New York University Abu Dhabi
CS-UH 2012: Software Engineering

Abdullah Ahmed, Rhea Braithwaite, Varvara Snitko, Yonathan Wagaye

Software Requirements Specification
for
MED+

Version: 1 **Date: 02/25/2023**

Table of Contents

3.6.4 Maintainability
3.6.5 Portability

4. Change Management Process

5. Document Approvals

6. Supporting Information

1. Introduction

1.1 Purpose

The purpose of this System Requirement Specification document is to provide a detailed description of the healthcare mobile application. It will describe the purpose and characteristics of the application, its interfaces, how the application will perform, the limitations under which it has to operate, and how the system will interact with external systems. This document is intended for customers, designers, and developers of the application.

1.2 Scope

The healthcare mobile application will be named MED+. The goal of this application is to ease the search of pharmaceutical products in the UAE region. The system will allow users to find the nearest pharmacy with the best price for a specified product at any time instead of going across pharmacies to check the availability and compare prices. The application will improve the access to healthcare products for its users by reducing healthcare costs and time spent on pharmaceutical purchases for individuals and families.

1.3 Definitions, Acronyms, and Abbreviations

AES- Advanced Encryption Standard
RSA- Rivest-Shamir-Adleman
API - Application Programming Interface
DB - Database
FDA- Food and Drug Administration
GPS - Global Positioning System
HIPAA- Health Insurance Portability and Accountability Act
HTTPS- HyperText Transfer Protocol
SSL- Secure Sockets Layer
TLS- Transport Layer Security
UI - User Interface
UAE - United Arab Emirates

1.4 References

Postel, J. (1981, September 1). *Internet protocol*. RFC Editor. Retrieved February 24, 2023, from https://www.rfc-editor.org/rfc/rfc791

Rescorla, E., & Schiffman, A. M. (1999, August 1). *RFC FT-IETF-WTS-shttp: The Secure Hypertext Transfer Protocol*. IETF Datatracker. Retrieved March 1, 2023, from https://datatracker.ietf.org/doc/html/rfc2660#:~:text=RFC%202660%20%2D%20The%20Secure%20HyperText%20Transfer%20Protocol

Federal Law No (15) of 2020 on Consumer Protection (2020). Ministry of Economy. https://www.moec.gov.ae/en/federal-law-no-15-of-2020-on-consumer-protection?p_l_back_url=%2Fen%2Fsearch-results%3Fq%3DFuture%2520Economy%26start%3D9

Ministerial Resolution No. 430 of 2007 Regulating Health Advertisement (2007). Ministry of Health and Prevention. Retrieved March 1, 2023. https://www.dha.gov.ae/uploads/042022/Standards%20for%20Medical%20Advertisement%20Content%20in%20Social%20Media2022433965.pdf.

Federal Law No (45) of 2021 on Protection of Personal Data (2021). Ministry of Artificial Intelligence. Retrieved March 1, 2023. https://ai.gov.ae/personal-data-protection-law/

1.5 Overview

Customers can refer to section 1 and 2 for a better understanding of the overall structure of the application. Developers of the system should go through section 3 to delve into technical details. UI designers may look through section 2 and 3.

2. The Overall Description

2.1 Product Perspective

MED+ is a newly constructed autonomous application, which can interact with external services. Although there are many healthcare solutions presented in the market, only a few have functionalities to compare prices for medications at local pharmacies. For example, GoodRx, LowestMed, and Medadvisor are available and applicable in the United States, Australia, and Portugal. So far, none of the existing applications focused on comparing prices for pharmaceutical products across different pharmacies are available in the UAE region. MED+ aims to fill in this gap.

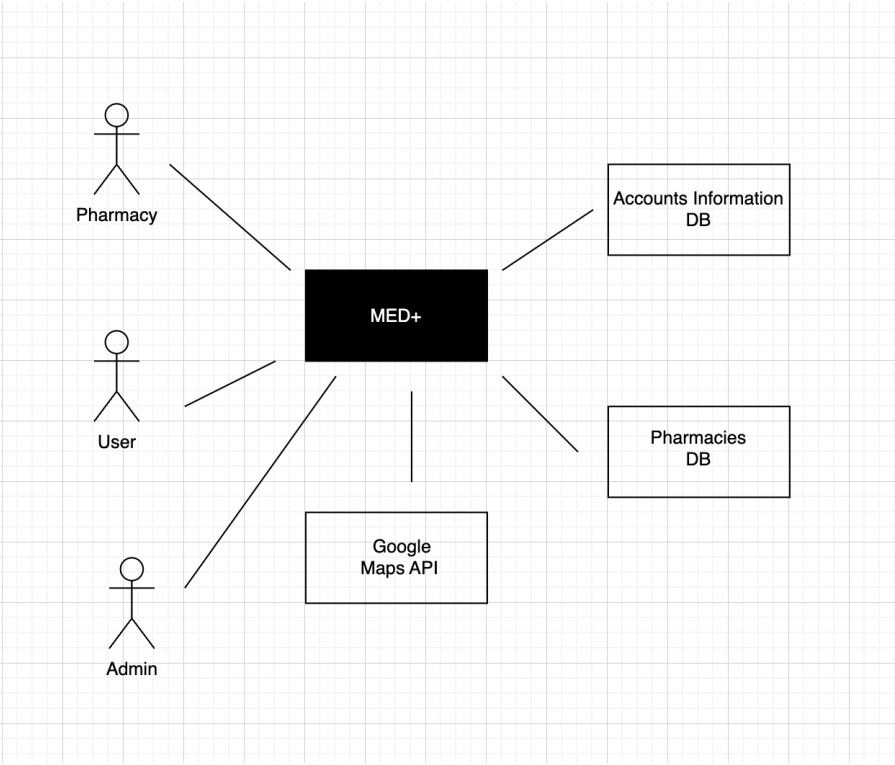

Fig 2.1 product perspective block diagram

2.1.1 System Interfaces

The system will incorporate the Google Maps API for location-based services. This will help the system to determine the nearest pharmacy.

The application shall interface with the existing pharmacies' database systems in order to get prices of products, advertisements, and coupon codes.

2.1.2 Interfaces

MED+ is a mobile application that requires installation prior to usage as well as internet access to display content and facilitate user interaction. Once the user opens the application, the authentication screen appears and the login screen will be shown. After the authentication check, the users can access the functionalities of the application. The home screen will be shown to the user by default, where the search bar will be located for a quick search of medication, as well as buttons with links to the other features of the application on the button of the screen. After the user enters the name of the desired product, the application will display a list of results that match their search query, listing them according to prices and relative location.

2.1.3 Hardware Interfaces

Since MED+ is a software application it does not require a hardware interface, however some features of the application may require interaction with hardware components of the user's mobile device such as the GPS.

2.1.4 Software Interfaces

The application itself would be designed with the help of Flutter framework for the front-end part and Dart Frog framework for the back-end development. Since the system would integrate with existing pharmacies' databases, PostgreSQL along with Prisma shall be used for database management. Heroku will be used to deploy, manage, and scale the mobile application.

Google Maps API would serve as a base building block for the retrieval of geographic-related data. It will help to identify the location of the user in order to choose the best pharmacy available based on the distance between them.

2.1.5 Communications Interfaces

The following protocols would help all the systems used for the platform development to communicate between each other:

- RFC 791: Internet Protocol[1]
- RFC 2660 - The Secure HyperText Transfer Protocol - HTTPS[2]

2.1.6 Memory Constraints

Cloud hosting memory will be allocated according to the expected amount of users and pharmacies with initial release. We are predicting, initially, at least 50,000 installations and 200 pharmacies to join. Therefore, 100 GB of cloud data should serve our purpose. If needed, more cloud storage can be bought.

In terms of user space, focus groups have revealed that the average smartphone in the UAE has at least 1GB of RAM, and on average, 2GB. Therefore, the mobile application should run smoothly with 600MB of RAM available to its disposal.

2.1.7 Operations

Customer User Operations:

- The user should be able to create an account.
- The user should be able to login to the system using the proper credentials.
- The user should be able to search for pharmaceutical products.
- The user should be able to narrow/filter the search using brand name and price filters.

Pharmacy User Operations

- Pharmacies should be able to create an account and register to the system
- Pharmacies should be able to upload their database on the system
- Phramices should be able to update their database (stock of product, adding/deleting new products).
- Pharmacies should be able to upload their advertisements and coupon codes.

Admin User

- An admin user should be able to view all the products from all pharmacies
- An admin user should be able to remove pharmacies from the system.
- An admin user should be able to get the user activity including frequent products bought and searched

Backup and recovery operations:

- The application must take regular backups of data every six hours by updating changes made to the system.
- The application must provide recovery operations in case any part of the system crashes. The application should remain accessible 24/7.

2.1.8 Site Adaptation Requirements

- The system should support multiple languages, including the primary languages spoken by the user base. This will make the system more accessible and user-friendly for non-native English speakers
- The system should be accessible to users with disabilities
- The system should comply with local regulations and guidelines, including those related to data privacy, medical product labeling, and advertising
- The system should consider the unique needs and preferences of users in different geographical locations.

2.2 Product Functions

The system should have the following functionalities:

- The system should allow any user (pharmacies and customers) to create accounts and register to the system
- Pharmacies must upload their license in order to register on the application.
- The system should allow pharmacies to upload and edit their databases
- The system should allow pharmacies to upload coupon codes and advertisements.
- The system should allow customers should be able to filter products by brands
- The system should allow customers to allow filter products using price
- The system should give the optimal recommendation such that the product has a low cost in terms of price and transportation
- The system should provide the users with pharmacy-specific coupons
- The system should allow customers to add reviews on pharmacies

2.3 User Characteristics

There are three categories of users for this system namely Customers, Pharmacies, and Administrators.

The potential customer users of MED+ are:

- Patients who need to purchase medication for themselves or individuals who need to buy medication for their family members. They could use the application to find the best prices for medicines in nearby pharmacies.

- Caregivers who take care of elderly or disabled family members or patients. They could use the application to find medicines on their behalf easily. Furthermore, for these individuals, the distance to the pharmacy is the major factor since they can not leave their patients for a long time

- Medical professionals who prescribe medications to their patients. They could use the application to search for the best prices of prescribed medicines and recommend them to their patients, thus helping them save on their medication costs. The application can be also useful if the patient is in urgent need of a specified product that can be found nearby the hospital.
- Pharmacy professionals/representatives who provide medical products to patients. They could use the application to compare the prices of medicines in different pharmacies and offer their patients the best possible options.

The user body is very diverse and comes from a variety of backgrounds. Not all of them might be familiar with technologies and their functionalities. A portion of users might have various disabilities.

2.4 Constraints

- The system must ensure that the information is consistent with the FDA-approved labeling and any other relevant FDA regulations
- The system must comply with the HIPAA regulations, which govern the use and disclosure of protected health information
- The system must adhere to local ethical and legal guidelines, including those related to informed consent, confidentiality, and data privacy
- The system must provide accurate information of every medical product; inaccuracy is intolerable as medical
- The system must be secure and user data must be protected from any kind of breach
- The system must provide up-to-date information about medical products, including any changes or updates to their use, indications, or dosage
- They system must remove pharmacies who failed to update their product information
- The system should provide reliable information on each product

2.5 Assumptions and Dependencies

The system assumes that pharmacies will provide accurate information on medical products. However, in case of misinformation the system uses customer reviews and ratings to check the accuracy of the information. Administrators can remove pharmacies with low ratings.
The system assumes every user has an iOS or Android smartphone. Furthermore, the system is dependent on external data sources provided by pharmacies for each product.

2.6 Apportioning of Requirements.

Potential features that can be implemented in future releases include

- Voice-assisted search which can be very useful for people of determination
- A social media feature that includes news on different medical products with feedback on different medical products
- Provide links for customers to order medical products online
- Alternative products suggestion feature. The system can suggest to the user an alternative product that is close to the product the user is looking for but with less cost
- Information about the accessibility of the pharmacy
- Bookmarking favorite pharmacies
- User account feature that allows users to save their preferred pharmacies and products
- Users can save their personal information, as well as their preferred pharmacies and medications

3. Specific Requirements

3.1 External Interfaces

This contains a detailed description of all inputs into and outputs from the software system to facilitate the developers building the application.

1. **Name of item:** full_name
    a. **Description:** Name of user
    b. **Source:** User
    c. **Valid range:** Character String of maximum length 25
    d. **Relationship to other inputs/outputs:** Used for account management, welcoming user to app
    e. **Data format:** String
2. **Name of item:** email
    a. **Description:** Email of user
    b. **Source:** Users, Pharmacies
    c. **Valid range:** alphanumeric string
    d. **Relationship to other inputs/outputs:** Used for account management
    e. **Data format:** Alphanumeric string including special characters
3. **Name of item:** password
    a. **Description:** Password for (username, password) pair
    b. **Source:** User, Pharmacy
    c. **Valid range:** Alphanumeric String of minimum length 8, at least one capital letter, at least one small cased letter, at least one special character, at least one number
    d. **Relationship to other inputs/outputs:** Account management
    e. **Data format:** Salted and encrypted alphanumeric string including special characters
4. **Name of item:** pharmacy_database
    a. **Description:** Database of the items and prices of a pharmacy
    b. **Source:** Pharmacy
    c. **Valid range:** Maximum 5GB.
    d. **Relationship to other inputs/outputs:** finds the best result when searching for item
    e. **Data format:** Dataset of at least the items and prices
5. **Name of item:** item_name
    a. **Description:** Item that the user is searching
    b. **Source:** User
    c. **Valid range:** Alphanumeric string of maximum length 25
    d. **Relationship to other inputs/outputs:** Used to search for item in dataset
    e. **Data format:** Alphanumeric string

6. **Name of item:** product_description
    a. **Description:** Description of the product
    b. **Source:** Pharmacy
    c. **Valid range:** Alphanumeric string of maximum length 100
    d. **Relationship to other inputs/outputs:** Used to display product to user
    e. **Data format:** Alphanumeric string

7. **Name of item:** current_location
    a. **Description:** Current location of the user using the application
    b. **Source:** GPS in user's smartphone
    c. **Valid range:** GPS coordinates within the UAE
    d. **Relationship to other inputs/outputs:** Used to find best result of item being searched
    e. **Data format:** Array of numbers (GPS coordinates)
8. **Name of item:** list_of_pharmacies
    a. **Description:** List of pharmacy names
    b. **Source:** Google Maps API
    c. **Valid range:** maximum number of elements in list: 500
    d. **Relationship to other inputs/outputs:** Used to find best result of item
    e. **Data format:** array of string

9. **Name of item:** list_of_prices
    a. **Description:** List of prices of a given/specific item across pharmacies
    b. **Source:** Collected through the range of different pharmacy databases
    c. **Valid range:** Double of value at least 0.0
    d. **Relationship to other inputs/outputs:** Used to find best result of item being searched
    e. **Data format:** Array of double
10. **Name of item:** map_of_result
    a. **Description:** map of <pharmacy_name, price>. This holds all the pharmacy names and their respective prices for a given item
    b. **Source:** System
    c. **Valid range:** maximum number of pairs in map: 500
    d. **Relationship to other inputs/outputs:** Facilitates showing best and ranked results to user
    e. **Data format:** map <string, double>
11. **Name of item:** brand_name
    a. **Description:** Argument to be used to filter the search results
    b. **Source:** User
    c. **Valid range:** Alphanumeric string of maximum length 25

    d. **Relationship to other inputs/outputs:** Used to filter result
    e. **Data format:** Alphanumeric string
12. **Name of item:** filter_price_min
    a. **Description:** Argument to be used to filter the search results
    b. **Source:** User
    c. **Valid range:** double, default value = minimum price in list_of_prices
    d. **Relationship to other inputs/outputs:** Used to filter result
    e. **Data format:** double
13. **Name of item:** filter_price_max
    a. **Description:** Argument to be used to filter the search results
    b. **Source:** User
    c. **Valid range:** double, default value = maximum price in list_of_prices
    d. **Relationship to other inputs/outputs:** Used to filter result
    e. **Data format:** double
14. **Name of item:** coupon
    a. **Description:** Coupon that allows users to get discounts upon their purchase
    b. **Source:** Pharmacy, randomly generated
    c. **Valid range:** alphanumeric string of maximum length 10
    d. **Relationship to other inputs/outputs:** Output to user
    e. **Data format:** alphanumeric string
15. **Name of item:** pharmacy_advertisement
    a. **Description:** Advertisements to be shown on the app
    b. **Source:** Pharmacy
    c. **Valid range:** Picture of maximum size 5MB
    d. **Relationship to other inputs/outputs:** Output - Shown on user interface throughout the app
    e. **Data format:** PNG, JPEG
16. **Name of item:** pharmacy_name_database
    a. **Description:** The name of the database that is uploaded for each pharmacy
    b. **Source:** system
    c. **Data format:** String
17. **Name of item:** pharmacy_license
    a. **Description:** The practicing license of a pharmacy
    b. **Source:** pharmacy
    c. **Relationship to other inputs/outputs:** Used for registering pharmacies onto the system.
    d. **Data format:** pdf, image
18. **Name of item:** review

a. **Description:** Review to be added by a user for a pharmacy
b. **Source:** User
c. **Valid range:** Int (number of stars): between 1 and 5, boundaries inclusive. String of maximum length 500 characters.
d. **Relationship to other inputs/outputs:** Review added to the profile of pharmacy
e. **Data format:** Object of attributes: string (text of review), int (number of stars), date of post.

3.2 Functions

This section includes the functional requirements of the application to facilitate developers in implementing the requirements.

1. user_register(user_name, email, password)
    a. The system shall create an account for the user.
    b. Check name length
    c. Check email (ensure it is an email)
    d. Check password strength:
        i. Should include at least one small letter
        ii. Should include at least one capital letter
        iii. Should include at least one number
        iv. Should include at least one special character
    e. Add (username, password) pair in cloud database
        Ensure password is salted AND encrypted

2. user_login (email, password)
    a. The system should log the user into the application if the login credentials are correct
    b. Find email against database of emails
    c. Locate password, check if passwords match
    d. If passwords match, log in. Otherwise show error.

3. pharmacy_register(pharmacy_name, username, password, pharmacy_license)
    a. The system shall allow pharmacies to register on the application.
    b. Follow validation steps from 3.2.1
    c. Upload pharmacy_license
    d. Wait for approval from the administrator (up to 48 hours).
    e. Add pharmacy data to cloud database

4. pharmacy_login(email, password)

    a. The system should log pharmacy into the application if the login credentials are correct
    b. Check login credentials similar to 3.2.2

5. upload_database(pharmacy_database)
    a. The system shall allow pharmacies to upload their databases on the cloud database.
    b. Get name of pharmacy that is uploading the database
    c. Check if a database named pharmacy_name_database exists. If not, Create new database in cloud database named pharmacy_name_database
    d. Upload data to pharmacy_name_database in cloud database

6. upload_advertisement(pharmacy_advertisement)
    a. The system shall allow pharmacies to upload their advertisements on the application.
    b. Upload file to advertisements database on cloud database

7. search_maps(current_location)
    a. The system should be able to use Google Maps API to search for nearby pharmacies.
    b. Get a list of all pharmacies in a 5 Kilometer radius (list_of_pharmacies) using Google Maps API.

8. search_item(item_name, current_location)
    a. The system shall look-up for the item the user wants to purchase
    b. Edit list_of_pharmacies to only include pharmacies that are present in the application's cloud database
    c. Retrieve price of the item_name from each pharmacy in list_of_pharmacies and place in list_of_prices.
    d. Each list of item should have a link for a product page that contains the description, image and price of the product

9. sort_result(list_of_prices)
    a. The system shall sort the results in ranked order up to ten.
    b. Create a map of pharmacies and their respective prices for the item searched for
    c. Sort the map according to prices in ascending order
10. show_results(map_of_result)
    a. The system shall display the results according to their ranked value.
    b. Show the first entry of sorted map as the best option

c. System will ensure description of product is shown
d. System will display whether the pharmacy supports online purchases or not
e. Show the result of the results in increasing order of index (ranked results)

11. filter_result(brand_name, filter_price_min, filter_price_max)
    a. The system should allow users to filter the results according to their preferences.
    b. If brand_name is given, filter results so that only results including the given brand are shown. Otherwise, all brands are shown.
    c. If fitler_price_min is given, filter the results so that all prices are greater than or equal to filter_price_min. Otherwise, filter_price_min takes on a default value of the least price in list_of_prices
    d. If filter_price_max is given, filter the results so that all the prices are less than or equal to filter_price_max. Otherwise, filter_prie_max takes on a default value of the maximum price in list_of_prices.

12. generate_coupon(pharmacy_name)
    a. The system should generate discount coupons for the users to be applied on checkout at the pharmacy.
    b. Check if pharmacy_name is in the cloud database of all pharmacies registered on the app. If no, throw an error
    c. Lookup coupon code in pharmacy database. They have provided us with coupons to be used.
    d. Return the coupon code. Return “no coupon available!” if no coupon is provided.
13. add_review(review)
    a. The system should ensure the input is within valid range as defined in subsection 3.1. If invalid, return an error message.
    b. If valid, add review to the pharmacy database of the relevant pharmacy.

14. get_advertisement()
    a. The system should be able to retrieve advertisement image files from cloud database
    b. Return any advertisement from the advertisement database.

15. show_interface()
    a. The system shall show the UI to the user and ensure smooth operations.
    b. Show UI to user

3.3 Performance Requirements

This subsection specifies the static and dynamic numerical requirements placed on the mobile application.

Static numerical requirements:

1. The system shall support a minimum of 10,000 users simultaneously.
2. The system shall support maintaining data of up to 300 pharmacies, 100,000 pharmaceutical products and 1,000,000 users.
3. The system shall support a minimum of 100 Terabytes of data for the system.
4. All requests shall be completed within 10 seconds.

Dynamic numerical requirements:

- The system shall support a minimum of 1000 logins per minute.
- The system shall support a minimum of 500 searches per minute.
- The system shall support a minimum of 10 databases being uploaded per hour.

3.4 Logical Database Requirements

- The system shall store user account information in cloud storage
    - Database outline for Prisma implementation:
        - Name : String
        - Email : String
        - Encrypted + salted password : String
        - Recent searches : List of String
- The system shall store the medical products and prices of each pharmacy in cloud storage.
    - Database outline for Prisma implementation:
        - Pharmacy name : String
        - Pharmacy ratings : Double
        - Online shopping supported? : Boolean
        - Email : String
        - Encrypted + salted password : String
        - Product : String
        - Product Description: String
        - Price : Double
        - Coupon : String
        - Advertisement : Image
        - License : Image

3.5 Design Constraints

Specify design constraints that can be imposed by other standards, hardware limitations, etc.

- The provided prices must be in Dirhams, given that the current pharmacies will be based in the UAE.
- The system must operate on both Android and iOS operating systems, which are predominantly on the mobile devices of the user-base.
- The system must eventually support Arabic, the official language of the UAE.

3.5.1 Standards Compliance

- The system shall not collect user data for marketing, in compliance with the UAE Federal Law No. 15 of 2020 on Consumer Protection[3].
- The system shall display the true and balanced prices of medical items as provided by the pharmacies in compliance with the UAE Ministerial Resolution No. 430 of 2007[4] Regulating Health Advertisement.
- The system shall not utilize a user's personal data without their explicit consent in compliance with the UAE Personal Protection Data Law[5].

3.6 Software System Attributes

3.6.1 Reliability

- The system shall perform as required to return to users the list of medicines that are closest to them and of lowest cost.
- The system will not crash on invalid data entry.

3.6.2 Availability

- The system shall be available to users 24 hours a day, 7 days a week.
- If there is a system failure, the redundant system shall resume operations within 15 minutes.

3.6.3 Security

- User account information files shall also be encrypted to ensure protection from unauthorized access.

- The system should use encryption algorithms to protect all sensitive data in transit and at rest. This includes using SSL/TLS protocols for secure communication between the app and the server, as well as encryption algorithms such as AES or RSA for storing data on the serve
- The system should use a data integrity mechanism to ensure the integrity of data in transit and at rest
- The system should receive regular security updates to address any vulnerabilities or security issues that may arise. This includes updating the app
- The system should have a strong authentication mechanism in place to verify the identity of users before granting them access to sensitive data
- Ensure all data overflows are dealt with accordingly.
- Ensure passwords are regularly updated by users.

3.6.4 Maintainability

The system must follow object oriented programming principles to ensure maintainability.

1. Modularity: System should be broken down into well-defined modules.
    a. Each module should be able to be tested independently.
    b. Each module can be updated without affecting the entire system.
    c. All modules should be able to integrate with the system at large.
2. Readability: All code should be easy to read with comments.
3. Documentation: The system should be thoroughly documented.

3.6.5 Portability

- The system shall be developed using Flutter, which will ensure its ability to run on both Android and iOS devices.

ID	Characteristic	H/M/L	1	2	3	4	5	6	7	8	9	10	11	12
1	Correctness	H												
2	Efficiency	H												
3	Flexibility	M												
4	Integrity/Security	H												
5	Interoperability	H												
6	Maintainability	M												
7	Portability	H												
8	Reliability	H												
9	Reusability	M												
10	Testability	H												
11	Usability	H												

12	Availability	H													

Definitions of the qualities used above:

- Correctness - extent to which program satisfies specifications, fulfills user's mission objectives
- Efficiency - amount of computing resources and code required to perform function
- Flexibility - effort needed to modify operational program
- Interoperability - effort needed to couple one system with another
- Reliability - extent to which program performs with required precision
- Reusability - extent to which it can be reused in another application
- Testability - effort needed to test to ensure performs as intended
- Usability - effort required to learn, operate, prepare input, and interpret output

4. Change Management Process

- Any proposed changes to the project specifications and requirements shall be first submitted via a written document to request the change.
- This document shall then be reviewed, and the change discussed with team members.
- Only when a majority of team members agree on the change shall it be implemented.
- In the event of a change, the SRS document shall be updated, and the date change noted in the document.
- It will be possible for users to provide suggestions and recommendations to the team. These shall be submitted to the app email, suggestions@medplus.com.
- A similar approval process will take place as above, requiring an initial review and then a majority agreement by the team for the change to be implemented.

5. Document Approvals

Identify the approvers of the SRS document. Approver name, signature, and date should be used.

Approver Name	Title	Signature	Date
Abdullah Ahmed	Developer/Client	Abdullah	2 March, 2023
Rhea Braithwaite	Developer/Client	Rhea	2 March, 2023
Varvara Snitko	Developer/Client	Varvara	2 March, 2023
Yonathan Wagaye	Developer/Client	Yonathan	2 March, 2023

6. Supporting Information

Along with the SRS, a requirement gathering document has also been produced. Please refer to the requirement gathering document for supporting information.