# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# CSC 431 Medication Monitor Software Requirements Specification (SRS)

**Team #1**

|  |  |
| --- | --- |
| Grace (Tianjiao) Gu | Developer |
| Isaac Attuah | Developer |
| Jin Curia | Scrum Master |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1 | 2/22/21 | Jin Curia,  Isaac Attuah,  Grace Gu | * First draft |
| 2 | 3/3/21 | Jin Curia,  Isaac Attuah,  Grace Gu | * Second draft |
| 3 | 3/9/21 | Isaac Attuah,  Grace Gu,  Jin Curia | * Added UML class diagram. * Changed names of functional & non-functional requirements. |

# Table of Contents

1. [**CSC 431  
     
   Medication Monitor  
     
     
   Software Requirements Specification (SRS)**](#_heading=h.zf3o7lrbhi7b) **1**
2. [**Version History**](#_heading=h.1fob9te) **1**
3. [**Table of Contents**](#_heading=h.3znysh7) **2**
4. [**Table of Tables**](#_heading=h.1tl4t44txhru) **4**
5. [**Table of Figures**](#_heading=h.tyjcwt) **5**
6. [**System Requirements**](#_heading=h.3dy6vkm) **7**
7. [Functional Requirements](#_heading=h.1t3h5sf) 7
8. [Registering an account](#_heading=h.4d34og8) 7
9. [Logging into an account](#_heading=h.o25t29f79hyu) 7
10. [Updating personal information](#_heading=h.9h5vll2vkxn8) 7
11. [Register medications](#_heading=h.gebzs3yg33zn) 8
12. [View medication information](#_heading=h.64vvvkrtn1m0) 8
13. [Count medications](#_heading=h.ddkotsoo8hsj) 9
14. [Identify medications](#_heading=h.f5m602lu6jzm) 9
15. Set alerts 9
16. [Non-Functional Requirements](#_heading=h.2s8eyo1) 10
17. [Secure patient information](#_heading=h.17dp8vu) 10
18. [Fast image segmentation](#_heading=h.6f3jsudc4ee7) 10
19. [Alert patient without delays](#_heading=h.m53nq3upz2jk) 10
20. [Requirement Title](#_heading=h.3rdcrjn) 11
21. [System Constraints](#_heading=h.5g35vt8srek3) 11
22. [Tool Constraints](#_heading=h.26in1rg) 11
23. [Mobile Application Framework Constraint](#_heading=h.lnxbz9) 11
24. [Language Constraints](#_heading=h.35nkun2) 11
25. React Framework 11
26. [Platform Constraints](#_heading=h.44sinio) 11
27. [Mobile Phone Service Platform](#_heading=h.2jxsxqh) 11
28. [Hardware Constraints](#_heading=h.z337ya) 12
29. [Requirement Title](#_heading=h.3j2qqm3) 12
30. [Network Constraints](#_heading=h.krcyvdix8l3o) 13
31. [Requirement Title](#_heading=h.4i7ojhp) 13
32. [Deployment Constraints](#_heading=h.2xcytpi) 13
33. [Requirement Title](#_heading=h.1ci93xb) 13
34. [Transition & Support Constraints](#_heading=h.3whwml4) 13
35. [Requirement Title](#_heading=h.2bn6wsx) 13
36. [Budget & Schedule Constraints](#_heading=h.qsh70q) 13
37. [Requirement Title](#_heading=h.3as4poj) 13
38. [Miscellaneous Constraints](#_heading=h.ewond11suupw) 14
39. [Requirement Title](#_heading=h.49x2ik5) 14
40. [Requirements Modeling](#_heading=h.2p2csry) 15
41. [User and System](#_heading=h.147n2zr) 15
42. [Class Diagram](#_heading=h.8a5ge4atoey7) 16
43. [Evolutionary Requirements](#_heading=h.o3l1ev30yz2f) 17
44. [Functional Requirements](#_heading=h.23ckvvd) 17
45. [Requirement Title](#_heading=h.ihv636) 17
46. [Non-Functional Requirements](#_heading=h.32hioqz) 17
47. [Requirement Title](#_heading=h.1hmsyys) 17

# Table of Tables

<Generate table here>

# Table of Figures

<Generate table here>

# System Requirements

#### Functional Requirements

*< List all functional requirements in the following example format >*

##### Registering an account

|  |  |
| --- | --- |
| Title | Registering an account |
| Description | Users can register an account to store their information. |
| Priority | Mandatory: 0 |
| Precondition(s) | None. |
| Basic Flow | * User opens application * User clicks Create Account * User enters information into the account creation form * User exits application to verify their account |
| Postconditions(s) | User is sent a verification email or SMS. |
| Use Case Diagram | <Link or number, if present> |

##### Logging into an account

|  |  |
| --- | --- |
| Title | Logging into an account |
| Description | Users can log into existing accounts using the account’s registration information. |
| Priority | Mandatory: 0 |
| Precondition(s) | User account exists in the account database. |
| Basic Flow | * User opens application * User enters phone number or email associated to account * User presses login button * User sees their full name displayed and enters their password * User presses login button again |
| Postconditions(s) | User is directed to the user dashboard. |
| Use Case Diagram | <Link or number, if present> |

##### 

##### Updating personal information

|  |  |
| --- | --- |
| Title | Updating personal information |
| Description | New users and existing users can update their personal information manually or by using an automated camera smart-feature. |
| Priority | Mandatory: 0 |
| Precondition(s) | User is logged into an account and has internet access. |
| Basic Flow | * User views personal information on their dashboard, empty or non-empty. * User presses update information button. * The system opens their camera application. Else user enters information manually. * User presents their ID or prescription label to the application. * The application records the information. * User presses “confirm”. User is now back at dashboard. |
| Postconditions(s) | User personal information is updated or remains the same. |
| Use Case Diagram | <Link or number, if present> |

##### Register medications

|  |  |
| --- | --- |
| Title | Register medications medication |
| Description | Users can register their medications manually or by using an automated camera smart-feature. |
| Priority | Mandatory: 0 |
| Precondition(s) | User is logged into an account and has internet access. |
| Basic Flow | * User is viewing their dashboard. * User selects “Medications” . * User selects Add medication. * The system opens their camera application. * User shows medication label or the medication itself. * The system records this information. * User presses “confirm”. User returns to medication list. |
| Postconditions(s) | User’s medication list is updated. |
| Use Case Diagram | <Link or number, if present> |

##### View medication information

|  |  |
| --- | --- |
| Title | View medication information |
| Description | Users can view detailed information about their medication, dose, dose quantity, refill quantity, date filled, directions, side effects, drug interactions, all simultaneously. |
| Priority | Mandatory: 0 |
| Precondition(s) | * User is logged in. * User’s medication list is non-empty. |
| Basic Flow | * User is viewing their medication list. * User targets a specific medication and presses “view information” * The system returns a summary of their user-specific medication information. * User selects non-user-specific medication information. * The system displays a summary of the medication, interactions, and side effects, etc. |
| Postconditions(s) | None. |
| Use Case Diagram | <Link or number, if present> |

##### Count medications

|  |  |
| --- | --- |
| Title | Count medications |
| Description | Users can identify and count medications using their camera. |
| Priority | Mandatory: 0 |
| Precondition(s) | User is logged in and has internet access. |
| Basic Flow | * User selects the Identification/Counter tool from the Tools menu. * The system opens the camera. * User targets the medication collection with camera. * The system counts and returns a list of medication names, quantities, and total quantity. * User can select update medication quantity or exit. |
| Postconditions(s) | User can view their dose scheduler in its own pane on the User Dashboard. |
| Use Case Diagram | <Link or number, if present> |

##### Identify medications

|  |  |
| --- | --- |
| Title | Identify medications |
| Description | Users can identify unknown medication(s) using their camera. |
| Priority | Mandatory: 0 |
| Precondition(s) | User is logged in and has internet access. |
| Basic Flow | * User opens Pill Identification tool. * The system opens the camera. * User targets the collection. * The system |
| Postconditions(s) | None. |
| Use Case Diagram | <Link or number, if present> |

##### Set alerts

|  |  |
| --- | --- |
| Title | Dose reminder |
| Description | Users can schedule their doses and receive reminders as alerts from the application. |
| Priority | Mandatory: 0 |
| Precondition(s) | User has a non-empty medication list. |
| Basic Flow | * User opens dose schedule. * System returns summary of schedule and pending alerts. * User selects adjust schedule. * System returns a menu for the user to include / exclude medications, and pre-constructed schedule suggestions. |
| Postconditions(s) | Schedule is updated in the system with the changes. |
| Use Case Diagram | <Link or number, if present> |

#### Non-Functional Requirements

*< List all non-functional requirements in the following example format >*

##### Secure patient information

|  |  |
| --- | --- |
| Title | Secure patient information |
| Description | Personal and medication information for users should be encrypted due to the sensitivity of the data. |
| Priority | Mandatory: 0 |
| Applicable FR(s) | * Registering/Logging in account. * Update personal information |

##### Fast image segmentation

|  |  |
| --- | --- |
| Title | Fast image segmentation |
| Description | The user’s device should complete image segmentation within 5 seconds. Fast is defined as between 0-5 seconds. |
| Priority | Important: 1 |
| Applicable FR(s) | * Smart-count/Smart-identifier * Update personal information |

##### Alert patient without delays

|  |  |
| --- | --- |
| Title | Alert patient without delays |
| Description | The dose alerts should be in sync with the user device system clock. No time delays at all. |
| Priority | Mandatory: 0 |
| Applicable FR(s) | * Dose reminder |

### System Constraints

#### Tool Constraints

**References:**

* https://nodejs.org/en/

##### Mobile Application Framework Constraint

|  |  |
| --- | --- |
| Title | Mobile Application Framework Constraint |
| Description | Our application will use Node.js for launching both the frontend and backend of the app. |
| Priority | Mandatory: 0 |

#### Language Constraints

**References:**

* <https://reactnative.dev/>

##### React Framework

|  |  |
| --- | --- |
| Title | React Native Framework |
| Description | JavaScript is required for using both Node.js and React Native. The frontend and backend will both be compiled as JavaScript. |
| Priority | Mandatory: 0 |

#### Platform Constraints

*< List all platform constraints in the following example format >*

##### Mobile Phone Service Platform

|  |  |
| --- | --- |
| Title | Mobile Phone Service Platform |
| Description | Node.js and React Native combined allows us to deploy the application for iOS and Android platforms simultaneously. |
| Priority | Mandatory: 0 |

#### Hardware Constraints

*< List all hardware constraints in the following example format >*

##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### Network Constraints

*< List all network constraints in the following example format >*

##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### Deployment Constraints

*< List all deployment constraints in the following example format >*

##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### Transition & Support Constraints

*< List all transition & support constraints in the following example format >*

##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### Budget & Schedule Constraints

*< List all budget & schedule constraints in the following example format >*

##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

#### Miscellaneous Constraints

*< List all miscellaneous constraints in the following example format >*

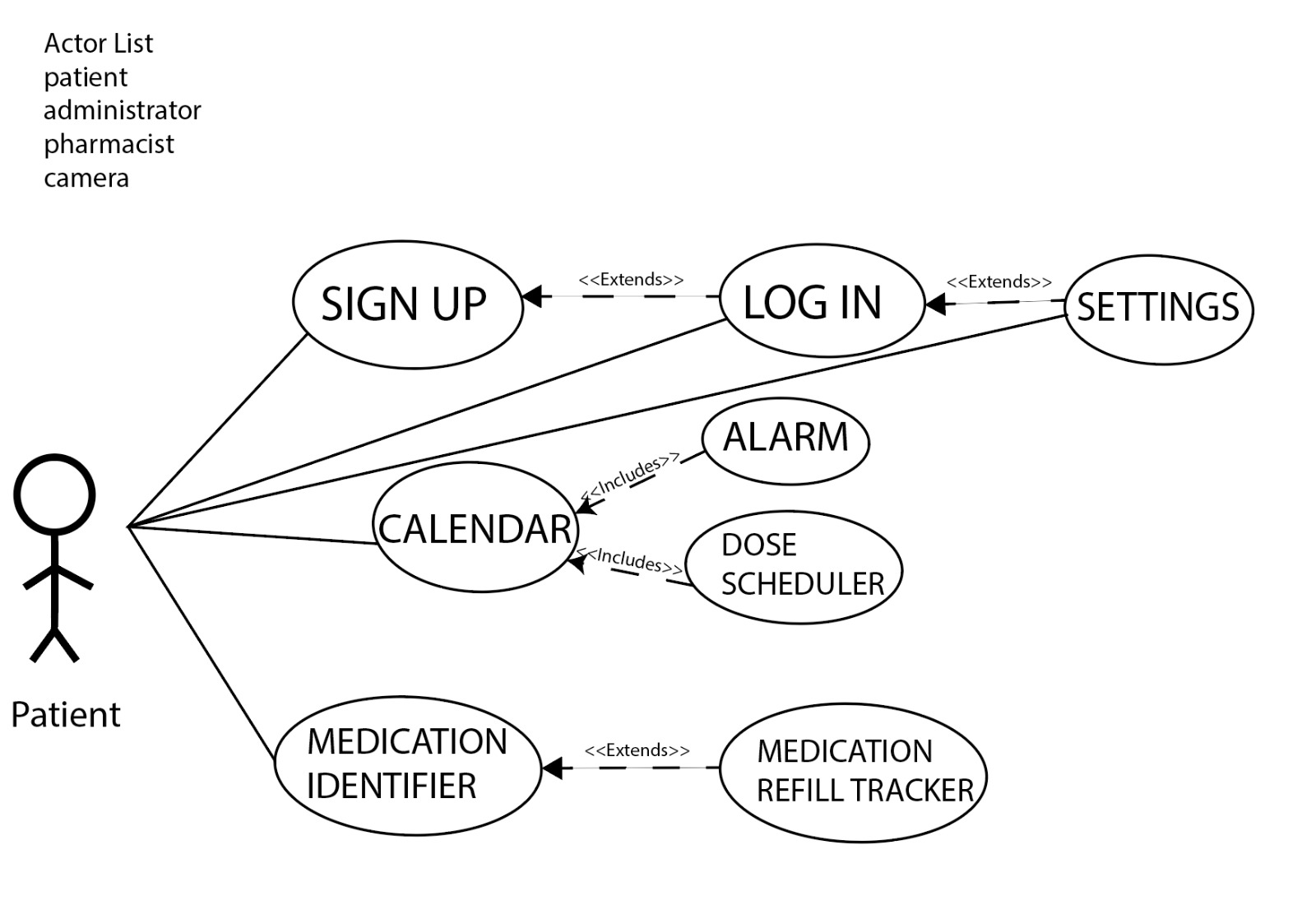
##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |

### Requirements Modeling

*< List all Use-case diagrams for the functional requirements in the following format>*

##### User and System



##### Class Diagram

### 

### Evolutionary Requirements

#### Functional Requirements

*< List all functional requirements in the following example format >*

##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |
| Precondition(s) | <What needs to happen before> |
| Postconditions(s) | <What happens as a result> |
| Use Case Diagram | <Link or number, if present> |

#### Non-Functional Requirements

*< List all non-functional requirements in the following example format >*

##### Requirement Title

|  |  |
| --- | --- |
| Title | <Insert title> |
| Description | <A one or two sentence description> |
| Priority | <Priority from 0 (highest) – 5 (lowest)> |
| Applicable FR(s) | <Which functional requirement(s) is this applicable to?> |

# 