***Zachary Wilson***

***Samuel Oduwole***

***Maksym Larouche***

*POCKET DOCTOR*

System Requirements Document



Introduction (Max)

This System Requirements Document for *Pocket Doctor* will feature a Description Model, Class Diagram, Use Case Diagram, Use Case Scenarios, and System Sequence Charts. The Description model will give an overview of everything required for *Pocket Doctor*. The Class Diagram will show the relations between classes in the app. The Use Case Diagram and Use Case Scenarios will go in-depth about every step involved in the user or member doing anything with *Pocket Doctor*. Finally, the Systems Sequence Charts show the sequence diagrams of the Use Cases.



Description Model (Sam)

1. **Initial Requirements**

*POCKET DOCTOR* being a mobile application, will work on Android or Apple platforms and has requirements critical to its design. A user of *Pocket Doctor* should have access to an internet connection via a phone plan with their cell phone provider or Wi-Fi in order to be able to use the features/functions of pocket doctor such as keeping track of appointments, viewing a list of Accepted Insurances, Diet checker, aggregated news on disease monitoring and GPS navigation.

1. **Security**

*POCKET DOCTOR*’s security stems from internal security features along with third-party security protocols. Our application verifies through specific usernames and encrypted passwords along with email authenticity. The user to us will provide this information. Files and personal information are secured through different security protocols, but mostly we will use Google and the operating system of the user’s phones. The Google calendar or the default calendar app on the user’s phone handles the privacy of assignments within the calendar feature. We will pull information from Google API for our disease monitoring feature, recommended diet checker, and GPS navigation. Google security protocol will be used to keep search history private on the app, location will only be accessible to the user.



1. **Processes**
2. *Login Screen*

To begin with the app, a new user needs to create a new account. To create a new account, a user would click “New Account” and enter their Full Name, Email, Desired password, and Date of birth. A confirmation email will then be sent to them, and they will be taken to the first tab, the*“Home Screen”*. A user may only be associated with one account at a time.

If it is a returning user, the client may simply enter their email and password then click log in. If a user does not remember their password, they may click the “Forgot password?” link on the login screen, in which they will have to enter the email they used to sign up, and after that, the user will be sent a temporary password. The temporary password must be entered when they log in again and the system will prompt them to change to a new password.

1. *Dashboard Summary Tab*

The Dashboard Summary Screen is the initial tab the user sees when they log onto the app. This screen includes a profile picture, specific information about the user such as weight, height, and a sticky note box that will help them keep track of tasks like setting a goal for the day or adding an appointment.

This tab also includes the diet tab that keeps track of the user's progress with their diet plan. It also has a calorie counter that counts the calories in each food consumed by the user that has been entered into the app by the user themselves or recommended meals based on web information found that might be useful to individual clients. This part of the app helps users set goals, monitor their weight trends, and track their intake based on the specific diet plan they select. It also offers detailed nutrient information for each ingredient in your food log and a daily analysis to help keep you on track.

1. *Accepted Insurance Tab*

This screen will provide the user with care making it easier to find doctors without having to go to the hospital. The user enters what kind of doctor they want and the type of insurance they have. Our database provides a list of specialists that can provide care and will accept the insurance of the client.

1. *Calendar*

The Calendar tab displays an overview of appointments and to-do tasks. This screen, populated by Google Calendar, produces a monthly calendar where users may click on specific dates and see what they need to get done on due dates. This tab helps users visualize the timeline for their tasks.

1. *GPS Navigation*

This tab is straightforward and simple as it seems. When a user chooses a doctor to go to or wants to see the location of a potential primary physician for their needs they can check their location to see if it is convenient for them to travel. Direction is populated by Google Maps and gives users full control in deciding how they want to embark on the trip. Car, bicycle, or transit all routes are available. The navigation system tells the user the direction for each turn they take to reach their destination. GPS works altogether weather, so they don’t have to worry about climate as in other navigating devices. This costs you very low compared to other navigation systems.

1. Tools

* *Support*

Here, is when Pocket Doctor users get to interact with the unlimited resources available to them. This screen will include a phone number and email address a representative can be reached all they have to do is call or fill out a request on the question/issue they are having a representative will be in touch with them 24-hour.

* *Health Encyclopedia*

This tab is a Medical Encyclopedia integrated with *medlineplus.gov* An external API that provides over 4,000 articles about diseases, tests, symptoms, injuries, and surgeries. It also contains an extensive library of medical photographs and illustrations. Users can use this feature to find out specific information on any health-related topics that may concern them or just for general knowledge.



**Class Diagram (Max)**

Create a class diagram. The Class Diagram should contain all the system objects, their attributes, and any known methods. This diagram may be included as a separate file – it does not need to be inserted into this Word document.

**Use Case Diagram (Max)**

Create a Use Case Diagram for all the “uses” of your system. This diagram may be included as a separate file – it does not need to be inserted into this Word document.

**Use Case Scenarios (Zach)**

Create an intermediate (sea level) Use Case Scenario for each use case of the system. This intermediate scenario should include an enumerated list of steps involved in the activity as well as any exception conditions.

**System Sequence Charts (Sam)**

For each Use Case Scenario, provide a sequence diagram. Use your class diagram, use case diagram, and scenarios to create the corresponding Sequence Diagram.