Software Requirements Specification

for

Team 6

Version 1.0 approved

Prepared by Samantha Stone, Caesar Nguyen, Justen Yergo, Suyog Dahal

AstraZeneca

8/31/2022

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.0 Purpose 1

1.1 Intended Audience and Reading Suggestions 1

1.2 Product Scope 1

1.3 Product Value 1

2. Overall Description 2

2.0 Product Perspective 2

2.1 Product Functions 2

2.2 Operating Environment 3

2.3 Design and Implementation Constraints 3

2.4 Assumptions and Dependencies 3

3. External Interface Requirements 4

3.0 User Interfaces 4

3.1 Hardware Interfaces

3.2 Software Interfaces

3.3 Communications Interfaces

4. System Features/Requirements

5. Other Nonfunctional Requirements

5.0 Performance Requirements

5.1 Safety Requirements

5.2 Security Requirements

5.3 Software Quality Attributes

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Samantha, Justen, Suyog, and Caesar | 9/1/2022 | Document is first created, sections 1 and 2 are created and revised. | 1.0 |
| Samantha, Justen, Suyog, and Caesar | 9/13/2022 | Sections 2 and 4 are created and revised | 2.0 |
| Samantha | 9/19/2022 | Revised section 4 | 2.1 |
| Samantha, Justen, Suyog, and Caesar | 10/19/2022 | Revised section 4, section 3 is created | 3.0 |



# Introduction

## Purpose

This document outlines the various requirements to be met by MedAlarm with respect to its stakeholders, end users, software developers, and owners. This application is expected to remind end users that it is time to take their medications. This document will encompass all system parts including front end and backend systems/operations as well as non-functional requirements.

## Intended Audience and Reading Suggestions

This document is intended for all members working on developing, advertising, managing, testing, documenting, and using the application. This project is a prototype for a medication reminder system and is available to a doctor, patient, and family members of the patient. The ideas have been implemented under the guidance of Doctors, Patients, as well as Family of the Patient. The application is useful for our patients who are forgetting to take their medications, Family who need to know if someone has taken their medication, along with doctors who need to know if the patient has missed more than a certain number of medications.

## Product Scope

The purpose of this software is to act as a form of reminder for the patient to take their medications and give a platform to loved ones and medical personnel to monitor if the patient took or did not take their medications. It was initially intended for senior citizens who have the highest proportion of need for medications and refills annually (ranging from 87-91% for ages 65+), but this software is available to people of all age ranges. However, this software will still be geared towards a senior citizen demographic as it seems to be our primary audience. Complete with larger fonts for increased legibility, larger buttons for ease of access, a loud but recognizably unique alarm so the patient can hear the notification and relate it to our software, and emergency notifications to medical personnel and loved ones, specified by the user, in order to monitor the user's activity in the form of mobile notifications and a phone call if needed.

## Product Value

The product value of this application is that it is a medical application that provides its users with the ability to be reminded to take their medications with a simple push of a button. It will provide families with peace of mind knowing that their loved ones are taking their medications on time with this application, and as for the doctors, they will also know when the patient is not taking it as well. For AstraZeneca, they stand to gain rapport with the community through the use of this free application.

# Overall Description

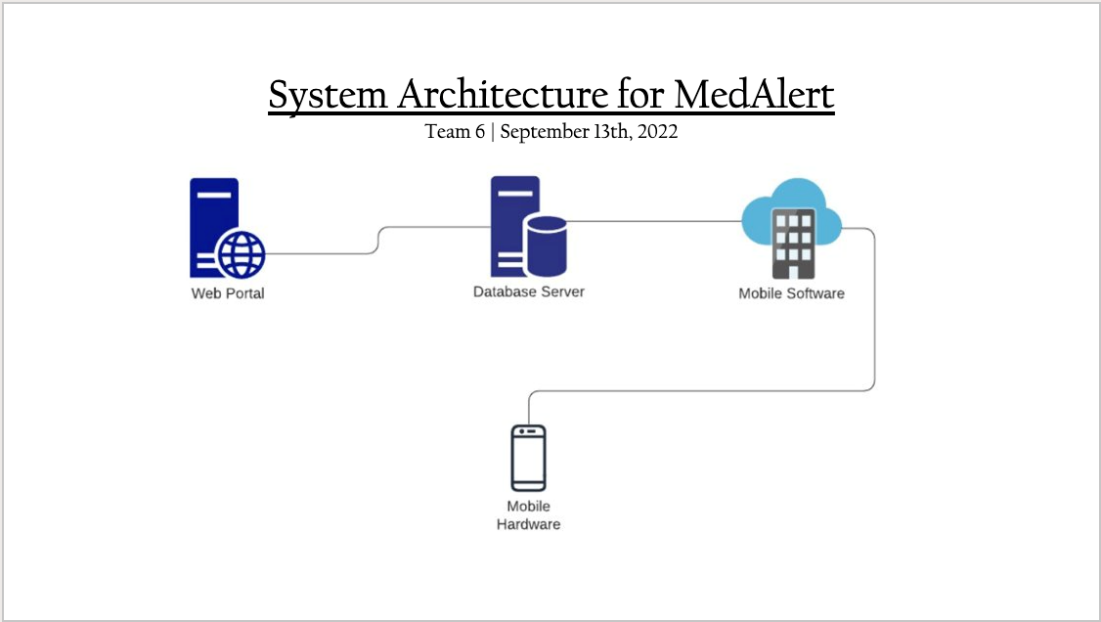
## Product Perspective

MedAlarm is a unique, self-contained system. This system allows health providers to input patient prescriptions and details regarding how and when to take the medications. If a patient fails to take their prescribed medication when prompted by the app, family members and health providers will be notified.

The application will be available for download on both iOS and Android. Upon the first login, the user will be prompted to enter in their default username and password that is provided by their provider, and then enter in detail about their emergency contacts. This information will be stored in a centralized secure database and will be linked to their unique profile that was provided by their medical provider.

This data-centric application will only be useable if their medical provider has signed them up and created a profile for them.

Geolocation services would need to be enabled on this application only for the purpose of tracking the user and send their location to medical providers and emergency contacts if they were to miss an alarm. Permission to view contacts will be optional as well if the user wants to import a contact from their local device as an emergency contact instead of manually adding it. These data will be securely stored in the database under the user’s unique profile, which will ensure confidentiality and adherence to HIPAA.



## Product Functions

The MedAlarm application should be able to perform the following major functions:

Login:

Upon opening the application after being provided the download link from the provider, the user should be able to log in to the app using the provided credentials. The user will be given an option to remember their credentials via Face ID/Touch ID so further input at start-up is not necessary.

Logout:

When a user clicks logout, all personal data will be removed until the user logs back in using their credentials.

Display Emergency Contacts:

This application identifies the user’s emergency contacts and medical provider and displays it within the user’s profile settings. The emergency contacts can be changed on the user’s and their medical provider’s end, but the primary medical provider cannot be changed from the user’s side.

Present Medication:

The app should be able to display the current medication and dosage for the user. This information is inputted from the medical provider’s side via the web portal.

Current Alarm Notice:

The app is going to notify the user of when to take their medications at a certain time. It is going to look like the stock alarm interface, but there will be 2 buttons, one to confirm that they’ve taken their meds, and the other to snooze for 5 minutes. If no buttons are pressed within a 10-minute time frame, the application will notify the emergency contacts and the medical provider, which they will also be provided with the location of the user. The time to take the medication cannot be changed by the user; it can only be modified by the medical provider with permission from the doctor.

## Operating Environment

Windows 11/10/8 (or) Windows Server 2022/ 2019/ 2016/ 2012 R2, latest macOS, latest iPhone, latest android. Browsers will need to be Chrome/ Firefox/ Edge, needs to enable JavaScript, cookies, working with iframes and third-party cookies.

## Design and Implementation Constraints

The constraints in implementation and design that will limit the options available for the developing team are as follows:

Developer access to working medical records in order to test, build, and maintain software will cause problems with HIPAA violations as they do not have the necessary credentials or authority to view patient medical records. This, and the function to send Emergency Contacts or caregivers a notification regarding the status of the patient. Although, this is easily mitigated by the acquisition of a BAA (Business Associate Agreement) that will detail our usage and responsibilities with patients’ PHI (Private Health Information).

Hardware limitations could prove to be an issue with downloading and using the software, all end-users must have the most updated operating system on their personal and mobile devices, and they must have some form of internet access in order to interact with the application; the provider needs this to use the web portal as well. The database memory must be adequate to sustain the application and the patients’ information.

Both applications, the web-based database and the mobile notification application, and the Provider’s EMR must communicate with one another in order to accurately get the patients' medical records and emergency contact information to notify both parties the correct notifications at the correct times. (To remedy this constraint, we have decided to use Amazon Web Services (AWS) to host our database and integrate it with the mobile app.)

Financial constraints include employee pay and the outsourcing of different services like AWS, these all must remain within budget without compromising the integrity of the applications.

All end-users must have their time zone or location specified prior to using this application to ensure the quality and correctness of the notifications. This needs to be in the web portal database before enrolling a patient so they get notified to take their medications at the correct times.

## Assumptions and Dependencies

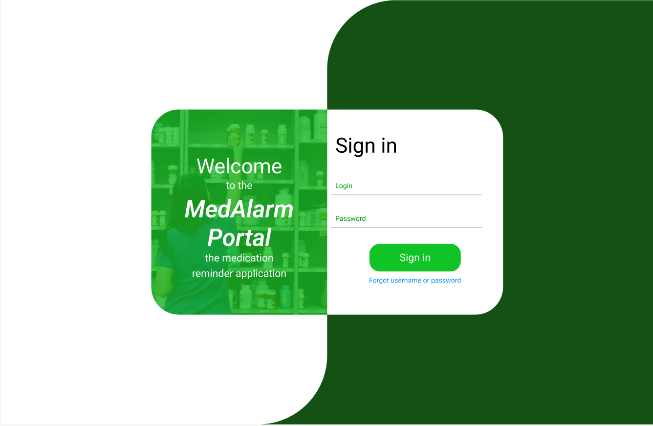
It is assumed that patient data will be accessible to the Doctor, along with his staff through our web-based portion of our application. We assume that the Doctors office has internet access because they will be able to access/enroll patients through the web-based application. We assume that our patients and their caregivers all have a mobile device (IOS or android) updated to the latest software so they can accept our invitation link via message to download the application. We assume that the patients and caregivers all have an internet connection stable to use the app in order to send Doctors and Caregivers information if the patient missed their medication. Our application is web based on the Doctor’s end, and Application based on the patient’s end. We assume that the Patient can get the application installed on their device along with sign up with their account, and that the Doctor/Staff have the knowledge to add/update/delete a patient within the system. This project Depends on Amazon Web Serivices (AWS) as our main cloud platform for organizing and storing data from our website to the mobile applications. Specifically, AWS AppSync. Our application is dependent on AWS AppSync to provide a single endpoint to securely query and retrieve new data from multiple databases. Another dependency for our application is Amazon Pinpoint. Amazon pinpoint is used for sending notifications such as SMS, Push, Voice, and in-app messages. We need Amazon pinpoint in order to send patients an SMS to sign up and download our application, and to notify caregivers or the Doctor if a patient may have missed a medication by a few minutes, or to remind the patient themselves.

# External Interface Requirements

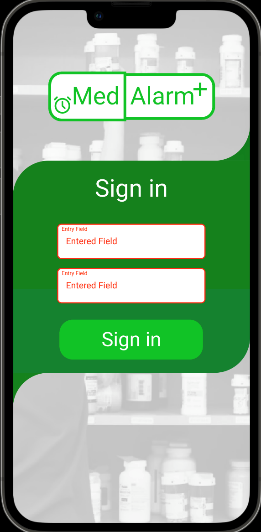
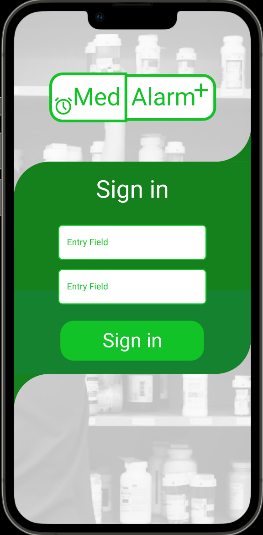
## User Interfaces

There are two separate user interfaces: that of the portal which is intended to be used by the Doctor’s Providers and that of the mobile application which is intended to be used by the patient and who the patient signs up to be their emergency contact. All parts of each user interface intend to be as user-friendly as possible, providing simplicity and usability with ease to the intended user. Each will begin with a simple log-in flow in order to maintain security and accessibility to the full application. In the case of incorrect login information inputting into the system, an error state will show an appropriate error message to inform the user of the mistake.

MedAlarm Portal Login Screen and Error States:



Medalarm Application Login Screen and Error States:



MedAlarm Portal, Provider Interface Specifications:

Once logged in, a Provider can navigate to the home page of the portal. On the top of the screen, they will see their job title and login username to confirm they are using the correct account. In the middle of their screen, the logged in user will see buttons to interact with; search patients, add patients, list of patients, and incomplete forms.

When interacted with, the search function gives you the option to search by first name, last name, phone number, or a combination of all three of those. After searching, you can click on a patient and it’ll pull up a quick view of their file with a button on the bottom to open it. After pressing open you can update, save, and enroll and patient or leave the file as it is.

The add patient function will give you a form to fill out necessary information to enroll a patient, at the end of the form it will give you the option to just save or “save and enroll” which will enroll the patient into the MedAlarm system and send them the text to download the application.

The list of patients button will pull up a complete list of all the patients in the database, there you can sort by different fields such as “enrolled” or “not enrolled”, “date added”, and “name” which will sort the list alphabetically by last name. You can also access patient files from the list like in the search function and update, save, or enroll them or leave the file as it was.

The incomplete forms button will bring up a list of all patients missing some necessary information in order to enroll them into the MedAlarm system. Like the search function, when a patient is clicked it’ll bring up a quick view of their file and can be interacted with in the same way.

MedAlarm Application, Patient/Emergency Contact Interface Specifications:

Once logged in, a Patient or Emergency Contact can see the time in which they need to take their medication, the amount of snooze attempts they have, the time in which they need to interact with the app before it is consider either a snooze or a missed alarm, and a settings button. The settings button, when clicked, will allow the user to customize their alarm settings; their alarm sound, how loud the alarm is, if they want their flashlight to flash.

When an alarm is active on the lock screen of both users, they will see a notification.

For Patients, it will sound when they need to take their medication. For them, it will be a notification reminder to take their medication, a countdown timer for them to interact with the app, a snooze button and a confirm taken button. The snooze button will enable the snooze time before the alarm sounds again and display a countdown timer. The confirmation button will end the alarm and the app not be active until their next alarm.

For Emergency Contacts, their alarm will only be active when a patient has run out of snoozes and has not confirmed they have taken their medication or if the timer to interact with the app runs out and the patient has not interacted with the alarm. Their notification will be a persistent emergency notification that will sound in periodic bursts until the notification is seen.

## Hardware Interfaces

The MedAlarm Portal Web Application runs on its own server, there are no specific hardware requirements of the end-user to access it other than connection to the internet.

The MedAlarm Mobile Application requires users to have iOS 16.0.3 or later for iOS users and Android OS 13 Tiramisu for Android users in order to download the application. Users must stay up to date with the native OS updates to continue using the app as the app will be updated for the current native OS.

## Software Interfaces

This product will need the use of several external software interfaces. Each mobile platform will need its own front-end development because they each use a different operating system and programming language. As we require customers to have iOS 16.0.3 or later, development for iOS will begin using the tools XCode and Swift, which both are installed at the latest version. We'll be using the Android Studio SDK for Android because it provides the most recent libraries for Android OS 13 Tiramisu. As the medical assistants and doctors will be utilizing the MedAlarm web portal, this interface for the web portal will be created with HTML and CSS. For the website's interactive features, such as data retrieval, form submission, or updating the patient database, JavaScript and React will also be used. The patient database will be created using AWS Lambda, Pinpoint, and DynamoDB as well. With AWS services, data will be accessible by all our front-end software components, enabling each component to access it at the fastest feasible data transmission speeds.

## Communications Interfaces

The MedAlarm web browser portal shall use the HTTP protocol for communication over the internet and internet communication will be through TCP/IP protocol suite. The Port number used will be 80. There shall be a logical address of the system in IPv4 format. For our application, SMS in the state of an emergency alert (AMBER Alert) will serve as the primary communication interface through the service Twillio. To make sure the system is working properly, we will use the regular SMS to send the patient and the emergency contact a first confirmation text. Anytime the program notices a patient missed an alarm, the SMS emergency notification will be used. If all SMS notifications are missed, then a phone call from the medical provider’s own cellular service contacts the emergency contact’s phone, which also is using cellular service. From this point on, all users can create an account via phone number and if they are enrolled within the medical provider’s systems. Doctors and medical assistants will be able to adjust the medicine and the appropriate times to take it by accessing the patients' database using the web browser portal. The portal will connect to the server via AWS Lambda, and it will only be permitted to add new rows, remove existing rows, or modify the data within the rows of the patients’ data.

# System Features/Requirements

* 1. **Provider Login**
     1. *Description and Priority:*

Provider provides username and password for MedAlarm provider portal to access providers MedAlarm database.

Priority: High

* + 1. *Stimulus/Response Sequences*

Provider enters username to username field. Provider enters password into password field. Provider clicks sign in to log into MedAlarm database. If successful, provider is logged in an home screen is generated. If unsuccessful, appropriate error message is generated.

* + 1. *Functional Requirements*

REQ-1: Form posts username and password from login form in a secure manner

REQ-2: collected data is securely checked against existing list of usernames and passwords

REQ-3: If the user is found: provider is routed to the portal home page for patient search

REQ-4: if the user is not found: the provider is notified with red bold text that the username and password combination is not found

* 1. **Provider Forgot Password**
     1. *Description and Priority:*

The provider enters the last known username and associated email into the system to reset the password via an emailed reset link.

Priority: Moderate

* + 1. *Stimulus/Response Sequences*

The provider enters the username into the username field. The provider enters the email into the email field. Provider clicks forgot password. New page alerts provider: “If the username and email combination exist in the system a password reset link will be sent to the corresponding email.” If successful, a password reset email is generated and sent to the provider

* + 1. *Functional Requirements*

REQ-1: Form posts username and email from forgotten password form in a secure manner

REQ-2: Collected data is securely checked against the existing list of usernames and email accounts

REQ-3: If the user is found: the provider is emailed a link to reset the password

REQ-4: If the user is not found: exit the search and notify the provider no such account exists

* 1. **Provider Search Patient**
     1. *Description and Priority:*

The provider uses patient search to query patient demographics and alarm settings for their existing patients in the Med Alert system.

Priority: High

* + 1. *Stimulus/Response Sequences*

Provider navigates to the search option. The provider chooses search criteria from a dropdown (First name, last name, DOB). The provider enters searchable data into a text box. Provider clicks search. Providers’ data is queried for patient information. All patients exactly matching searched patient data are returned in table format. Patient names are clickable so that the provider may access the patient information for editing. If the search text box is empty or the searchable criteria dropdown option is not selected, nothing is returned.

* + 1. *Functional Requirements*

REQ-1: Provider has successfully signed in to an enabled MedAlarm Account.

REQ-2: Provider selected search criteria from dropdown box

REQ-3: Provider has entered text into search text box

REQ-4: Provider has clicked search button to query results

REQ-5: If patients are found: Table with patient first name, last name, and DOB is generated and returned to list all patients that match search criteria.

REQ-6: If patient(s) not found: Table is not generated. Note appears on screen that no patients matched the searched criteria

* 1. **Provider Update Patient**
     1. *Description and Priority:*

Provider accesses existing patient chart to update patient demographics, alarm settings, and/or emergency contact information

Priority: Moderate

* + 1. *Stimulus/Response Sequences*

Provider has searched patient and has navigated to patient chart by clicking on the patient name. Provider navigates to fields to be edited (patient demographics, Medication alarms, or Emergency Contact. Provider applies necessary updates. Provider saves changes. Database is updated.

* + 1. *Functional Requirements*

REQ-1: Provider has successfully logged in and queried a patient that exists under their organization.

REQ-2: Provider submits updates to existing patient via secure webform

REQ-3: Data is checked for valid characters (no blank entries in required fields)

REQ-4: Valid data is mapped to corresponding field and existing data is overwritten including fields that contained no changes

REQ-5: Updates with invalid submissions are ignored, system notifies user which fields contained invalid data

REQ-6: Updated information is sent back to the form to reflect the changes that have been saved

REQ-7: Page is updated with notification that updates have been saved

* 1. **Provider Add/Enroll New Patient**
     1. *Description and Priority:*

Provider selects add new patient from home page or patient search screen. New patient page is displayed so that provider may fill in patient demographics, alarm settings, and emergency contact information. Patient first/last name, DOB, and phone number are required fields.

Priority: High

* + 1. *Stimulus/Response Sequences*

Provider adds new patient by at least entering patient first name, last name, DOB, and phone number. If available, provider enters patient alarm settings including alarm name, time, time zone requirements, days needed, maximum snooze time, provider alert enabled, provider alert requirements, emergency contact name, and emergency contact phone number. Provider saves entered data. Data is error checked. If errors occur provider is notified of errors and new patient is not created. If no errors occur, data is processed and saved to database. A text is generated and sent the patients phone with a link to download the application and information about the patient login credentials.

* + 1. *Functional Requirements*

REQ-1: Provider successfully logs in and clicks “Add new patient” from home page or patient search screen. A new webpage is generated with blank fields so the user may enter patient information.

REQ-2: Collected data is checked for errors (missing required fields, max character length, and syntax issues for data types).

REQ-3: If errors exist user is notified and patient is not added to the system

REQ-4: If no errors exists, collected data is securely checked against existing list of patients. If an exact match to patient first/last name and DOB is found the new patient is not added and the user is notified a the patient already exists

REQ-5: If no errors and existing patient exist, new patient is added and corresponding tables are updated with new patient data.

REQ-6: Adding a new patient triggers and enrollment text to be sent to the patient (and emergency contact if applicable) phone with information related to downloading the app and the enrolled username/password.

* 1. **Provider is Alerted that patient missed medication**
     1. *Description and Priority:*

Providers office is notified in the event the patient has missed specified number of medication alarms or medication cant be found

Priority: High

* + 1. *Stimulus/Response Sequences*

Provider has enrolled patient and the specified number of medication alarms have been missed by the patient. The system triggers an automated call response to the provider depending on their notification settings. A notification is sent containing the patient first initial, last name, DOB, and alert type as per the providers settings (# medication alarm missed)

* + 1. *Functional Requirements*

REQ-1: Missed alarm on patient phone triggers count

REQ-2: Missed alarm count update check against triggers provider notification settings for patient.

REQ-3: If missed alarm count is greater than or equal to provider notification trigger, notification settings are checked and notification is released to required system (email/text).

REQ-4: If the count is less than provider trigger nothing is done.

* 1. **Patient downloads app via text link**
     1. *Description and Priority:*

Provider has enrolled patient with correct name, DOB, and phone number. Patient receives text with link to download application and information about login credentials Priority: High

* + 1. *Stimulus/Response Sequences*

Patient receives text message to phone and opens the message. Patient clicks on link to download application. Application downloads. First opening of application prompts user to login for first time or sign into existing account. User selects login for first time. Patient is prompted to select doctors office name from dropdown list and enter first name, last name, and DOB. Database is queried for patient related to patient data entry. If no patient is found given the provided information, the patient is notified no such patient exits and to contact doctors office for more information. If the patient is found, patient is prompted to enter username and password. Data is submitted and patient table is updated with username and password. Patient is prompted to login.

* + 1. *Functional Requirements*

REQ-1: The link directs the patient to download the app.

REQ-2: Once the app is downloaded and opened initial login screen prompts the patient to either login into an existing account or create a new account.

REQ-3: If the user selects login to an existing account, the login page is loaded

REQ-4: if the user selects login to new account, database is queried for list of doctors offices, dropdown option allows patient to select from list of doctors offices, patient enters first name, last name, and DOB. Doctors office patient table is queried for an exact match.

REQ-5: If no exact patient match occurs, patient is prompted to contact doctors office for more information.

REQ-6: If an exact match exists with no previous username/password saved, the patient is prompted to save and username and password. Patient username and password are updated in database, patient is prompted to stay logged in, answer is recorded, and patient page loads to login screen.

REQ-7: If an exact match exists but username and password exist in database the patient is notified login information already exits and is prompted to login to existing account.

* 1. **Patient Login**
     1. *Description and Priority:*

Patient has downloaded application and is prompted to login. Fields are present for patient to enter username and password. Patient is notified if username/password are incorrect. Patient is routed to alarm home page if correct username and password are entered.

Priority: High

* + 1. *Stimulus/Response Sequences*

Patient enters username and password to form fields. Data is checked against database. If user exists, patient is routed to their alarm home screen. If username/password combination does not exist patient is alerted of error with user/password combination.

* + 1. *Functional Requirements*

REQ-1: Form posts username and email from form in a secure manner

REQ-2: collected data is securely checked against existing list of usernames and email accounts

REQ-3: If the user is found: the user is directed to the alarms homepage

REQ-4: If the user is not found: exit the search and notify the user

* 1. **Patient forgot password**
     1. *Description and Priority:*

The patient enters the last known username into the system to reset the password via and text reset link.

Priority: Moderate

* + 1. *Stimulus/Response Sequences*

The patient enters a username in the username field. The patient clicks forgot password. The database is searched for patient username. If found, the patient is prompted to enter the doctor’s office name from the dropdown list queried from the system, the patient is prompted to confirm the first name, last name, and DOB. The database is searched for an exact match of first name, last name, and DOB connected to the doctor’s office. If found, the patient is prompted to update the password. If not found, the patient is instructed to contact the provider’s office for assistance.

* + 1. *Functional Requirements*

REQ-1: Form posts username from forgotten password form

REQ-2: Collected data is securely checked against the existing list of usernames

REQ-3: If the user is found: the patient is prompted to confirm their name and DOB. Submitted information is checked against found demographics tied to the username.

REQ-4: If information tied to username matches patient is prompted to reset the password

REQ-5: If the username is not found or submitted information does not match found username: the patient is notified with red bold text that the username and/or demographics were not found and prompted to contact the providers office for assistance.

* 1. **Patient Marks Medication As Taken**
     1. *Description and Priority:*

Patient medication alarm has alerted the patient. Patient turns off alarm by marking the medication as taken

Priority: High

* + 1. *Stimulus/Response Sequences*

Patient clicks on medication alert to open app. Patient presses “DONE” button. System records medication as taken and sets snooze count and countdown time to zero in the database. Last login time/date is updated in database. The next medication alarm is queued.

* + 1. *Functional Requirements*

REQ-1: System medication alarm is triggered by (current date/time + 5 seconds) matching alarm date/time.

REQ-2: Alarm settings are queried

REQ-3: Alarm is set to sound when current date/time = alarm date/time

REQ-4: User opens phone and clicks on notification. App alarm buttons are displayed on the user’s screen.

REQ-5: The user presses the “DONE” button and the system response is recorded in the application.

REQ-6: Recorded response triggers alarm to be stopped on the users device

REQ-7: Application exports recorded response and current date/time to database

REQ-8: Database updates snooze count and time to 0.

* 1. **Patient Snooze Alarm**
     1. *Description and Priority:*

The MedAlarm app has alerted the patient to take their medication. PThe patienthas clicked opened the app by clicking on the notification. The patient has clicked on the “SNOOZE” button.

Priority: High

* + 1. Stimulus/Response Sequences

The patient clicks on the medication alert to open the app. Patient presses the “SNOOZE” button. The system records the medication as snoozed and increments the snooze count. The snooze countdown begins. Max snooze time is decremented. The last login time/date is updated in the database.

* + 1. *Functional Requirements*

REQ-1: System medication alarm is triggered by (current date/time + 5 seconds) matching alarm date/time.

REQ-2: Alarm settings are queried

REQ-3: Alarm is set to sound when current date/time = alarm date/time

REQ-4: User opens phone and clicks on notification. App alarm buttons are displayed on the users screen.

REQ-5: The user presses the “SNOOZE” button and the system response is recorded in the application.

REQ-6: Recorded response triggers alarm to be snoozed on the users device

REQ-7: Application exports recorded response and current date/time to database

REQ-8: Snooze countdown begins

REQ-9: Max snooze count down begins

REQ-10: Database updates snooze count and login date/time

* 1. **Patient Mark Medication as Lost/Out**
     1. *Description and Priority:*

Patient medication alarm has alerted the patient. Patient reaches the max snooze amount set for their medication. New menu appears on screen to mark the medication as lost or empty

Priority: Moderate

* + 1. *Stimulus/Response Sequences*

The patient clicks on the medication alert to open the app. The patient presses “SNOOZE” or does not hit the “DONE” button enough times to reach their max snooze time. The menu now offers to stop the snooze by marking the medication as taken or marking their medication as lost or empty. Marking the medication as lost or empty notifies the provider and emergency contact when applicable.

* + 1. *Functional Requirements*

REQ-1: System medication alarm is triggered by (current date/time + 5 seconds) matching alarm date/time + max snooze time.

REQ-2: Alarm settings are queried

REQ-3: Alarm is set to sound when current date/time = alarm date/time + max snooze time

REQ-4: User opens phone and clicks on notification. App alarm buttons are displayed on the user’s screen.

REQ-5: The user presses the “NEED ASSISTANCE” button and the system response is recorded in the application.

REQ-6: Recorded response triggers notification to be sent to doctor and emergency contact if enabled.

REQ-7: Application exports recorded response and current date/time to database

REQ-8: Database updates login date/time

* 1. **Patient Adds Other Alarm**
     1. Description and Priority:

The patient adds extra alarms to the application.

Priority: Low

* + 1. *Stimulus/Response Sequences*

Patient logs into MedAlarm app. Database logs in last login time and date. The patient navigates to alarm settings. The patient clicks “add new alarm”. The patient names the alarm and sets the day and time requirements. The patient clicks save.

* + 1. *Functional Requirements*

REQ-1: Patient locates application alarm settings to store new alarms

REQ-2: Application stores extra alarms locally

* 1. **Emergency Contact Enrollment**
     1. *Description and Priority:*

Emergency contact is enrolled to receive notifications by the provider or through the patient application. The emergency contact is texted a link to

Priority: Low

* + 1. *Stimulus/Response Sequences*

The provider or patient navigates to the emergency contact section of the MedAlarm application. The patient or provider enters the emergency contact information into the contact first name, last name, and phone number fields. Upon saving the updated information and enrollment text thread is sent to the emergency contact.

* + 1. *Functional Requirements*

REQ-1: Form post emergency contact information from the form in a secure manner

REQ-2: Collected data is securely checked against the existing list of emergency contacts related to the patient. If changes are observed the enrollment text is sent to the emergency contact.

* 1. **Emergency Contact Login**
     1. *Description and Priority:*

Emergency contact has downloaded the application and is prompted to log in. Fields are present for contact to enter username and password. Contact is notified if username/password are incorrect. Contact is routed to the notification settings home page if the correct username and password are entered.

Priority: Low

* + 1. *Stimulus/Response Sequences*

The contact enters their username and password to form fields. Data is checked against the database. If the user exists, the contact is routed to their notifications home screen. If the username/password combination does not exist contact is alerted of the error with the user/password combination.

* + 1. *Functional Requirements*

REQ-1: Form posts username and email from the form in a secure manner

REQ-2: Collected data is securely checked against an existing list of usernames and email accounts for emergency contacts

REQ-3: If the user is found: the user is directed to the notification homepage

REQ-4: If the user is not found: exit the search and notify the user

* 1. **Emergency Contact Adjust Notification**
     1. *Description and Priority:*

The contact logs into application and navigates to notification settings. The contact selects which patient notifications are to be adjusted for. The contact selects the frequency and events to receive notifications for and the settings for each alarm.

Priority: Low

* + 1. *Stimulus/Response Sequences*

Contact selects notification settings. The database queries which patients the contact account is linked to. The database queries provider alarms tied to the patient. The application displays the patient’s first name, last name, and alarms. Setting selection is updated upon selection.

* + 1. *Functional Requirements*

REQ-1: Database returns patient names and alarms associated with the contact user account.

REQ-2: Contact can edit personal notification settings for each alarm set up by the provider for each patient they are an emergency contact for.

* 1. **Emergency Contact Alerted** 
     1. *Description and Priority:*

Emergency contact is notified in the event the patient has missed a specified number of medication alarms or medication can’t be found

Priority: High

* + 1. *Stimulus/Response Sequences:*

The emergency contact has been linked to an enrolled patient and the specified number of medication alarms have been missed by the patient. The system triggers an automated notification response to the contact (depending on their notification settings). A notification is sent containing the patient first initial, last name, and alert type as per the contacts settings (# medication alarm missed)

* + 1. *Functional Requirements:*

REQ-1: Missed alarm on patient phone triggers count

REQ-2: Missed alarm count update check against notification settings of the emergency contact.

REQ-3: If missed alarm count is greater than or equal to the contact notification trigger, notification settings are checked and notification is released to the required system.

REQ-4: If the count is less than the contacts notification trigger(s) nothing is done.

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>