

Functional Requirements/Specification

ALL CARE-DOCTOR’S OFFICE

Ahsan habib

DECEMBER 30, 2024

Abstract

The purpose of this project is to create a web application to aid small to medium size doctor office. This document outlines and illustrates the planned internal functionality of the online healthcare doctor’s office - we named as **All Care - Doctor’s Office**. It details what features the application has and how each one aids a patient or doctor.

Table of Contents

[Abstract 1](#_Toc49703162)

[Table of Figures 4](#_Toc49703163)

[1. Introduction 5](#_Toc49703164)

[2. Application Overview 6](#_Toc49703165)

[2.1. Product Concept 6](#_Toc49703166)

[2.2. Target Users 7](#_Toc49703167)

[2.2.1. Patients 7](#_Toc49703168)

[2.2.2. Doctors 7](#_Toc49703169)

[2.3. Functionality & Features 8](#_Toc49703170)

[2.3.1. Progressive Web Application (PWA) 8](#_Toc49703171)

[2.3.2 Authentication (Patient & Doctor) 8](#_Toc49703172)

[2.3.3. Medical Record (Patient) 8](#_Toc49703173)

[2.3.4. Appointments (Patient & Doctor) 9](#_Toc49703174)

[2.3.5. Prescriptions (Patient & Doctor) 9](#_Toc49703175)

[2.3.6 Communication 9](#_Toc49703176)

[2.3.7. User Profile (Patient) 10](#_Toc49703177)

[2.3.8. Dark Mode (Patient) 10](#_Toc49703178)

[2.3.9 Track Medication (Patient) 10](#_Toc49703179)

[3. System Architecture 11](#_Toc49703180)

[4. Use Cases 12](#_Toc49703181)

[4.1. Full User Use Case Diagram 12](#_Toc49703182)

[4.2. Detailed Use Cases 13](#_Toc49703183)

[4.2.1. Register 13](#_Toc49703184)

[4.2.2. Login 13](#_Toc49703185)

[4.2.3. Change Password 14](#_Toc49703186)

[4.2.4. Logout 14](#_Toc49703187)

[4.2.5. View Medical Profile 14](#_Toc49703188)

[4.2.6. Update Medical Details 15](#_Toc49703189)

[4.2.7. View Appointment 15](#_Toc49703190)

[4.2.8. Add Appointment 16](#_Toc49703191)

[4.2.9. Receive Consultation 16](#_Toc49703192)

[4.2.10. Provide Consultation 16](#_Toc49703193)

[4.2.11. Provide Prescription 17](#_Toc49703194)

[4.2.12. View Prescription 17](#_Toc49703195)

[4.2.13. Accept/Deny Prescription 17](#_Toc49703196)

[5. Supplementary Specification (FURPS+) 18](#_Toc49703197)

[5.1. Functionality 18](#_Toc49703198)

[5.2. Usability 18](#_Toc49703199)

[5.3. Reliability 18](#_Toc49703200)

[5.4. Performance 19](#_Toc49703201)

[5.5. Supportability 19](#_Toc49703202)

[5.6. + (Security) 19](#_Toc49703203)

[6. Metrics 20](#_Toc49703204)

[6.1. Potential Risks & Issues 21](#_Toc49703205)

[7. Conclusion 22](#_Toc49703206)

[8. Plagiarism Declaration 23](#_Toc49703207)

[9. Bibliography 24](#_Toc49703208)

[9.1. References 24](#_Toc49703209)

# Table of Figures

[Figure 1: All Care - Doctor Office Web App Flow 10](#_Toc49551776)

[Figure 2: Full User Use Case Diagram 11](#_Toc49551777)

# 1. Introduction

The purpose of this document is to outline the functionality and requirements necessary for the development of the proposed online healthcare application.

A general overview of the application is given with regards to the product concept so that the project can be clearly understood. Any target users pertaining to the application are examined, their needs analyzed and any paint points they might have with an online based healthcare system explored. Once these aspects are explored the application features and functionality are outlined and explained, with regard to what the feature does, who it impacts and how it helps.

The document then depicts system architecture i.e. any interesting architectural components of the application.

Any users and their role within the application are depicted through use case diagrams. These diagrams illustrate the users’ interactions with the web-based application and the features it contains. A number of brief use cases break down the users’ interaction with the application into smaller levels of functionality. Supplementary specification is explored in relation to the applications development to determine requirements that may not have been identified by the use cases.

A metric system has been put in place and a number of goals set to determine how the projects development is going. This will ensure that each goal is being met and measure the projects’ overall success rate. A potential risk analysis identifies problematic areas or issues that might arise from developing the application in the manner and consider how best to handle these possibilities.

# 2. Application Overview

## 2.1. Product Concept

According to S. Ather, the Vice President of GainAm, in the future we should apply real world project to get real environment experiences - so we implementing this in-house project to build web-app for the doctor’s office.

Since I have experiences being a Business Analyst, being a Project Manager and being a Scrum Master - I would try to create business refinements/specification document - so that our developer could start building up web-app - utilizing Agile Scrum process.

Inspired by the current situation, All Care - Doctor’s Office aims to modernize the way in which medical services are delivered in small to medium sized clinics, specifically for those who are in receipt of a medical card. By providing users with access to a range of online health related services, including the ability to communicate with doctors, patients will be able to avail of medical services without their physical presence being required, provided their requirements are not of a vital nature. At the same time, doctors will only be exposed to patients who most absolutely require their services face-to-face.

The application itself will be developed as a web application in order to maximize its user base by making it accessible on a variety of devices and platforms. In addition to this, the application will be built as a single page application (SPA) so that each page is loaded dynamically rather than a browser load, maximizing the applications efficiency and providing a straight-forward user experience. The application will take inspiration from preexisting online health services whilst implementing features which are more appropriately associated with small to medium sized clinics.

## 2.2. Target Users

The online All Care - Doctor’s Office web application, has two main target markets to whom it must satisfy, patients and doctors, both of which have roles that are mutually exclusive from one another; though some of their functionality will cross over.

### 2.2.1. Patients

A patient is anyone who seeks to avail of a medical service or requires medical attention. Specially, regarding this application, a patient will be anyone who wishes to avail of a medical service through a more digital means, is above sixteen years of age and owns a medical card. They do not wish to directly contact the clinic in person or over the phone and prefer the ease of controlling their dealings through their own profile on an application. Where this application will differ from many that are currently available is that it has been designed for patients in possession of a medical card to cater to a niche in the current online healthcare market that does not appear to be fully met.

### 2.2.2. Doctors

A doctor is a licensed health professional in the medical sector, who provides care, or rather who’s role is to maintain and restore health to patients seeking support with medical queries, as well as ailments both physically and non-physically visible to the eye. Their job is to examine and determine ailments based on a patients’ medical history, family history or physical condition and prescribe an appropriate treatment with the hopes of resolving the patients’ problems. In small and medium size clinics alike, a doctor will often see a plethora of patients on a given day. Not only will having patients use an online application afford doctors the ability to limit the number of non-vital patients they must physically deal with in a day, it would also allow them to conduct their business in a more organized manner. Optionally, some doctors within a clinic could be solely dedicated to online queries, allowing these doctors to conduct their business online from their home rather than directly in the clinic.

## 2.3. Functionality & Features

Within this application, there are a number of core features and functionality that must be considered with development.

### 2.3.1. All Care - Doctor’s Office Web Application

In order to ensure that the application is fast and reliable, it must be developed based on object oriented programming language - I.e. JAVA or .NET - In doing so, it will provide an install-able, native app-like experience on desktop and mobile and work in any browser. For the application it requires a web app manifest and a service worker. The web app manifest is a JSON file that informs the browser that the application stores information such as the app name, any icons in use and the URL that opens when the app is launched. The service worker is a JavaScript file that will run separately from the main browser, intercepting network requests, caching or retrieving resources from cache and delivering push messages.

### 2.3.2 Authentication (Patient & Doctor)

As with any application, All Care - Doctor’s Office must have a solid foundation off of which it can grow. This application is of a private nature and thus it must be secure from public view and individual to each user on the application. In order to maintain this privacy, the application needs a registration and login system. A user can register with the application and is designated their own user credentials to which they can then use to log into the application with and be recognized as a separate entity from any and all other users. Naturally, they will also be able to log out and, as a fail-safe, request a password change if they have forgotten their log in credentials, by sending them an email to register a new password for their account.

As mentioned previously, this application will have two main user types, the patients who will have a role of patient and the doctors who will have a role of doctor. Both of these users will be using the application, the screens and functionality that they have will be different. Therefore, the application should register their roles in different ways. The view and access to screens will also need to be controlled so that a patient cannot see a doctors’ screens and doctor cannot see a patients’ screens.

When registering, the user will be required to enter information such as their name, date of birth, medical card/medical credentials, email, password and confirm password. They must accept any terms and conditions before they are allowed to complete registration.

When a user logs in they will be redirected to a dashboard page.

### 2.3.3. Medical Record (Patient)

Within the application, a patient will be able to store their medical records, so that they may easily access their medical history wherever they are. Information such as their gender, blood type, weight and height can be stored and updated whenever they so desire. They will also be able to store any allergies, conditions or immunizations that they have had. These can be deleted and re-added as they want.

### 2.3.4. Appointments (Patient & Doctor)

On the patients’ side of the application, a patient will be able to book an appointment. When booking the appointment, all of the patients’ personal details will be dynamically loaded into the form from their user profile. They will need to select whether the appointment is online or in a clinic. They will also need to select a doctor, date, time and include details about their reason for the appointment. When they have booked the appointment, they will be redirected to a screen where they can view all of their appointments, new and old. They will also be able to cancel the appointment, as long as it is not on the day of the actual appointment.

When an appointment is being booked there must be checks to ensure that patients cannot book on top of a time-slot that another patient has already taken. They should also not be allowed to book more than one appointment per day.

On the doctors’ side of the application, the doctor will be able to keep a schedule of all appointments, pertaining to them. When a patient books an appointment the appointment details will populate into a calendar, letting the doctor know which patient has booked, what type of appointment it is and also what time the appointment starts.

### 2.3.5. Prescriptions (Patient & Doctor)

Patients will be able to select a prescription to request, depending on their gender. Each prescription form will be slightly different, with questions tailored to what the doctor will need to know in advance. For instance, if the patient is female and they are looking for a contraceptive prescription, the doctor might wish to know if they have had the contraceptive previously.

Doctors will be able to receive a prescription request from the patient, analyse the request and determine whether they should be prescribed it or not. They can then let the patient know that the request has been accepted or deny it with a reason why, such as insisting the patient must have a blood test first. The doctor will presumably phone the local chemist with the order.

### 2.3.6 Communication

Patients and Doctors will have at least one channel of communication in the form of an online chat. Doctors can create a chat if they have an appointment for that day marked type online. The doctor will be able to select the patients name and create a room; a space where the two can talk privately that no other doctor or patient will have access to. Once the doctor has created a room a message box will appear on the patients’ side where they can then converse with the doctor. Once the chat has concluded, the doctor will close the room, deleting the chat history and the room itself.

A secondary form of communication, a video consultation feature, could optionally be added if there is time to allow more flexibility to the feature. This would work in the same manner regarding a private room creation for both the doctor and patient to chat visually and audibly.

### 2.3.7. User Profile (Patient)

A patient will have access to a user profile page. On this page all the information from their registration will be populated in. They can edit non-vital information and it will edit in real-time on the screen. They will also be able to see a calendar which will populate all their appointments.

### 2.3.8. Dark Mode (Patient)

An optional feature of the application would be to provide patients with a dark mode. This would be a good user experience as some of the people using the application may be unwell and the brightness of the screen may cause them discomfort.

### 2.3.9 Track Medication (Patient)

Patients’ will be able to add medication to a calendar. This will be beneficial to them, as it will allow them to keep a visual record of all medication ingested, down to the date, time, type of medication and dosage taken. They will optionally be able to delete these records in case they make a mistake.

# 3. System Architecture

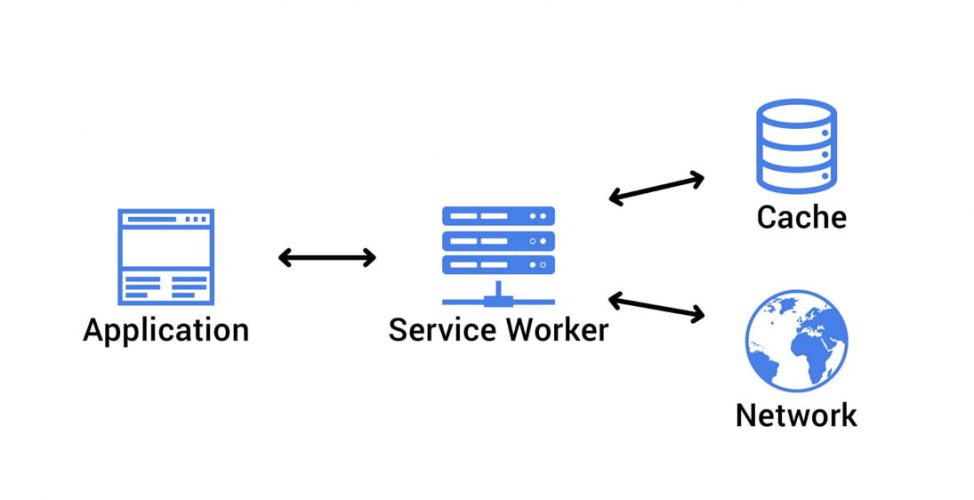


Figure 1: All Care - Doctor’s Office Web App Flow

# 4. Use Cases

## 4.1. Full User Use Case Diagram

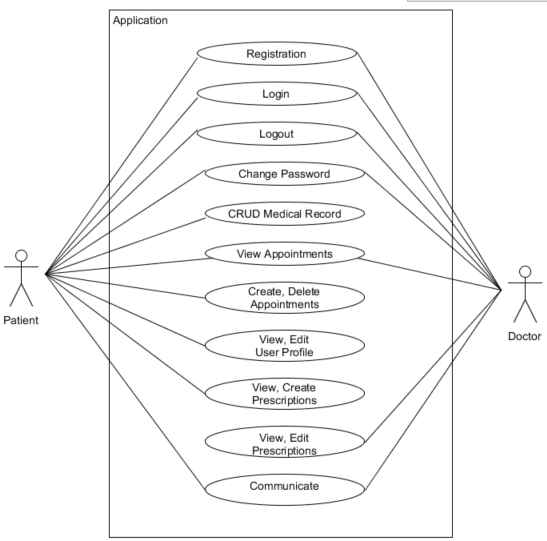


Figure: Full User Use Case Diagram

## 4.2. Detailed Use Cases

The following brief use cases refer to the interactions that both patients and doctors will have with the healthcare application. ‘CRUD’ refers to create, read, update and delete.

### 4.2.1. Register

|  |  |
| --- | --- |
| **Name** | **Register** |
| **Actors** | Patient, Doctor |
| **Preconditions** | The user has loaded up the application on their device. |
| **Activity** | This use case begins when the user loads up the application and is prompted to sign in or sign up. The user will select to sign up if they have not done so previously and upon doing so will be asked to enter relevant information i.e. name, email, password etc. The user will then confirm their password and submit the information which will be stored securely in the database so as to recognize the user in future. |
| **Consequences** | The user is registered with the application. |
| **Alternative(s)** | 1. The user enters an invalid email address or a password that does not meet the requirements i.e. must be of a specific length or have certain characters. 2. The user’s password confirmation does not match the original input. |

### 4.2.2. Login

|  |  |
| --- | --- |
| **Name** | **Login** |
| **Actors** | Patient, Doctor |
| **Preconditions** | The user has successfully registered with the application. |
| **Activity** | This use case begins when the user wants to log in to the application; they are prompted to sign in or sign up; they will select sign in and will then be prompted to enter their email address and password. Upon entering their credentials, the input is validated and if deemed correct the user is logged in to the application. |
| **Consequences** | The user can successfully log in to the application and can view their home screen. |
| **Alternative(s)** | 1. The user enters their credentials incorrectly and must re-enter the data. 2. The user enters their credentials incorrectly after a number of attempts and is temporarily locked out of the application until a set of requirements are met or certain amount of time has passed. |

### 4.2.3. Change Password

|  |  |
| --- | --- |
| **Name** | **Change Password** |
| **Actors** | Patient, Doctor |
| **Preconditions** | The user is registered with the application |
| **Activity** | This use case begins when the user forgets their password and goes to the forgot password button on the login page. A dialog opens and the user inputs their email address and submits. On successful submission an email is sent to the users email address with a link allowing them to change password. |
| **Consequences** | The user has successfully changed their accounts password. |
| **Alternative(s)** | 1. The user enters the wrong email address and does not receive an email |

### 4.2.4. Logout

|  |  |
| --- | --- |
| **Name** | **Logout** |
| **Actors** | Patient, Doctor |
| **Preconditions** | The user has successfully logged in to the application. |
| **Activity** | This use case begins when the user is finished using the application and wishes to log out of the system; they select the ‘log out’ button and are promptly logged out and redirected to the log in screen. |
| **Consequences** | The user successfully logs out of the application. |

### 4.2.5. View Medical Profile

|  |  |
| --- | --- |
| **Name** | **View Medical Profile** |
| **Actors** | Patient |
| **Preconditions** | The user has successfully logged in to the application. |
| **Activity** | This use case begins when the user wishes to view their medical record within the application. They select the ‘Medical Record’ option in the dashboard which they are promptly redirected to. From here the user can scroll through and view their personal medical details. |
| **Consequences** | The user successfully views their medical profile. |

### 4.2.6. Update Medical Details

|  |  |
| --- | --- |
| **Name** | **Medical Details** |
| **Actors** | Patient |
| **Preconditions** | The user has successfully logged in to the application and is viewing their medical profile. |
| **Activity** | This use case begins when the user wants to edit information within their medical profile. The user will be able to simply select the field and input a new value. Once the user clicks out of the field the value will automatically update in their record. |
| **Consequences** | The user successfully updates information in their medical profile. |
| **Alternative(s)** | 1. The user inputs values which are not accepted, and no changes are made to the field. |

### 4.2.7. View Appointment

|  |  |
| --- | --- |
| **Name** | **View Appointment** |
| **Actors** | Patient, Doctor |
| **Preconditions** | The user has successfully logged in to the application and is viewing their appointments. |
| **Activity** | This use case begins when the user wants to view their appointments. They will select the ‘Appointments’ option in the dashboard which they are promptly redirected to. Patients will be presented with a list of all upcoming appointments which display information about the date, time and appointment type. The user can also view old appointments in an alternative tab. Doctors will be able to view all appointments pertaining to them in a calendar format. |
| **Consequences** | The user successfully views their appointments. |
| **Alternative(s)** | 1. The time or date of the appointment does not suit the user and they delete the appointment. |

### 4.2.8. Add Appointment

|  |  |
| --- | --- |
| **Name** | **Add Appointment** |
| **Actors** | Patient |
| **Preconditions** | The user has successfully logged in to the application. |
| **Activity** | This use case begins when the user wants to book an appointment i.e. regular blood test in the clinic or follow-up on a consultation. They will select the ‘Appointments’ option where they will have the option of adding an appointment. When creating an appointment, they will select the type, date/time, doctor and include details. Upon entering this information, they will submit the appointment and be redirected to the view appointment screen. |
| **Consequences** | The user successfully books an appointment. |
| **Alternative(s)** | 1. The user does not submit the form before exiting the screen. 2. The user has already booked an appointment on this date. 3. The user tries to select a doctor who is already booked out. |

### 4.2.9. Receive Consultation

|  |  |
| --- | --- |
| **Name** | **Receive Consultation** |
| **Actors** | Patient |
| **Preconditions** | The user has successfully logged in to the application. |
| **Activity** | The use case begins when the user has successfully booked an appointment of type online. On the day of the booked appointment the doctor will create a room where the user and doctor can converse privately. The user will go to the “Chat” screen and wait to receive their message box once the doctor has initiated conversation. |
| **Consequences** | A message from the doctor appears on the users’ screen. |
| **Alternative(s)** | 1. The user does not attend the online consultation. |

### 4.2.10. Provide Consultation

|  |  |
| --- | --- |
| **Name** | **Provide Consultation** |
| **Actors** | Doctor |
| **Preconditions** | The user has successfully logged in to the application. |
| **Activity** | This use case begins when the user creates a room for themselves and a patient to talk in. On selection of the patient, the doctor can create a room and send the first message. |
| **Consequences** | The user initiates conversation with the patient. |

### 4.2.11. Provide Prescription

|  |  |
| --- | --- |
| **Name** | **Provide Prescription** |
| **Actors** | Doctor |
| **Preconditions** | The user has successfully logged in to the application. |
| **Activity** | This use case begins when the user goes to the prescriptions screen and clicks the add new prescription button. A list of prescriptions, depending on gender will populate into a selection. The user will select one and fill in the corresponding form before submitting it. |
| **Consequences** | The user has successfully requested a prescription. |
| **Alternative(s)** | 1. The user exits out of the form before clicking submit, clearing the form data. |

### 4.2.12. View Prescription

|  |  |
| --- | --- |
| **Name** | **View Prescription** |
| **Actors** | Patient |
| **Preconditions** | This user has successfully logged in to the application and has requested a prescription. |
| **Activity** | This use case begins when the user wants to view their prescriptions. They will go to the prescriptions screen and any prescriptions they have requested will be populated here. |
| **Consequences** | The user successfully views their prescriptions. |
| **Alternative(s)** | 1. The user has been denied a prescription. |

### 4.2.13. Accept/Deny Prescription

|  |  |
| --- | --- |
| **Name** | **View Prescription** |
| **Actors** | Doctor |
| **Preconditions** | This user has successfully logged in to the application and has been requested to approve or deny a prescription. |
| **Activity** | This use case begins when the user has ongoing prescription requests. They will be able to look at the form data, assess the prescription and either accept or deny it. |
| **Consequences** | The user successfully accepts the prescription. |
| **Alternative(s)** | 1. The user denies the prescription. |

# 5. Supplementary Specification (FURPS+)

The Supplementary Specification FURPS+ is a model for a classification system that was devised by Robert Grady at Hewlett-Packard and is used to identify any requirements that are not explored or easily defined in use cases. FURPS stands for Functionality, Usability, Reliability, Performance, Support ability and is used for functional requirements whilst the ‘+’ outlines any additional considerations for non-functional requirements such as design, implementation, interface or physical constraints.

## 5.1. Functionality

Functionality refers to the main product features, its capabilities and the systems functional requirements.

* The system will allow communication between a patient and doctor.
* Internet access is required to log in to the application.
* The application will allow the doctor to view appointments.
* Patients will be able to view their medical profile and update details on it.

## 5.2. Usability

Usability refers to the users’ experience (UX) with the application. The UX design, aesthetics, consistency, responsiveness and accessibility are all vital in providing an experience that engages the user. When developing the application, the ten principles of usability should be considered.

* The application should be easily accessible on a range of devices and browsers.
* The users should be capable of logging into the application within 5 seconds.
* The navigation of the application should be straight forward, and any information displayed clear, concise and visible. For instance, the user should be able to find and navigate between two separate pages within 10 seconds.
* A patient should be able to view their medical record within 5 seconds.
* A patient should be able to add an entry to their medical record within 15-20 seconds.
* A doctor should be able to view appointments within 5-10 seconds.

## 5.3. Reliability

Reliability refers to the availability and stability of the system and its general durability against failures and the extent and the length of time it takes to recover from failures.

* The application should be accessible most of the time, internet depending, as the only downtime should be for maintenance or unforeseen circumstances.
* Forms should submit 95% of the time.

## 5.4. Performance

Performance refers to how efficiently the system handles data throughput, the systems response time in transmitting and receiving data and the systems overall capacity and scalability in terms of handling multiple users simultaneously.

* The application performance will be dependent on the user’s internet connection.
* A data submission should take no longer than 10 seconds.
* The application should be capable of handling a large volume of users at any given time.
* The system should load in information dynamically instead of refreshing every time the user goes to a different section of the application.

## 5.5. Support ability

Support ability refers to the flexibility, adaptability and maintainability of the system and includes other requirements such as test ability, sustainability and compatibility with other platforms.

* The application will be supported on different platforms and devices and should work with any web browser.
* The application should be flexible enough to incorporate additional features in the future.

## 5.6. + (Security)

Security should take in to account the *“privacy, security, and confidentiality of virtual patient and e-patient records; legal and ethical considerations regarding data collected, analyzed, and distributed electronically; and policies regarding security and network standards.”*

The HIPAA Security Rule establishes national standards to protect individuals' electronic personal health information that is created, received, used, or maintained by a covered entity. The Security Rule requires appropriate administrative, physical and technical safeguards to ensure the confidentiality, integrity, and security of electronic protected health information.

Protecting Patient Safety Information - for doctor’s office we not suppose to share patient personal information, medical records etc. to any unauthorized person. This is strictly prohibited as per HIPAA guidelines, CMS rules - so we need to follow that strictly.

# 6. Metrics

*“Metrics are measures of quantitative assessment commonly used for assessing, comparing, and tracking performance or production.”* [3] In order to assess the overall success of the application, the project will be measured against a number of critical and optional criteria.

The critical criteria of the project is as follows:

* Authentication of users within the application
* The app must be ready and single page application.
* The patient must be able to view and make edits to their medical record.
* The patient must be able to view and make edits to their user profile.
* The patient must be able to view appointments, delete them so long as not on the day of the appointment, and book a new appointment once per day.
* The patient must be able to select a prescription and request it.
* The patient must be able to communicate with the doctor in some way.
* The doctor must be able to view any appointments pertaining to them.
* The doctor must be able to accept or deny prescriptions.
* The doctor must be able to communicate with the patient in some way.

The project must deliver a All Care - Doctor’s Office web application, capable of being used by multiple users and should have the ability to run on various devices and browsers. All users’ must be able to register, login, change password and logout of the application, and their credentials should be securely stored. There must be at least one functioning implementation of communication available to the patients and doctors. Patients should be able to store and access their medical information, as well as view and book appointments. Doctors should be able to view appointments. Doctors should also be capable of providing prescriptions if requested.

By developing the above features, the application should provide adequate online healthcare. To further measure whether the application is successful, a number of people will be asked to try out the application and give their feedback. Their opinions of their engagement with the application will determine whether the experience is satisfactory or not.

## 6.1. Potential Risks & Issues

During the development of this application, a number of potential risks may arise and have been considered prior to development so as to ensure unnecessary complications or delays are avoided if possible.

The chosen application model may present unknown limitations that may hinder the development of the proposed features. The languages, libraries and frameworks chosen may not be the most suitable for the application model and unforeseen compatibility issues between the technologies may also occur. There is also a possibility that the learning curve for these technologies may be too steep given that there is a lack of prior understanding of the chosen technologies and a limited amount of time to up-skill and develop the application, which could affect the overall quality of the project and limit the amount of features.

Due to the design of the application there will be limitations as to how much of it, if any, can function offline. As it is highly dependent on a stable internet connection, the users experience of the application is dependent on their network service, though this is an issue that is out of the developers’ control. Designing the application to load dynamically should hopefully provide a slightly more fluid experience for users with these difficulties.

A general issue that should be considered is the time management around the development of this application. In order to deliver a fully functioning application, given the time constraints, a project development timeline should be planned, and development of the application and its’ features will be divided into a number of iterations or sprints.

# 7. Conclusion

The above document provides an overview of the product concept, target users and the overall functionality of the application. It depicts the initial system architecture concept, the user’s interaction with the system and provides a detailed breakdown of each of the core functions available. The supplementary specification is taken into consideration with regards to the applications development and a number of risks and issues assessed. Metrics have been designed to measure the products success or failure.

## 8. Declaration



This document is for development team - internal use only - please do not share with any outside vendor, client or customer.

December 30, 2024

Ahsan Habib

Senior Business Analyst

## 9. Bibliography

### 9.1. References

|  |  |
| --- | --- |
| links to all the resources (websites, documents, etc.) that we have referenced in this document and to build up web app | |
| **Name** | **Location** |
|  |  |
|  |  |
|  |  |