SOFTWARE REQUIREMENTS SPECIFICATION

PREPARED FOR

Prof. Salman Abdul Moiz

PREPARED BY

Yasaswini Tiramdas

18MCME22

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to give a detailed description of the requirements for a Voice prescription application, enabling doctors and patients to create e-prescriptions. This application is necessary as it eases the paperwork for doctors and helps them focus more on the patient's needs and requirements. Because of the global pandemic, medical consultations are being done virtually either via phone or video calls. This application makes it easy for a doctor to take note of symptoms, diagnose, prescribe, and advise quickly without any hassle, virtually, and can directly send it to the patient.

The patient can have access to their medical consultations and prescription history. And these e-prescriptions can be easily carried and stored. This is opposite to the current situation where patients have to maintain physical files to store prescriptions and their test reports.

1.2. Scope

The "Voice Prescription" is a web application for doctors and patients. Where they can interact and doctors can diagnose the patient. The main focus is to generate a formatted prescription based on dictation from the doctor using keywords. The doctor would be able to edit/delete any entry that has been made in case there are any mistakes. The doctor will be able to preview the prescription and pre-sign the prescription digitally. After the diagnosis, the patient receives the e-prescription. The patient can also book an appointment before consultation using the application.

1.2.1. In-Scope

- Login system for Doctors.
- Login system for patients.
- Patients can book an appointment with the doctor.
- Doctors provide a formatted prescription.
- Send the prescription to the patient.
- Facility to pre-sign the prescription digitally.
- Storing the medical history of a patient.
- Patients have access to their medical history.

1.2.2 Out-Scope

- Mailing the prescription to the nearby pharmacy in case of an emergency.
- Suggestions for various pharmacies depending on the availability of the drugs prescribed.
- Provide suggestions for diagnosis to the doctor based on the input symptoms.
- Provide information to the patient about the availability of the doctor for surgery or consultation.

1.3. Definitions, Acronyms, & Abbreviations

Table 1 - Definitions

Term	Definition
User	Someone who uses the web application.
Doctor	Someone who writes the prescription and signs it.
Patient	Someone who receives the diagnosis and prescription.
Web-Portal	A web application where the prescription is written by doctors and read by patients.
E-Prescription	An Electronic prescription that can be opened/viewed in mobiles or Computers
DESC	Description.
RAT	Rational
DEP	Dependency.
GIST	A short, simple description of the concept containing a programming language statement.
SCALE	The scale of measure used by the requirement contained in a language statement.
METER	The process or device used to establish a location on a SCALE contained in a language statement.
MUST	The minimum level required to avoid failure

	contained in a programming language statement.
DEFINED	The official definition of a term contained in a programming language statement.
PLAN	The level at which good success can be claimed contained in a language statement
WISH	A desirable level of achievement that may not be attainable through available means contained in a language statement

1.4. References

1. IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830™-1998(R2009). (Revision of IEEE Std 830-1993).
2. https://pypi.org/project/SpeechRecognition/

1.5. Overview

The remainder of this document includes three chapters and an appendix. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the functional requirements specification in detailed terms and a description of the different system interfaces. Performance requirements along with design constraints and software system attributes are also discussed.

The Appendix at the end of the document includes the Use Case Specifications in detail.

2. Overall Description

This section gives you an overview of the whole application including the constraints and assumptions. The application will be explained with the basic roles and functionalities of each role.

2.1. Product Perspective

This application will basically have two roles: doctor and patient. The doctor can write prescriptions based on the diagnosis s/he made. The patient can access his/her prescriptions and book an appointment with the doctor.

The doctor will first create an account by inputting his/her details such as name, specialization, designation, qualification, contact details, etc. After successfully creating an account they can confirm the appointments booked by a patient and schedule a consultation. During the consultation, the doctor will be able to write a prescription based on the keywords they say and sign it digitally. Before the prescription is submitted the doctor has a chance to edit the errors if any. If the patient had a consultation before this, the doctor will be able to access his older prescriptions.

The patient would also create an account by inputting their details such as name, email, age, etc. After the successful creation of an account, they can book an appointment with the concerned doctor. After the consultation, the patients will get a digital prescription. The patient will be able to access all their old prescriptions.

2.2. Product functions

The functionalities of the application/web portal are as follows:

- Manage Account
  Both user roles Doctor and Patient can manage their accounts, that is, create an account and login to their account.
- Manage Prescriptions
  The Doctor can create a prescription, and it cannot be edited once saved.
- Get Patient's History
  The Doctor can view the patient's older prescriptions.
- Book Appointment
  The Patient can book an appointment with the doctor.

- Get Prescription
  The Patient will receive the prescription once the doctor saves it.

2.3. User characteristics

There are two types of users for this application: Doctors and Patients. Each of these users has different uses of the application so each of them has its own requirements.

Doctors must be able to know how to use the internet and a web portal. They should be able to log in to their accounts and schedule appointments with the patients. During an appointment, they should be able to understand the functionality of the voice typing, edit and saving before ending the consultation.

Patients should be able to know how to use the internet and a web browser. They should be able to log in to their account and view the correct prescriptions they want.

2.4. User constraints

The Internet connection is a constraint for the application. Since the application gets and saves the prescriptions over the Internet, it is crucial that there is an Internet connection for the application to function.

Doctors must have a valid practising certificate and qualification.

2.5. Assumptions & dependencies

No specific assumptions and dependencies.

2.6. Apportioning requirements

If the application is widely accepted, we can incorporate the out-scope features into the existing application.

3. Specific Requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

3.1. Interface requirements

3.1.1. User Interface

As soon as the user logs in, he/she will be able to see the login page. If the user has not registered, he/she should be able to do that on the log-in page. Every user should have a profile page where they can edit their name and password.

The doctor will be able to confirm appointments and schedule them on one screen, they will also be able to write, edit and save the prescriptions. They will be able to see the patient's history if it's available.

The patient will be able to book an appointment with their desired doctor. They will also be able to view all their prescriptions on one page.

The screen captures are added in Appendix 2.

3.1.2. Hardware Interface

Since the web portal does not have any designated hardware, it does not have any hardware interfaces.

3.1.3. Software Interface

Since the web portal does not have any other software usage, it does not have any software interfaces.

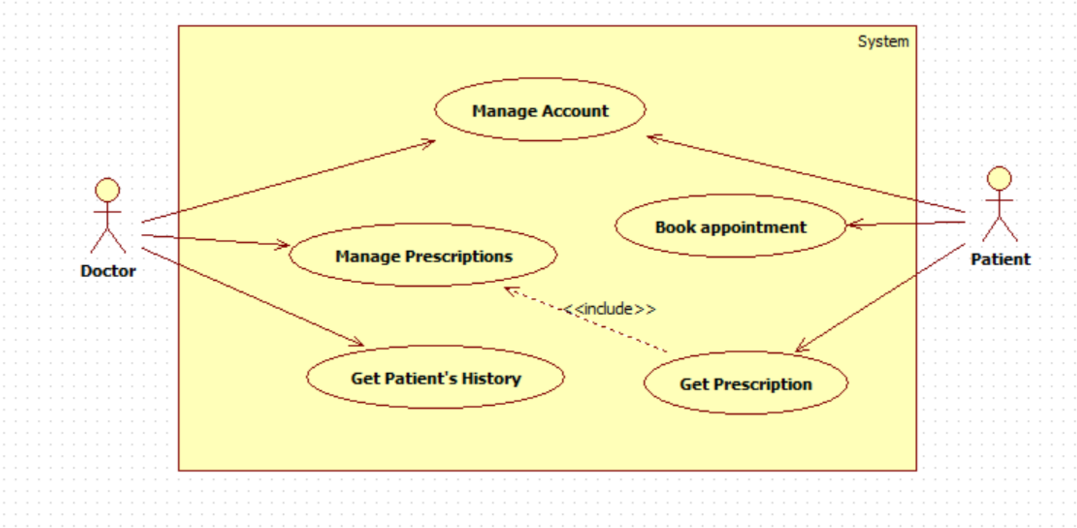
3.1.4. Communication Interface

Communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the web portal.

3.2. Functional Requirements

3.2.1. Use case model

There are 5 use cases and 2 actors namely Doctor and Patient for this application

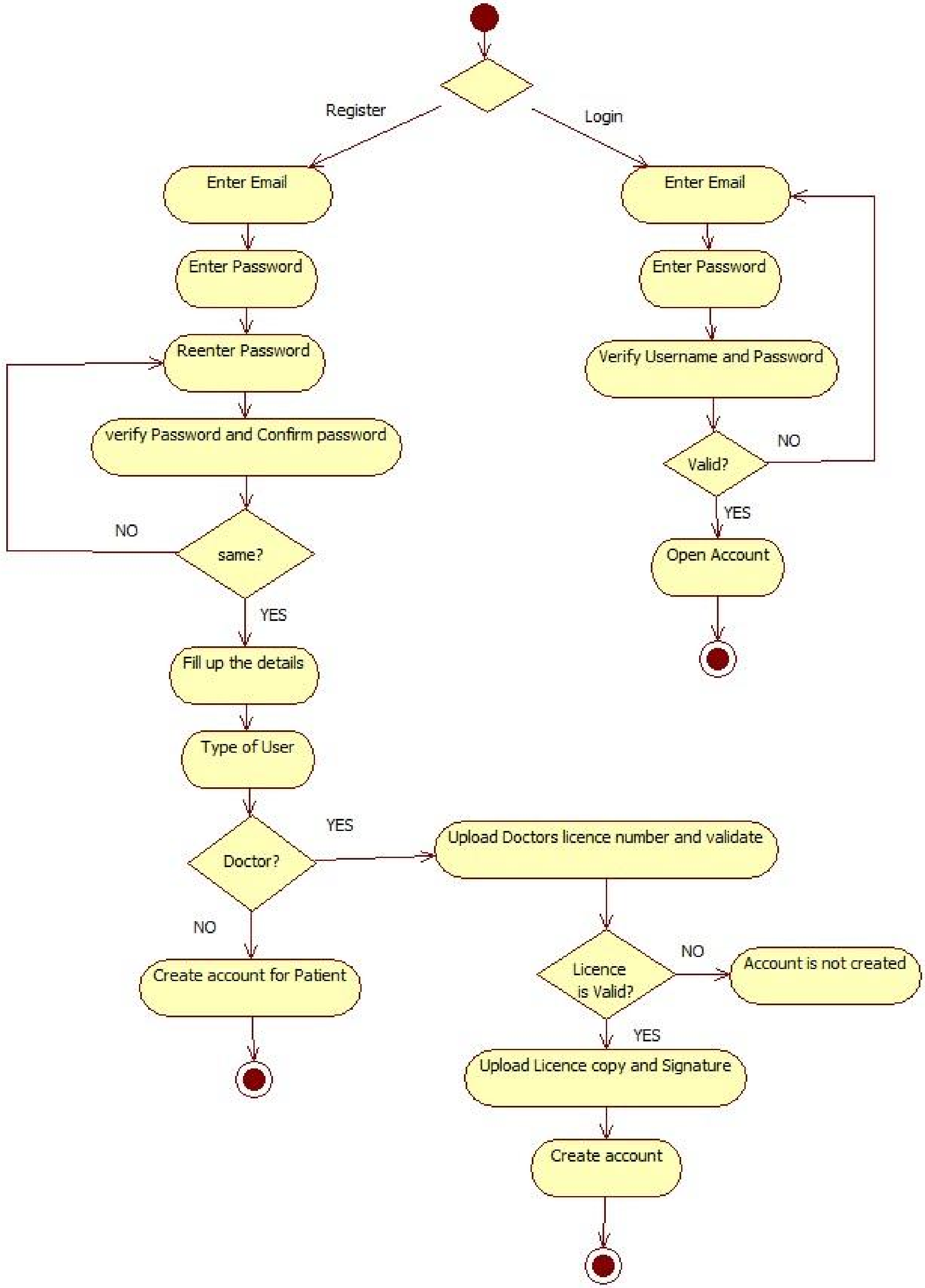


3.2.2. Use Case Specifications

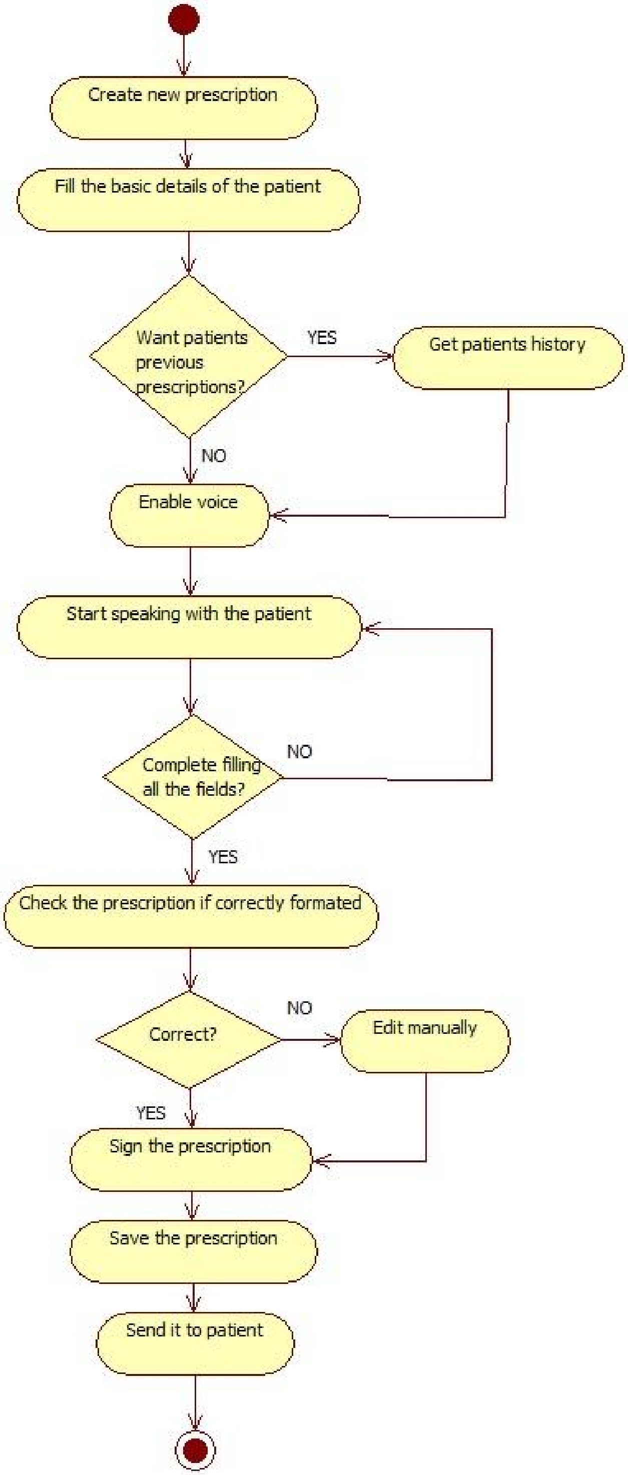
The specifications are clearly mentioned in Appendix 1.

3.2.3. Activity Diagram

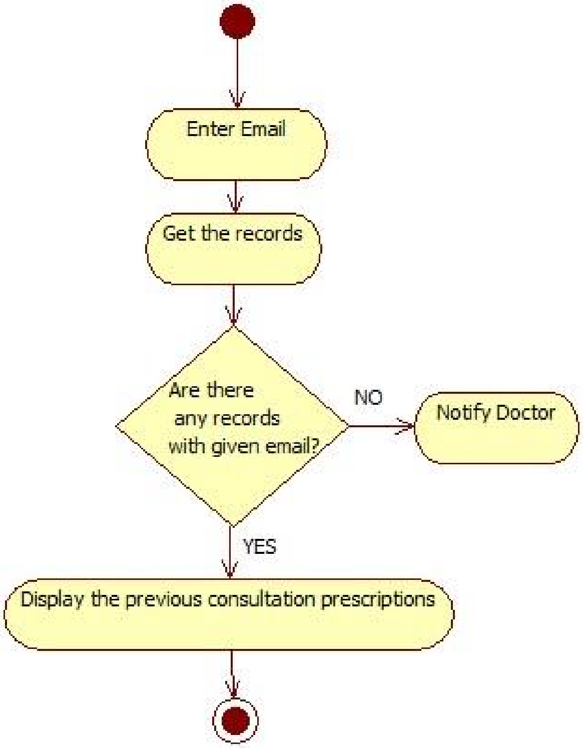
Manage Account


Register
Login
Enter Email
Enter Password
Reenter Password
verify Password and Confirm password
NO
same?
YES
Fill up the details
Type of User
Doctor?
YES
NO
Create account for Patient
Upload Doctors licence number and validate
Licence is Valid?
NO
Account is not created
YES
Upload Licence copy and Signature
Create account
Enter Email
Enter Password
Verify Username and Password
Valid?
NO
YES
Open Account

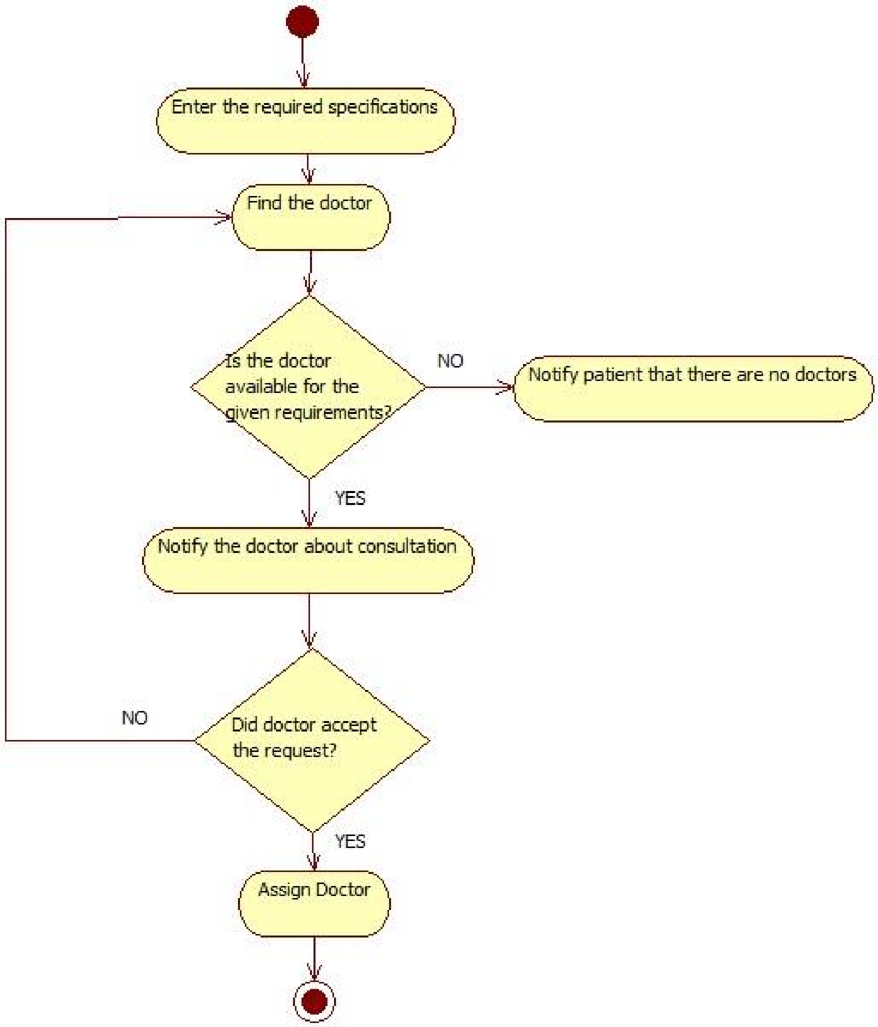
Manage Prescriptions


Create new prescription
Fill the basic details of the patient
Want patients previous prescriptions?
YES
Get patients history
NO
Enable voice
Start speaking with the patient
Complete filling all the fields?
NO
YES
Check the prescription if correctly formated
Correct?
NO
Edit manually
YES
Sign the prescription
Save the prescription
Send it to patient

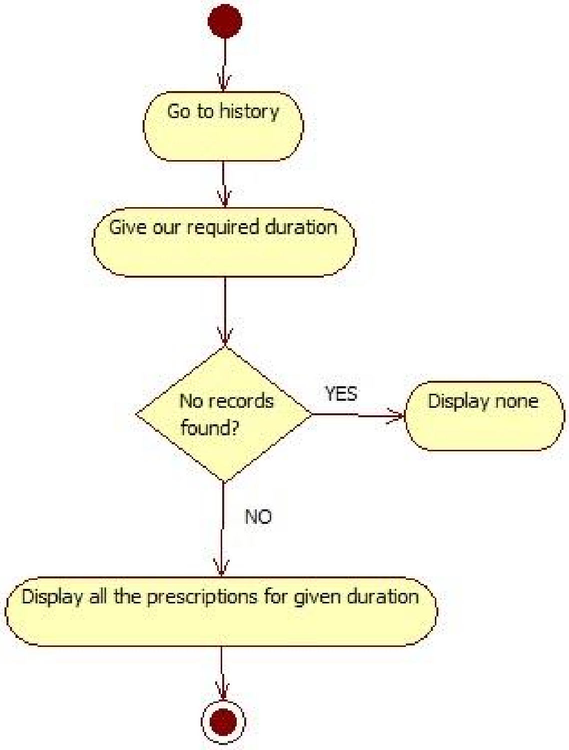
Get Patients History


Enter Email
Get the records
Are there any records with given email?
NO
Notify Doctor
YES
Display the previous consultation prescriptions

Book appointment


Enter the required specifications
Find the doctor
Is the doctor available for the given requirements?
NO
Notify patient that there are no doctors
YES
Notify the doctor about consultation
NO
Did doctor accept the request?
YES
Assign Doctor

Get Prescription


Go to history
Give our required duration
No records found?
YES
Display none
NO
Display all the prescriptions for given duration

3.2.4. Analysis Classes

Identified Classes are:

- User
- Doctor
- Patient
- Prescription
- Appointment

Description of the analysis classes:

- User[Email, Password, type(doctor/patient)]
- Doctor[Doctor ID, Name, Specialization, designation, Experience, Licence number, Digital signature]
- Patient[Patient ID, age, Is Diabetic, Has Hypertension]
- Prescription[Patient ID, Doctor ID, Date, Symptoms, Medicines, Dosage]
- Appointment[Patient ID, Doctor ID, Date, Time, Approved]
- User is the superclass of Doctor and Patient.

3.3. Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

There should be a proper list view of all the details as the results displayed in the list view should be user friendly and easy to understand. Selecting an element in the result list should only take one click.

The response time of data flowing should be no more than 1 sec. System dependability should be 100% of the time.

Appointments will contain data such as patient Id and doctor ID along with the time and date of the appointment.

By considering the security issues all the data insertions and deletions are done only through the application. The application takes data from the users and by the server, the data will be stored in the database and vice versa.

3.4. Logical database requirements

We create Databases using MySQL, based on the Use Class requirements.

There are 2 types of users, they are Doctor and Patient. Both the users are generalised and have the attributes UserId, email, password and type. UserId is a unique entity that identifies the user. Type value determines whether the user is a doctor or a patient.

Doctors have extra attributes like his qualification details, his specialisation, digital signature and contact details. Whereas for a patient we have details such as age, his general medical details like diabatic, hypertension etc.

Prescriptions have various attributes like doctor id, patients id, symptoms, medicine name, dosage. Doctor id and patient id specify that this particular doctor has consulted the patient with that ID.

3.5. Design Constraints

This section includes the design constraints on the software caused by the hardware.

TITLE: Application Memory Usage

GIST: The amount of Operating System memory occupied by the application.

SCALE: MB.

METER: Observations done from the performance log during testing

MUST: No more than 1000 MB.

PLAN: No more than 500 MB

WISH: No more than 400 MB

Operating System: DEFINED: The mobile Operating System on which the application is running on.

MB: DEFINED: Megabyte.

3.6. Software System attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

3.6.1. Reliability

TITLE: The reliability of the system.

SCALE: The reliability that the system gives the right result on login.

METER: Measurements obtained from 1000 searches during testing.

MUST: More than 98% of the searches.

PLAN: More than 99% of the searches.

WISH: 100% of the searches.

3.6.2. Availability

TITLE: The availability of the system.

GIST: The availability of the system when it is used.

SCALE: The average system availability (not considering network failing).

METER: Measurements obtained from 1000 hours of usage during testing.

MUST: More than 98% of the time.

PLAN: More than 99% of the time.

WISH: 100% of the time.

TITLE: Internet Connection

DESC: The application should be connected to the Internet.

RAT: In order for the application to communicate with the database.

DEP: None

3.6.3. Security

TITLE: Communication Security

GIST: Security of the communication between the system and server.

SCALE: The messages should be encrypted for log-in communications, so others cannot get email and password from those messages.

METER: Attempts to get email and password through obtained messages on 1000 log-in sessions during testing.

MUST: 100% of the Communication Messages in the communication of a log-in session should be encrypted.

Communication Messages: DEFINED: Every exchange of information between client and server.

TITLE: User login System.

GIST: Security of accounts.

SCALE: If a patient/Doctor tries to log in to the web portal with a non-existing account then the restaurant owner should not be logged in. The user should be notified about log-in failure.

METER: 1000 attempts to log-in with a non-existing user account during testing.

MUST: 100% of the time

TITLE: User account - create security.

GIST: The security of creating an account for users of the system.

SCALE: If a user wants to create an account and the desired email is occupied, the user should be asked to choose a different email.

METER: Measurements obtained on 1000 hours of usage during testing.

MUST: 100% of the time.

3.6.4. Maintainability

TITLE: Application extendability

DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

RAT: In order for future functions to be implemented easily to the application.

DEP: none

TITLE: Application testability

DESC: Test environments should be built for the application to allow testing of the application's different functions.

RAT: In order to test the application.

DEP: none

3.6.5. Portability

TITLE: Application portability

DESC: The application should be portable with any browser in either PC or System.

RAT: The adaptable platform for the application to run on.

DEP: None

Appendix 1

Use case Specification:

Manage Account

Use Case ID:	1		
Use Case Name:	Manage Account		
Created by:	Yasaswini Tiramdas	Last Updated by:	Yasaswini Tiramdas
Date Created:	24-03-2021	Date Last Updated:	03-06-2021

Actors:	Doctor, Patient
Description:	This use case is used by doctors and patients to create an account to perform various activities.
Preconditions:	Doctors must have a valid certification and degree. The patient must have an email account created.
Postconditions:	An account is created.
Normal Flow:	Account Registration:- 1.1 Open create account form.(A-1) 1.2 Enter Email 1.3 Give password and confirm password. (A-2) 1.4 Fill in the basic details like name, contact details etc. 1.5 Select the type of user Doctor or Patient. 1.6 If Patient, YES go to next step(A-3) 1.7 Upload the licence number and scanned copy of the same.(A-4) 1.8 Validate the Licence number. (A-5) 1.8 Upload digital signature as an image. (A-6) 1.9 Account is created. (A-7) Login:- 2.1: Give the email. 2.2: Give Password. 2.3: Verify and validate the email and password with the user's data. (A-8)

Alternative Flows:	A-1: If an account with that email and details already exists terminate redirect to 2.1 A-2: If password and confirm password do not match, ask to reenter the password. A-3: If the user is patient, create his account. A-4: If the file is not uploaded then give an alert or If the documents are invalid restrict from creating the account. A-5: If the licence number is invalid terminate the use case. A-6: If the sign is not uploaded, notify the user. A-7: If the given email address is fake then terminate the use case. A-8: If the details don't match restrict from logging in and ask the user to register that is redirect to 1.1.
Exceptions:	-
Priority:	Medium
Frequency of Use:	Moderately
Business Rules:	- The account should be created before logging in. - The doctor details should be stored properly for future use or verification if any problem occurs. - The account should be verified. - The password must have an upper case, lower case and a special character included with a minimum length of 8 characters. - The patient account can be created with a proper and valid Gmail account.
Assumptions:	- The Doctor creating the account has a legal licence.

Manage Prescription

Use Case ID:	2		
Use Case Name:	Manage Prescription		
Created by:	Yasaswini Tiramdas	Last Updated by:	Yasaswini Tiramdas
Date Created:	24-03-2021	Date Last Updated:	03-06-2021

Actors:	Doctor
Description:	This use case is used by doctors to create, edit, save the prescription and send it to the patient.
Preconditions:	The doctor must have an account.
Postconditions:	The doctor will be able to send the prescription to the patient.
Normal Flow:	2.1 Create a new prescription. 2.2 Enable voice. 2.3 Start speaking, the application captures the patient's name, age, phone number and format the prescription based on speech. (A-1) 2.4 Preview the prescription. 2.5 Attach Sign to the prescription. (A-2) 2.6 Save the details. 2.7 Send it to the patient.
Alternative Flows:	A-1: If the formatting is not done properly or if there is any mistake in converting from speech to text, the doctor can edit it manually. A-2: If the sign is not attached the prescription is considered invalid and can't send it to the patient.
Exceptions:	-
Priority:	High
Frequency of Use:	Frequently
Business Rules:	- The doctor manually checks whether the formatting is done properly or not. - A doctor can view the patient's history. - Once the prescription is saved the doctor cannot edit or delete it in the future. - The data store will be considered as the Patient's history when he consults in the future.
Assumptions:	- It is assumed that the sign is valid and is uploaded.

Get Patients History

Use Case ID:	3		
Use Case Name:	Get Patient's History		
Created by:	Yasaswini Tiramdas	Last Updated by:	Yasaswini Tiramdas
Date Created:	24-03-2021	Date Last Updated:	03-06-2021

Actors:	Doctor
Description:	This use case is used by doctors to get the medical history of a patient.
Preconditions:	The doctor gives the name/email of the patient.
Postconditions:	Previous medical prescriptions are shown.
Normal Flow:	3.1 Give the name/email of the patient. 3.2 Validate the name/email. (A-1) 3.3 Get the details
Alternative Flows:	A-1: If the details don't exist in the database then no output is produced.
Exceptions:	-
Priority:	High
Frequency of Use:	Frequently
Business Rules:	- If the history exists, it'll be displayed else new patient's details will be inserted
Assumptions:	-

Book appointment

Use Case ID:	4		
Use Case Name:	Book Appointment		
Created by:	Yasaswini Tiramdas	Last Updated by:	Yasaswini Tiramdas
Date Created:	24-03-2021	Date Last Updated:	03-06-2021

Actors:	Patient
Description:	This use case is used by the patient to book an appointment for a consultation.
Preconditions:	The Patient must have an account.
Postconditions:	The patient will be able to book an appointment.
Normal Flow:	4.1 Give the specialisation of the doctor as required. 4.2 Provide the specification of diagnosis. 4.3 Give date and time of consultation. 4.4 Submit the details. (A-1) 4.5 Application will find the doctor as per our requirements and send the request to the doctor. 4.6 Doctor accepts and the doctor is assigned to the patient.(A-2)
Alternative Flows:	A-1: If we are not able to find any doctor then send the alert message to the patient. A-2: If the doctor is busy, go to step 4.5.
Exceptions:	-
Priority:	Medium
Frequency of Use:	Moderately
Business Rules:	- The doctor will be assigned automatically by the application using a standard mechanism.
Assumptions:	-

Get Prescription

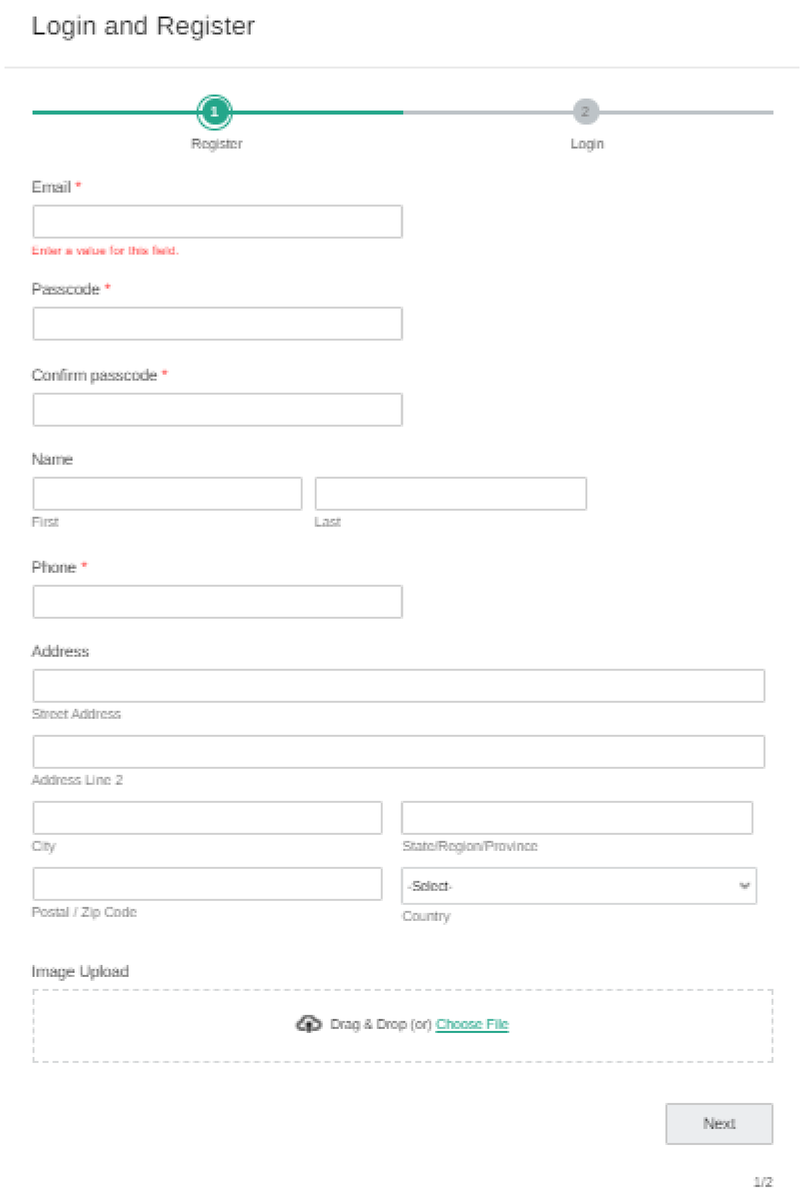
Use Case ID:	5		
Use Case Name:	Get Prescription		
Created by:	Yasaswini Tiramdas	Last Updated by:	Yasaswini Tiramdas
Date Created:	24-03-2021	Date Last Updated:	05-05-2021

Actors:	Patient
Description:	This use case is used by the patient to get his latest medical prescription.
Preconditions:	The patient has an account and has at least one consultation.
Postconditions:	Gets his prescriptions.
Normal Flow:	5.1 Go to history. 5.2 Select the range of history he wants(1 week, 1 month, or the latest one) (A-1) 5.3 Gets his prescription list. 5.4 Can click on mail to get the prescription to mail/Download it.
Alternative Flows:	A-1: If the details don't exist in the database then no output is produced and the use case is terminated
Exceptions:	-
Priority:	High
Frequency of Use:	Frequently
Business Rules:	- The patients can download the prescription once the email is verified. - The details of the patient are automatically stored when the prescription is created by the doctor.
Assumptions:	-

Appendix 2

User Interfaces:

Registration [Generalized]



Login and Register

1 Register

2 Login

Email *

Enter a value for this field.

Passcode *

Confirm passcode *

Name

First

Last

Phone *

Address

Street Address

Address Line 2

City

State/Region/Province

Postal / Zip Code

-Select-

Country

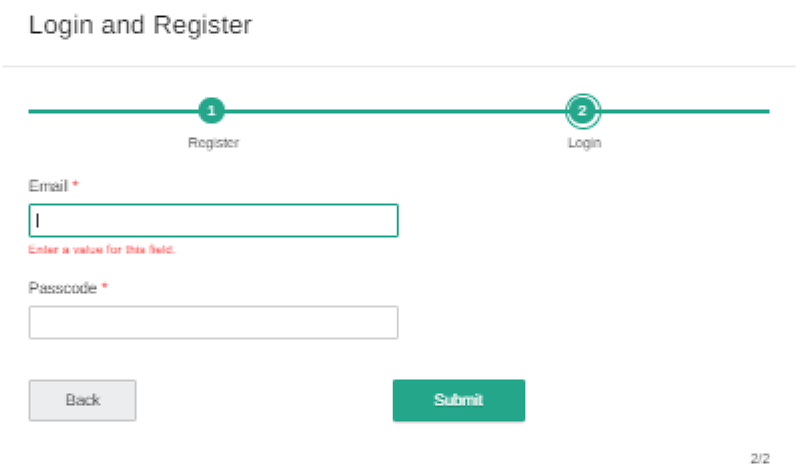
Image Upload

Drag & Drop (or) Choose File

Next

1/2

Login [Generalized]



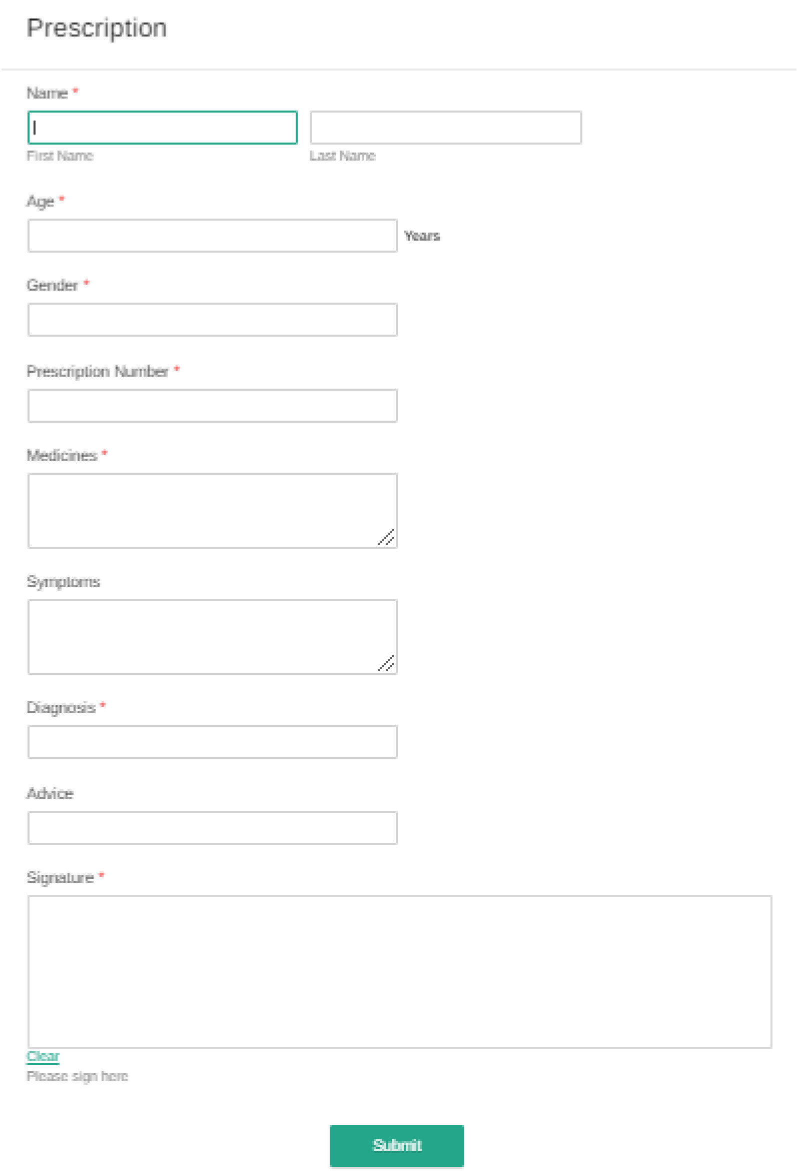
Login and Register

Email *

Enter a value for this field.

Passcode *

Prescription



Prescription

Name *

First Name

Last Name

Age *

Years

Gender *

Prescription Number *

Medicines *

Symptoms

Diagnosis *

Advice

Signature *

Clear

Please sign here

Submit