Functional specification

[Abstract 3](#_Toc498518821)

[Use Case Diagram 4](#_Toc498518822)

[Brief use case 5](#_Toc498518823)

[Patient Login 5](#_Toc498518824)

[Patient Logout 5](#_Toc498518825)

[Register 6](#_Toc498518826)

[Manage Account 6](#_Toc498518827)

[Process Account Creation 7](#_Toc498518828)

[Process Login 7](#_Toc498518829)

[Doctor Login 8](#_Toc498518830)

[Doctor Logout 8](#_Toc498518831)

[Manage Patients 9](#_Toc498518832)

[Food Diary 9](#_Toc498518833)

[Process Food Diary 10](#_Toc498518834)

[Process Manage Patients 10](#_Toc498518835)

[Process Motion 11](#_Toc498518836)

[Process Spirometer 11](#_Toc498518837)

[Supplementary Specification 12](#_Toc498518838)

[Functionality 12](#_Toc498518839)

[Useability 12](#_Toc498518840)

[Reliability 12](#_Toc498518841)

[Performance 12](#_Toc498518842)

[Supportability 12](#_Toc498518843)

[Iteration Plan 13](#_Toc498518844)

[Iteration 1 13](#_Toc498518845)

[Iteration 2 13](#_Toc498518846)

[Iteration 3 13](#_Toc498518847)

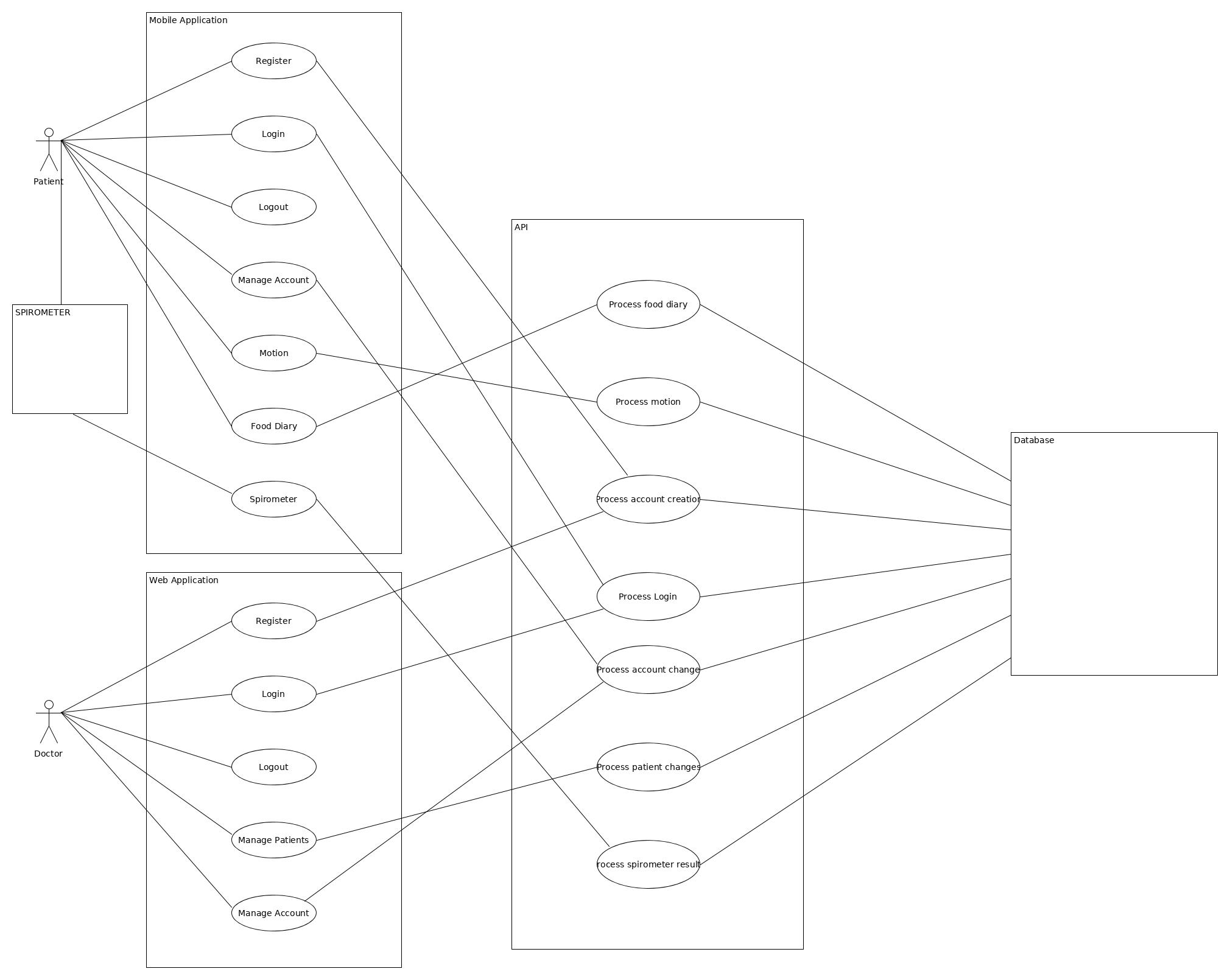
[Conclusion 14](#_Toc498518848)

## Abstract

The point of this document is to give the reader the overall feeling of the application. The document provides a view of what the application is going to do and how it will work internally.

## 

## Use Case Diagram



## Brief use case

### Patient Login

|  |  |
| --- | --- |
| Name | Patient Login |
| Actors | Patient,Mobile App, API |
| Preconditions | The patient has downloaded the CFAPP |
| Activity | Begins when the patient opens the mobile app. The system listens for the input of patients credentials(username.password). The credentials are sent to the API for validation purposes. |
| Consequences | The API performs validation on the patient's login credentials that were entered |

### Patient Logout

|  |  |
| --- | --- |
| Name | Patient Logout |
| Actors | Patient, Mobile App |
| Preconditions | The patient has successfully logged into the system. |
| Activity | Begins when a patient has finished with the application. The patient selects the logout button then they are prompted if they really want to logout and confirm that they do. |
| Consequences | The patient is logged out. |

### Register

|  |  |
| --- | --- |
| Name | Register |
| Actors | Patient/Doctor, Mobile/Web App, API,Database |
| Preconditions | The mobile app has successfully loaded. |
| Activity | Begins when a patient/doctor opens the mobile app. The patient/doctor clicks the register button and a new page appears. The patient/doctor will be prompted either to sign in with their google, facebook or setup one using there own email account. The account creation request is sent to the API for processing. |
| Consequences | The API processes the patient/doctor register request. |

### Manage Account

|  |  |
| --- | --- |
| Name | Manage Account |
| Actors | Patient, Doctor, Mobile App,Web App API,Database |
| Preconditions | The patient and doctor have successfully logged into the mobile app. |
| Activity | Begins when a Patient/Doctor navigates to there profile page in the mobile app. Patient/Doctor are presented with the option to change their basic information is sent to the API. The API then validates the request. |
| Consequences | Once validation takes place the API makes the changes to the profile of the Doctor/Patient and saves it to the database |

### 

### Process Account Creation

|  |  |
| --- | --- |
| Name | Process Account Creation |
| Actors | Patient, Doctor, Mobile App,Web App API,Database |
| Preconditions | The Doctor/Patient has successfully completed the registration form to create a profile. |
| Activity | Begins when a Doctor/Patient clicks the submit button and the request to register and create a profile is sent to the API. The API then validates the request. |
| Consequences | Once validation takes place the API creates Doctor/Patient and stores it in the database. The Doctor/Patient are redirected to login on their respective apps. |

### Process Login

|  |  |
| --- | --- |
| Name | Process Login |
| Actors | Patient, Doctor, Mobile App,Web App API,Database |
| Preconditions | The Doctor/Patient have created a profile and logged in. |
| Activity | The Doctor/Patient has previously created a profile. The Doctor/Patient has entered their login credentials and pressed login. |
| Consequences | Once a match in credentials takes place the Doctor/Patient is granted access to their respective homepages. |

### Doctor Login

|  |  |
| --- | --- |
| Name | Doctor Login |
| Actors | Doctor,Web App,API |
| Preconditions | The web app has successfully loaded and the initial login screen successfully presented itself. The doctor has already signed up for an account. |
| Activity | Begins when the doctor starts the web app in their browser. The web app listens for the input of the doctor's credentials. The credentials are sent to the API for validation |
| Consequences | Once validation has taken place. The homepage of the web app is displayed to the doctor. |

### Doctor Logout

|  |  |
| --- | --- |
| Name | Doctor Logout |
| Actors | Doctor,Web App |
| Preconditions | The doctor has successfully logged into the web app. |
| Activity | Begins when the doctor has completed all interactions with the web app. The doctor selects logout button and is asked to confirm selection. The doctor is then logged out. |
| Consequences | The doctor is redirected to the login screen of the web app. |

### Manage Patients

|  |  |
| --- | --- |
| Name | Manage Patients |
| Actors | Doctor,Web App, API |
| Preconditions | The doctor has already logged into the web app |
| Activity | Begins when a doctor clicks a patient tab on the web app. The website then loads the patient screen and the doctor can view and manage their patients. |
| Consequences | The changes/notes are sent to the API for processing. |

### Food Diary

|  |  |
| --- | --- |
| Name | Food diary |
| Actors | Patient,Mobile App,API,Database |
| Preconditions | The patient has successfully logged on to the mobile app. |
| Activity | Begins when the patient clicks the food diary tab on their homepage. The mobile app displays the food diary screen and the patient can manage their food diary. This means that they can select meals, snacks, track creon doses and supplements. |
| Consequences | The changes are sent to the API for processing. |

### Process Food Diary

|  |  |
| --- | --- |
| Name | Process Food Diary |
| Actors | Patient, Mobile App, API,Database |
| Preconditions | The Patient has make changes to their food diary. |
| Activity | Begins when a Patient selects the update button on there food diary and the changes are sent to the API. The API then validates the request. |
| Consequences | Once validation takes place on the updated food diary it stores it in the database. The Patient is redirected to the food diary home page. |

### Process Manage Patients

|  |  |
| --- | --- |
| Name | Process Manage Patients |
| Actors | Doctor,Web App API,Database |
| Preconditions | The Doctor has successfully logged into the web app. |
| Activity | Begins when a Doctor clicks the manage patients tab and is brought to the manage patient screen where they can select a patient and view their data and make notes depending on how they are doing these notes are is sent to the API. The API then validates the request. |
| Consequences | Once validation takes place the API the notes are stored it in the database. The Doctor is redirected to the manage patient page. |

### Process Motion

|  |  |
| --- | --- |
| Name | Process Motion |
| Actors |  |
| Preconditions |  |
| Activity |  |
| Consequences |  |

### Process Spirometer

|  |  |
| --- | --- |
| Name | Process Spirometer |
| Actors |  |
| Preconditions |  |
| Activity |  |
| Consequences |  |

## Supplementary Specification

### Functionality

The mobile and web applications must have access to the internet to interact with the API and database to access the data.

### Usability

A patient should be able to create a profile in less than 5 minutes. Once logged on a patient shouldn't have to logout. The applications should be user friendly and while navigating through the application it should give the user the feeling that they know where everything is.

### Reliability

In the case of a failure in data retrieval or verification, the user interface should stay responsive. Also the application needs to be able to store motion and other data if a connection drops. It should then relay the information to the database once internet has been re-established.

### Performance

The user interface should load in a timely manner. This is to not make the user feel as if they are waiting for extended periods of time to load pages.

### Supportability

Support for multiple browsers must be added without negative impact on the mobile/web applications.

## Iteration Plan

### Iteration 1

### Iteration 2

### Iteration 3

## Conclusion