Software Requirement Specification

CS 345-346

Medical Assistance Application

Group 13

Name	Roll Number
Tanay Maheshwari	190101092
Vignesh Ravichandra Rao	190101109
Tattukolla Lokesh	190101094
Shreyansh Meena	190101084

Contents

1 Introduction

1.1 Purpose

This Medical Assistance application is a telemedicine application which assists in the remote delivery of healthcare services like consultations. This SRS document is a part of the project under CS 346 - Software Engineering Lab.
The intended audience is our course instructor, Dr Samit Bhattacharya, and the teaching assistants. This document shows our progress in the requirement gathering and analysis stage of the software development life cycle.

1.2 Product Scope

The COVID-19 has shown the world the importance of telemedicine. More and more people have started to prefer short online consultations over physically visiting clinics. This application allows healthcare providers to evaluate, diagnose and treat patients without the need for an in-person visit. The application aims to connect doctors and patients in an easy-to-use and convenient manner.

1.3 Overview

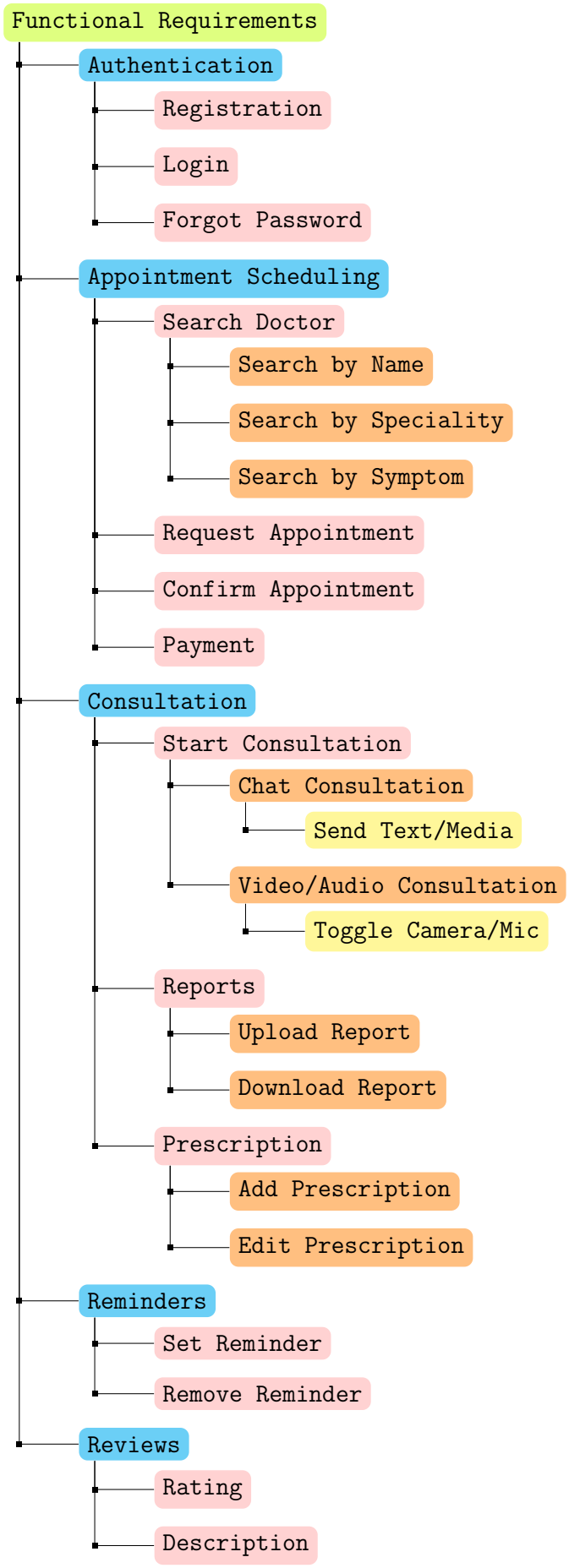
The remaining part of the SRS document contains:

- Hierarchical Depiction of Functional Requirements
- Functional Requirements
- Contextual Inquiry

1.4 References

- CS345 Lecture Slides by Dr. Samit Bhattacharya.
- Fundamentals of Software Engineering, Rajib Mall

2 Functional Requirements - Hierarchical Depiction



3 Functional Requirements

R.1 Authentication

R.1.1 Registration

- **Input**: User information
- **Output**: Success/failure prompt message
- **Description**: User enters their details. Based on the validity of entries new account is created or an error message is displayed

R.1.2 Login

- **Input**: User credentials
- **Output**: Success/failure prompt message
- **Description**: On successful authentication, the user is logged in.

R.1.3 Forgot Password

- **Input**: Email and New password
- **Output**: Success/failure prompt message
- **Description**: Update the user password on verifying email

R.2 Appointment Scheduling

R.2.1 Search Doctor

R.2.1.1 Search By Name:

- **Input**: Doctor Name
- **Output**: List of matching doctors
- **Description**: Filters the list of doctors based on their name

R.2.1.2 Search By Speciality:

- **Input**: Speciality
- **Output**: List of matching doctors
- **Description**: Filters the list of doctors based on their speciality (like cardiologist / dermatologist)

R.2.1.3 Search By Symptoms:

- **Input**: Symptoms
- **Output**: List of matching doctors
- **Description**: Filters the list of doctors based on the symptoms entered by patients (like nausea/headache)

R.2.2 Request Appointment

- **Input**: Date, time for appointment, patient details
- **Output**: Confirmation/Rejection prompt message
- **Description**: Users can request an appointment at a specified date and time with the available doctor.

R.2.3 Confirm Appointment

- **Input**: Request for appointment with patient details
- **Output**: Success/Failure prompt message
- **Description**: Doctors can confirm or reject appointment request from patients

R.2.4 Payment

- **Input**: Account details and amount
- **Output**: Amount transferred from sender to receiver
- **Description**: Users pay their fee online to doctor

R.3 Consultation

R.3.1 Start Consultation

R.3.1.1 Chat Consultation

R.3.1.1.1 Send Text/Media

- **Input**: User entered Text/Media
- **Output**: Text/Media shared and displayed
- **Description**: Facilitates two way chat communication between user and patient through text and media

R.3.1.2 Video/Audio Consultation

R.3.1.2.1 Toggle Camera/Mic

- **Input**: Current state of Camera/Mic - On or Off
- **Output**: Toggled state of Camera/Mic
- **Description**: Turn camera/mic on if it is currently switched off and vice versa.

R.3.2 Reports

R.3.2.1 Upload Report

- **Input**: Report in valid format
- **Output**: Upload status
- **Description**: Patients can upload reports for doctors to examine

R.3.2.2 Download Report

- **Input**: Report uploaded by patient
- **Output**: Download report into device
- **Description**: Doctors can download reports of patients for medical examination

R.3.3 Prescription

R.3.3.1 Add Prescription

- **Input**: Prescription content
- **Output**: Share prescription
- **Description**: Doctors can write prescription and upload it for patients

R.3.3.2 Edit Prescription

- **Input**: New prescription content
- **Output**: Previous prescription updated
- **Description**: Doctors can edit previous prescription

R.4 Reminders

R.4.1 Set Reminders

- **Input**: Date, time and frequency of dosage
- **Output**: Reminder notification at set time
- **Description**: Patients can set reminders to take their medicine on time.

R.4.2 Remove Reminders

- **Input**: Selected reminder
- **Output**: Delete reminder and display status
- **Description**: Delete a reminder that is no longer required.

R.5 Reviews

R.5.1 Rating

- **Input**: User Rating
- **Output**: Rating uploaded and displayed
- **Description**: Patients can give rating (in some appropriate scale) to doctors after consultation

R.5.2 Description

- **Input**: Review content
- **Output**: Review uploaded and displayed
- **Description**: Patients can write their consultation experience for other patients to see

4 Contextual Inquiry

4.1 Subject 1

- **Context:** Clinical Visit
- Subject visits General Physician quarterly.Reports are collected from the lab even though soft copies are available on the "SRL Diagnostics" app.
- It was inconvenient for the subject to carry the large report file in a rickshaw.
- Doctor observes reports and prescribes medication (if required)
- Doctor strikes through and changes prescription based on the patient's allergies.
- Doctor requires visual observations of certain parts for better diagnosis.

4.2 Subject 2

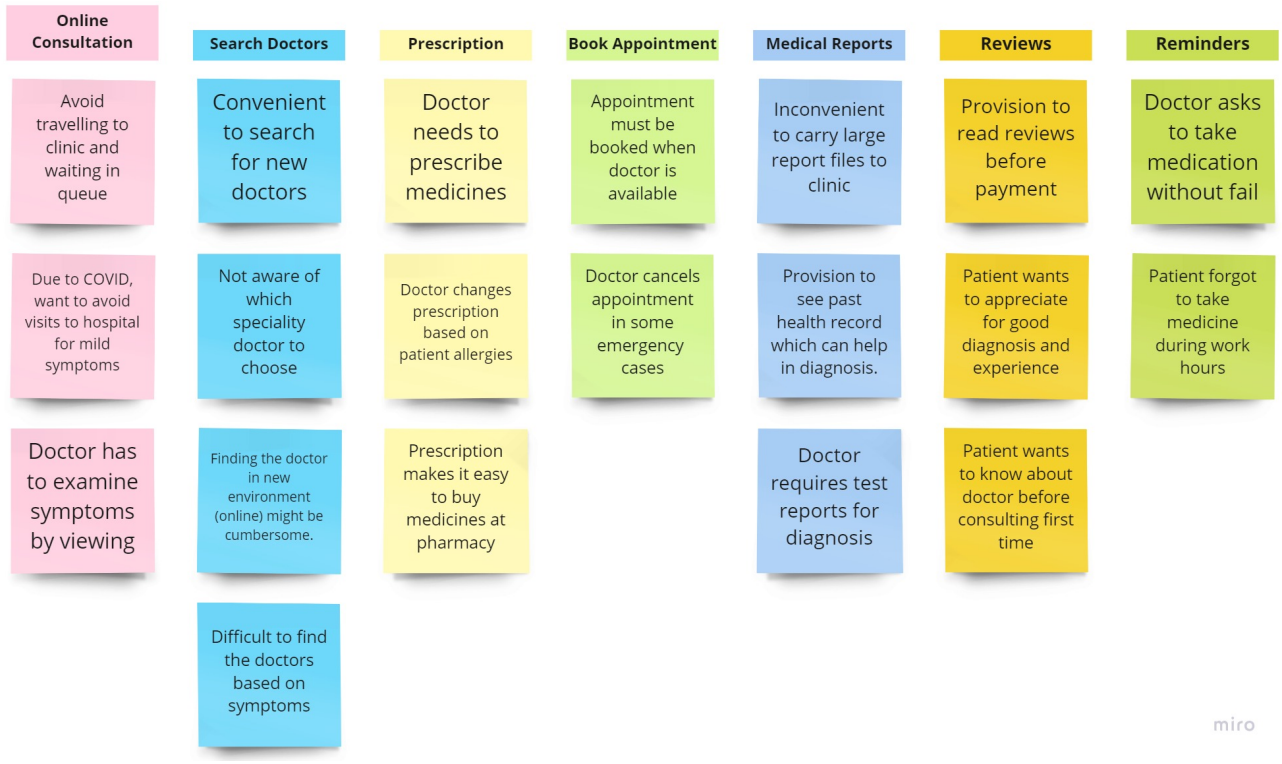
- **Context:** Apollo 247 Application
- The subject is a relative of a team member. Due to COVID restrictions, he could not physically go to a clinic and chose to use Apollo 247 App.
- The subject had to create an account and find an appropriate doctor.
- It was difficult to determine which speciality doctor to choose, based on symptoms - Pulmonology or Respiratory.
- Chat option gets enabled to communicate with doctor. Video or audio modes can also be used. The doctor asks to upload certain previous health records.
- The doctor examines the symptoms in video mode and gives a prescription. The subject is satisfied with the consultation

4.3 Subject 3

- **Context:** Practo App
- The subject is a team member and used the Practo App to avoid going to the clinic.
- The subject found a list of matching doctors after entering his symptoms.
- The subject was anxious about which doctor to choose since there was no account of their past consultations. A video consultation was booked after paying fee
- After prescribing medicine, the doctor advised the subject to set reminders on the app to take medicine timely since he had forgotten to take them regularly.

4.4 Affinity Diagram

The final drawing of the affinity diagram is given below:



4.5 Reflection

Based on the contextual inquiries, the following usability requirements were inferred:

- Doctors might have to examine symptoms and hence the application should allow doctors to view the patient.
- Doctors might require patient reports. Hence, patients must be allowed to upload their reports.
- Patients might not know which speciality doctor to consult, hence they should be able to search for doctors based on their symptoms.
- The application should allow doctors to share prescriptions with their patients. The prescriptions should be editable for modifications.
- Patients might want to know other users' reviews on a particular doctor before consulting. Hence, the application should display user reviews for a doctor.