Software Requirements Specification

for

Healthy

Version 1.0 approved

Prepared by

Sanjay Shetty (16IT136),
Srinag Rao (16IT141),
Shreyas Seshadri (16IT135),
Ram Aditya (16IT131)

NITK

17 January 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Initial SRS	17-1-19	Initial draft	1.0
SRS1	31-1-19	Added functional requirements	1.1
SRS2	20-2-19	Added requirements, Finalized interfaces	1.2

1. Introduction

1.1 Purpose

This SRS describes the first version of Healthy, an all-in-one digital medical solution. The SRS describes software requirements of the two apps in our system: the first version of patient app and the first version of doctor app.

1.2 Document Conventions

The following are the document typographic conventions used throughout this document:

- PV: Patient's View
- DV: Doctor's View
- ASV: Ambulance Service View
- User: any actor mentioned above
- Priority: 0 is of least priority, 10 is of highest priority

1.3 Intended Audience and Reading Suggestions

This SRS is intended for developers and project managers. It contains the high-level details necessary for the project managers to guide developers. Certain technical details like schema of objects are present to guide developers. It can additionally be used by marketing staff to generate a feature list. Developers focused material is under the System Features heading. Marketing staff are advised to read about product scope.

1.4 Product Scope

The product is an all in one system that caters to the general needs of a user with respect to healthcare like availing ambulances, issue of prescription by doctors, viewing prescription and chatting with doctors. The product to move smaller healthcare establishment go paperless. Most clinics do not have enterprise solutions like large organisations to manage patient information and have to maintain paper records. This product aims to counter this with mobile applications for each actor.

1.5 References

1. UI design principles: https://material.io

2. Overall Description

2.1 Product Perspective

This product is a new venture. We are starting from scratch. The product is a collection of mobile applications. The product consists of a PV, DV and ASV.

2.2 Product Functions

- Register: The user registers their information to our database through the interface.
- Login: The user logs in to the application.
- Call ambulance: The user can avail an ambulance.
- Doctor Searching: The patient can search for the right doctor pertaining to their illness.
- Doctor chat: The patient can chat with chosen doctor.
- Digital Prescription Issuing: The doctor uploads the prescription of the patient after an appointment.
- Digital Prescription Viewing: The patient views the uploaded prescription.

2.3 User Classes and Characteristics

2.3.1 Patient

The Patient actor is any generic user who wishes to avail the services of the application for himself/herself or on behalf of someone else. The technical expertise of this Actor must include basic capability to use a touch interface based Android app on a mobile device. Actor is expected to be proficient in the use of 'English'.
The actor will be one of the most frequent user of the app.
The use cases of the Patient are:

- Book Appointment
- Receiving Prescription
- Chat with the doctor

2.3.2 Doctor

This actor of the application is a certified doctor and will have to be registered to our database through the interface.
The use cases of the doctors are :

- Accepting appointments
- Uploading prescriptions
- Chatting with the patient regarding the illness.

2.3.4 Ambulance service actor:

This actor is any ambulance service that we choose to send information about location to call an ambulance.
The use cases of ambulance service actor are:

- Processing information given by ambulance booking done by any user of the application.
- Sending ambulance to the corresponding location.

2.4 Operating Environment

The application is build using React Native and therefore ought to run on both iOS and Android devices. Due to technological restrictions we will be testing the application only on Android devices above Oreo. If we can procure more physical devices, we will update this, else we are limited to testing on emulators. Android devices should have Google Play-store installed and iOS devices should have the App Store installed.

2.5 Design and Implementation Constraints

The following are the constraints on the design and implementation:

Regulatory Policies:

- Patient-doctor confidentiality policies.
- License restrictions of Doctor.
- Policies regulating Accreditation to Hospitals. The app must allow doctors from accredited hospitals only.

Hardware Limitation:

- The application cannot be tested on windows phones.
- The app is restricted to mobile phones only.
- Only touch and text based input/output is provided.

Software Limitation:

- The app runs only on Android and iOS.
- The minimum RAM support required is 1GB.
- The minimum Android version 6.0.
- A SQL based DBMS must be used due to structured natured of data that will be received.
- The app must be Cloud-deployable using AWS.

2.6 User Documentation

TBD

2.7 Assumptions and Dependencies

2.7.1 Assumptions

It is assumed that:

- The app will use '108' as the default ambulance number as the App is going to be launched primarily in India.
- The language of communication is English.
- There is a stable network connectivity through GSM.
- There is a stable internet communication available to all Patients and Doctors at a minimum bandwidth of 2G.

Dependencies:

- Google Translate API.
- Google Maps API.
- OAuth module for authentication.
- API to get location data of the Mobile.

3. External Interface Requirements

3.1 User Interfaces

The following are the Interfaces based on the actors of the application:

1. Patient View (PV):
   This is the view of the application for the patient actor:
   - Register page: The patient registers his information into the database through this interface
   - Login Page: After registration the user logs in to the application
   - Doctor searching: The patient can search doctors based on their illness
   - Doctor chatting: The patient can chat with the chosen doctor with this interface.
   - Prescription viewing: The patient can view the prescription uploaded by the doctor after the appointment.

2. Doctor's View (DV):
   This is the view of the application for the doctor actor:
   - Register Page: The doctor registers to our database through the interface.
   - Login Page: The doctor logs in to the application after registration.
   - Prescription Uploading: The doctor uploads the prescription of the patient after the appointment.
   - Chat with patient: The doctor can chat with the patient after accepting to do so the request to chat with the patient.

Both the patient and doctor can avail an ambulance using the ambulance avail button present in the home page of both views.

3.2 Hardware Interfaces

- The application is designed to run on mobile phones, hence appropriate screen sizes should be observed.
- The button to avail an ambulance is present in the bottom left corner of the screen to facilitate easy usage.
- Other components of the application will be present on the center of the screen and will be center justified.

3.3 Software Interfaces

The user will be abstracted from the backend part of the application.The interfaces that have connections to other software components are:

- Login/Register: The information obtained from these pages will be communicated to the Database.And authentication will be carried out.

- Chatting: The chatting component of both PV and DV should be facilitated through a server having a database.
- Ambulance availing: This feature has to communicate to any ambulance services with location information.

3.4 Communications Interfaces

The app will use the following communication functions:

- Email
- Phone
- SMS
- HTTPS GET, POST
- Web Socket
- Web Server: Node

Communication Standards:

- Hyper-Text Transfer Protocol with secure connection using SSL.
- HTTP 2.0+
- Simple Mail Transfer Protocol for Emails.
- Short message peer-to-peer protocol for SMS.

4. System Features

4.1 Login/Register

4.1.1 Description and Priority

It's the most basic and functionality required for unique identification of doctors and patient.The login page for doctor takes specific information required for doctors identification and for patient the basic information. This feature is of 10 priority as all other feature use the ID and session obtained from login. This feature is medium risk given several existing modules exist.

4.1.2 Stimulus/Response Sequences

User(patient/doctor) clicks on the register button. Then he is given a form layout where he can fill the required details. The verification is done if any error the the feedback is given. Then user can proceed to login page, enter his ID/password.

4.1.3 Functional Requirements

REQ-1: Patient should have basic identification data to register

REQ-2: Patient has to remember their username and password to login

REQ-3: The login page must be the default page to appear whenever the app Doctor/patient should use appropriate apps.

4.2 Ambulance booking

4.2.1 Description and Priority

This provides the user to book ambulances readily in case of need during emergency times such as, emergency involving others nearby or emergency involving oneself.The app automatically sends location information to the ambulance services in order to speed up the process. This feature is of 8 priority. It should be the available in the first release of the app as it is the spotlight feature. It is of low risk as a well defined protocol for ambulances already exists.

4.2.2 Stimulus/Response Sequences

When the need arises that the user has to avail an ambulance, the application allows the user to do so with a push of a button.The corresponding emergency button will be available in the home page of the application.Once clicked it calls an ambulance service by also sending the location.The user also has the option to communicate the location directly to the ambulance service upon calling.

4.2.3 Functional Requirement

REQ-1: The user needs to be logged in to the application
REQ-2: The user must allow access to his location.
REQ-3: The user must allow the application to make phone calls.

4.3 Doctor searching

4.3.1 Description and Priority

This feature satisfies the need to help patients search the right doctors taking into consideration, the speciality to which the illness belongs to, the gender of both the doctor and the patient, the availability of doctors .This app provides this feature by making use of the information obtained during registration, and a form which the user will have to fill out to search the doctor. This feature is of priority 9. One of the basic features that will attract users to our app.

4.3.2 Stimulus/Response Sequences

When the patient finds the need to search for a doctor, a simple interface containing the sequences required to obtain the information needed to search for the right doctor will be initiated on the click of a button.Patients will have to fill out a form in which basic questions about the illness is asked.The most relevant doctors chosen based on the above obtained information will be listed in the interface.The patient can then proceed to choose the right doctor based on their convenience.

4.3.3 Functional Requirements

REQ-1: The user needs to be logged in to the application.
REQ-2: The user would need to give information regarding the illness using interfaces like form.
REQ-3: Inorder for the sequence to be complete the user must select a doctor from the list of doctors provided based on his convenience.

4.4 Doctor chat

4.4.1 Description and Priority

Finding the right doctors sometimes becomes difficult sometimes. So it will be useful for patient if he is able to chat with the doctor before meeting the doctors. The app provides the option to chat with the doctor in specific specialisation. This feature is of 5 priority as it is not projected to be frequently used. It is of high risk as it has legal consequences.

4.4.2 Stimulus/Response Sequences

Patient have to login into the app and then click on chat with doctor option.Patient will be given a list of specialisation to choose.After choosing the specialisation patient is given list of doctors with their consultant fees.On sending the message the doctor will get the notification.

4.4.3 Functional Requirements

REQ-1: Patient should be registered user of the app.

REQ-2: Patient should make pre payment to chat with random doctor in particular specialisation.
REQ-3: From the network perspective the should be connection to internet.

4.5 Digital Prescription

Digital prescriptions are paperless prescriptions issued by doctors to patients in a safe and trackable way.

4.5.1 Description and Priority: A digital prescription is a paperless prescription created by a doctor on their device and assigned to a patient. The patient can access the prescription on their device and can use in a drug-store to claim their prescription. This feature has a priority of 7 and is high risk.

4.5.2 Stimulus/Response Sequences: The doctor enters the patient ID into the prescription view of the app to generate an empty prescription. The doctor proceeds to enter each drug name along with its dosage and additional comments if any. The app attempts to autocomplete drug names to assist the doctor.

4.5.3 Functional Requirements
REQ-1: A doctor has to be logged in to issue a prescription. This is because the prescription entry in the database is linked to the doctor ID got from login process.
REQ-2: The doctor should know the patient ID as the prescription is linked to the patient ID.
REQ-3: A patient has to be logged in to view prescriptions linked to their patient ID.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The applications are quite light in nature. No intense computation occurs on the devices themselves. Transactions between applications should be snappy as they correspond to real life interactions. For example the issuing of prescription by a doctor and patient receiving it on their device should happen instantaneously as there is a real world interaction behind the process. This improves the quality of service. Transactions like retrieving all prescriptions or patient details have lower priority compared to the prescription issuing or hailing ambulance.

5.2 Safety Requirements

The chat has to be secure to confer with doctor patient confidentiality regulations. The integrity of issued prescription must be maintained to avoid wrong drug being prescribed.

5.3 Security Requirements

User authentication is critical. Unauthorised access to the doctor's credentials will allow the attacker to issue illegal prescriptions. Unauthorised access to the patient's credentials will allow the attacker to access patient data and perform fake ambulance calls.

5.4 Software Quality Attributes

- Availability:
  The app must deliver a service which must be available at all times. This means there must be minimal delays in fetching and processing data and server-downtimes must also be kept to a minimum.

- Usability:
  The User Interface/User Experience must cater to a wide variety of users with differing capabilities to efficiently and effectively use the app with a minimum learning curve. The app must ensure at least more than 7 design rules defined by 'Nilsen' as 'Nielsen's 10 Golden Rules for Usability'.

- Correctness:
  Accuracy of the information delivered such as Prescription, Appointment time, etc is very crucial. There is a zero-tolerance for any mistakes in information delivered as these are critical to health of the user.

- Reusability:
  The app must be developed keeping in mind that this can be incorporated as part of a much larger software project. Thus the module must provide an API like endpoints for ease of reuse.

5.5 Business Rules

TBD

6. Other Requirements

- Database: TBD
- Legal
    - Confer with doctor patient confidentiality requirements
    - Determine legality of digitally signed prescriptions

Appendix A: Glossary

TBD

Appendix B: Analysis Models

TBD

Appendix C: To Be Determined List

TBD