Software Engineering

Software Requirements Document

< Project Name: Aceso- Cancer Patient Portal >

<Team Name: Animated Reality>

<Date: 9/6/20>

<Version: 1.0>

<Honor Code: We have abided by the UNCG Academic Integrity Policy>

<By:Amantii Samson
Gilberte Fleurisma
Imran Al Nafiee>

Table of Contents

1. Introduction

1.3 Purpose

This is the System Requirement Document (SRD) for the **Aceso – Cancer Patient Portal** project.

These requirements denote the work and specifications necessary to complete an open-source patient portal application for cancer patients to set and cancel appointments with private and/or general physicians.

1.4 Document Conventions

Not applicable

1.5 Intended Audience

This application is to be completed at the behest of the client, Ike Quigley, to service the market of cancer patients who are unsatisfied with the process of setting and canceling appointments with oncologists and primary care physicians.

1.6 Definitions/Jargon

Term	Definition
API	Application Programming Interface
GUI	Graphical User Interface
MVC	Model-View-Controller
SRD	System Requirement Document
PC	Personal Computer

1.7 Project Scope

The scope of this application is comprised primarily by the completion of its basic functionality; to set and cancel appointments for cancer patients through the use of a user-friendly Graphical User Interface (GUI).

1.8 Technical Challenges

The technical challenges of developing an application of this scope include:

- Learning and applying functionalities associated with an Application Programming Interface (API)
- Applying well designed and functional code that follows Model-View-Controller (MVC) architecture
- Utilizing JavaFX libraries to implement an aesthetically pleasing and adaptable GUI
- Implementing the API using proper formatting to allow for setting and canceling appointments

1.9 References

References are not applicable within the scope of this application.

2. Overall Description

2.1 Product Features

Features of this application include, but are not limited to:

- Setting doctor appointments for users
- Cancelling doctor appointments for users
- Creating an account
- Logging in to an account

Some of these features like setting an appointment and canceling an appointment are specific functionalities that are the core of this application, while the others are such as the login operation are implemented to create separation between user appointments.

2.2 User Characteristics

Although the primary users of this application are cancer patients who are scheduling doctor's appointments for chemotherapy and check-ups, the Cancer Patient Portal could also be utilized by patients who do not have cancer. The appointment setting functionality is easily usable by patients who do not have cancer, under the condition that they create an account.

2.3 Operating Environment

The application is easily usable on any personal computer(PC) within a given users home and will also be utilized by hospitals to check and confirm user appointments once they are created remotely.

The virtual nature of this application will allow for hospitals to engage more efficiently with their patients as direct communication through phone calls will no longer be necessary.

2.4 Design and Implementation Constraints

The design and implementation constraints for this application include:

- Use of a consistent data storage
- MVC program design
- Implementation of an API

The implementation of the API for this application has further constraints as the make appointment function of the API can only be utilized if the parameters are formatted in a pre-defined manner.

2.5 Assumptions and Dependencies

Due to the volatile nature of API's, it is likely that in the future the API implemented within this application to create, cancel, and call appointments may be subject to change and/or deletion.

In order to combat this eventuality, the application has been designed with consideration to the ability to easily replace the API with another that has a similar functionality. This operation shall be implementable through the use of the API adapter class and interface.

3. Functional Requirements

3.1 Primary

Sign In
The Aceso- Cancer Patient Portal shall provide compliant implementation of a SignIn operation in which the user must enter a valid email and password that have been stored in the consistent data storage in order to login to their account.

Set an Appointment
The Aceso- Cancer Patient Portal shall provide compliant implementation of a SetAppointment operation in which the user will be able to select a start time, end time, and appointment type for a doctor's appointment.

Cancel an Appointment
The Aceso- Cancer Patient Portal shall provide compliant implementation of a CancelAppointment operation in which the user will be able to delete pre-existing appointments by entering in the appointment ID of the scheduled appointment.

Create an Account
The Aceso- Cancer Patient Portal shall provide compliant implementation of a CreateAccount operation in which the user will be directed to enter their name, email, and a password.

Patient Details

The Aceso- Cancer Patient Portal shall provide compliant implementation of a PatientDetails operation in which the user, upon completion of the create account operation, will be directed to a form in which to enter personal information.

3.2 Secondary

Forgot Password

The Aceso- Cancer Patient Portal shall provide compliant implementation of a ForgotPassword operation in which the user enters email to retrieve a verification code from and uses the given code to create a new password.

Find Doctor

The Aceso- Cancer Patient Portal shall, in the future, provide compliant implementation of a FindDoctor operation in which the user is given the option to find doctors within the area using a find doctors API.

4. Technical Requirements

4.1 Operating Systems/Compatibility

The Cancer Patient Portal application shall be platform independent, meaning it will be executable on any operating system whether it be MacOS or the latest version of Windows.

4.2 Interface Requirements

4.2.1 User Interface

Users shall interact with the application through the use of a mouse/trackpad and keyboard on their PC's in order to navigate and operate the service. Future iterations of the application could allow for accommodations for users with disabilities.

4.2.2 Hardware Interface

The scope of this application does not require any external hardware interface.

4.2.3 Software Interface

Adequate functionality of the application entails a connection to an API as well as a consistent data storage.

4.2.4 Communications Interface

5. Nonfunctional Requirements

5.1 Performance Requirements

The application shall respond quickly enough to send created appointments to the API as well as the consistent data storage within an adequate amount of time. 60 seconds or less would be the optimal time to allow the program to save the appointment both within the API as well as the data storage system.

The optimal time may be limited by the API's ability to receive and send a request, though some adjustments could be made in the future to optimize this limitation.

5.2 Safety/Recovery Requirements

In the event of a system failure, user appointments will be recoverable from the API and consistent data storage through the get appointments operation. Appointments made by the user are consistently stored within the API's own schedule tracker and will not be removed until done so by the user.

5.3 Security Requirements

Security is kept at a minimum as the application does not host user sensitive information, though login credentials are required to access a user's account.

5.4 Policy Requirements

Policy requirements are not currently applicable to this version of the application but may be implemented in future iterations.

5.5 Software Quality Attributes

5.5.1 Availability

The application shall be available to all users at all times of the day in order to allow adequate time for users to schedule appointments days or even weeks ahead of time.

5.5.2 Correctness

The functional requirements of this application do not demand a high level of precision as its core functionality is to set and cancel appointments, which requires little to no correctness in terms of mathematical calculations.

5.5.3 Maintainability

5.5.4 Reusability

5.5.5 Portability

5.6 Process Requirements

5.6.1 Development Process Used

5.6.2 Time Constraints

5.6.3 Cost and Delivery Date

The estimated cost of an application of this scope and this