.
- The app should be easy to follow, and the user should be able to navigate between the applications pages within 10 seconds.
- The user should be able to register a new account within 20 seconds.
- The user should be able to login within 5 seconds.
- The app should upload data to the database within 5 seconds, 90% of the time.
- The app should retrieve data from the database within 5 seconds, 90% of the time.

Mobile Application

- The user interface design should be both engaging and easy to use.
- The app should be easy to follow, and the user should be able to navigate between the applications activity pages within 10 seconds.
- The app should connect to the Movesense device over BLE within 30 seconds 90% of the time when within range.
- The user should be able to login within 5 seconds.
- The app should upload to the database within 5 seconds, 90% of the time.
- Data gathered from the sensor device should be displayed to the screen within 2 seconds, 90% of the time.
- The app should retrieve data from the database within 5 seconds, 90% of the time.
- Documentation should be provided to allow new users carry out the exercises within the first minute of initial use.

Reliability

Reliability comprises of requirements that relate to the “availability, accuracy, and recoverability” (COETL, 2016) of the application. This provides a measure for the acceptable frequency of failure, the extent and time length between failures, and the time it takes to recover.

Web application

- The app should recover from failures 95% of the time.
- Data should be recovered 99% of the time.
- The app should open without issue 99% of the time.

Mobile application

- The app should recover from failures 95% of the time.
- The app should open without issue 99.99% of the time.
- The accuracy of the data gathered from the sensor should be within 0.1 of the actual movement, accounting for noise.

Performance

Performance details the requirements for performance related issues such as response times, throughput, and recovery times. The metrics for analysing the performance of the application include.

Web application

- Start-up time should take less than 2 seconds 98% of the time.
- Requests to the database should respond within 5 seconds 95% of the time.
- Connection loss to the database should be recovered within 5 seconds 99% of the time.

Mobile Application

- Start-up time should take less than 5 seconds 95% of the time.
- The Movesense device response time to requests from the app should take milliseconds 98% of the time.
- On connection loss from the Movesense device the app should reconnect with the device within 20 seconds 90% of the time.
- Requests to the database should respond within 5 seconds 95% of the time.
- Connection loss to the database should be recovered within 5 seconds 99% of the time.

Supportability

Supportability refers to the ease at which the application software can be modified to accommodate enhancements and changes. “Supportability refers to the flexibility, maintainability, testability, instability, scalability, and compatibility of the system” (COETL, 2016). The applications should:

Web Application

- The app should be available on 90% of web browsers
- The app software should be easily maintained using recognised software patterns, documentation and comments describing the functionality of algorithms.
- The project should be able to scale easily to allow for many users to use the application.

Mobile Application

- The app should be compatible across 95% of android devices.
- The app software should be easily maintained using recognised software patterns, documentation and comments describing the functionality of code snippets.
- The project should be able to scale easily to allow for a large number of users to use the application.

+ Security, Implementation and Interface constraints

The '+' in the FURPS+ model describes constraints of the application which can include security, design, interface, or physical constraints.

Security

- Authentication must be provided to access personal data from the application.
- The user's login information will be stored on the database. Any user sensitive information should be encrypted.

Implementation constraints

- The application should be implemented to standards set out for mobile applications developed using Android.

Interface constraints

- The application will have two external interfaces, to connect with the Movesense device over BLE and to connect to the cloud database from the mobile device and to connect with the database from the web application. The remaining components of each of the applications will be self-contained.

Success Metrics

Success metrics will be used to evaluate a measure of the applications success. The following criteria will be used to outline the metrics from which the success of the project can be measured.

Medical Personnel

- The user should be able to register their details on the application.
- The user's details should be encrypted and securely stored using Firestore authentication procedures.
- The user should be able to log in to their account and perform CRUD operations for patients and balance activities.
- The user should be able to set activities for a patient.
- The user should be able to retrieve a patient's balance data related to an activity in order to monitor and analyse the patient's progress.

Patient

- The mobile application was developed using Android 5.0 (Lollipop) which should run successfully on 98% of android devices.
- The user should be able to register their details on the application.
- The user's details should be encrypted and securely stored using Firestore authentication procedures.
- The user should be able to connect successfully to the Movesense device 98% of the time.
- The user should be able to log in to their account and access the balance activities set by the medical personal.
- The user should be able carry put an activity while the application successfully gathers sensor data from the Movesense device.
- The data should be recorded and accurately stored to the database.
- The user should be able retrieve their previous weeks performance from the database and view their progress.

Testing

Software testing will be applied to the mobile and web applications to test the performance of the functional and non-functional requirements illustrated in this document. Various testing techniques will be applied to the application to test the interfaces and functionalities provided. “Software testing is the act of examining the artifacts and the behaviour of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation” (Wikipedia, 2021).

Black box, White box and User testing techniques will be carried out on the applications to ensure that the code is verified and validated and that no issues or bugs are found within the code. User testing will be applied to the user interface of the applications from a user’s perspective. This will be used to help determine that the UI is behaving as expected and will give an indication of how well the non-functional requirements of the application are performing. Black box and White box testing will be used to test the inner functionalities of the code. This will help to indicate that the code is performing as expected and as further changes are added that the existing code is not broken.

User testing of the application will be caried out in conjunction with the health science department at the college. Feedback will be gathered on the applications functionality, usability, reliability, and performance to help establish how to improve upon the overall functionality of the application. Unit tests will be used to carry out white box testing on the each of the methods provided in the application. As changes are made to the code, unit tests will be used to inform whether the changes have broken or caused a bug in the code.

Conclusion

The balance health application is an application that aims to allow a medical personal to set activities and to monitor and analyse a patient's performance remotely from the web application provided. The patient will be able to carry out the activities from the on the mobile application. This document describes the intended functionalities and non-functionalities of the project.

The functionalities of the project were illustrated using a use case model to demonstrate the user interactions when using the web and mobile applications. The document also describes the non-functional requirements of each of the applications. Metrics for each of the non-functional requirements and the overall success of the project are outlined and will be used as a barometer at the conclusion of the project to determine how successful the project is.

This document has described the functional and non-functional requirements of the application as well as providing metrics which will be used as a measurement in determining the success of the project. The testing procedures that will be carried out throughout the project were also discussed.

Bibliography

COEPD, 2014. What is FURPS+?
Available at: What is FURPS+? – Business Analyst Training in Hyderabad – COEPD (wordpress.com)
[Accessed 05 January 2022].

Gangadar, K., 2016. What is FURPS+?
Available at: What is FURPS+? - COETL Business Analyst Training in Hyderabad
[Accessed 18 November 2021].

Wang, S-H., Samadhiya, D., Chen, D., 2011. Software Quality: Role and Value of Quality Models, International Journal of Advancements in Computing Technology, Vol. 3, No. 6, pp. 65 – 74.

Wikipedia, 2021. Software Testing
Available at: Software testing - Wikipedia [Accessed 06 January 2022].