



## IT632 Software Engineering Group 1

### Members:

Anil Mangtani (202112005)  
Arpitha Sreenivasan (202112030)  
Gautam Patel (202112086)  
Kalp Vora (202112018)  
Khusboo Chhabra (202112014)  
Nikhil Chhabria (202112026)  
Parth Soni (202112007)  
Ravi Agarwal (202112021)  
Reniz Shah (202112126)  
Vivek Doshi (202112063)

### Guided by

Prof. Saurabh Tiwari  
Mr. Varun Shah

## Scope

In the existing system patients are given physical copies of the prescription. Patients need to maintain files and prescriptions given by different doctors and there are high chances of losing some prescriptions.

In this project, we are creating a website that will allow patients to store their medical records at a centralized location and will be accessible to them whenever they want. Along with patients, doctors can view the medical records during the consultation which will help them in their diagnosis. The chances of losing a record are very low so doctors can view a patient's entire medical history.

## Process Model - Waterfall model

The requirements of the systems are very straightforward and well-understood. Since they are well-understood there wouldn't be any change in the requirements at a later stage. Thus we chose the Waterfall model as our process model.

## Users and Stakeholders

### End Users:

Any person with a valid Aadhar id can register as a patient in the system.  
Any doctor with a valid license number and Aadhar id can try to register in the system.  
Admin is exclusive for the stakeholder representatives.

### Stakeholders:

Government health bodies are the stakeholders for the system as the patient health details should be handled and viewed by the government bodies only.

## Functional and Non-functional requirements

### Functional requirements:

Registration : Patients need to register using Aadhaar since it's a universal record so multiple registrations of the same Patient are not possible. Doctors need to register and provide a doctor's license and other details which will be used for background checks. Only after getting verified by the admins, doctor's account will be created

Login : The registered Patient/Doctor/Admin needs to log in to access the functionalities of the system.

Patient and Doctor can edit his/her profile.

Patient and Doctor can view his/her profile.

Doctor and Admin can view medical records of any Patient

#### Patient:

1. After registration is completed Patients will fill in the basic details and initial medical details.
2. Patients can view his/her medical records with timelines.
3. Patients can add previous prescriptions/consultations as well.
4. Patient can add any prescription/consultation which he/she got from a doctor not registered to our application.

#### Doctor:

1. Doctors can add a new prescription/consultation.

#### Admin:

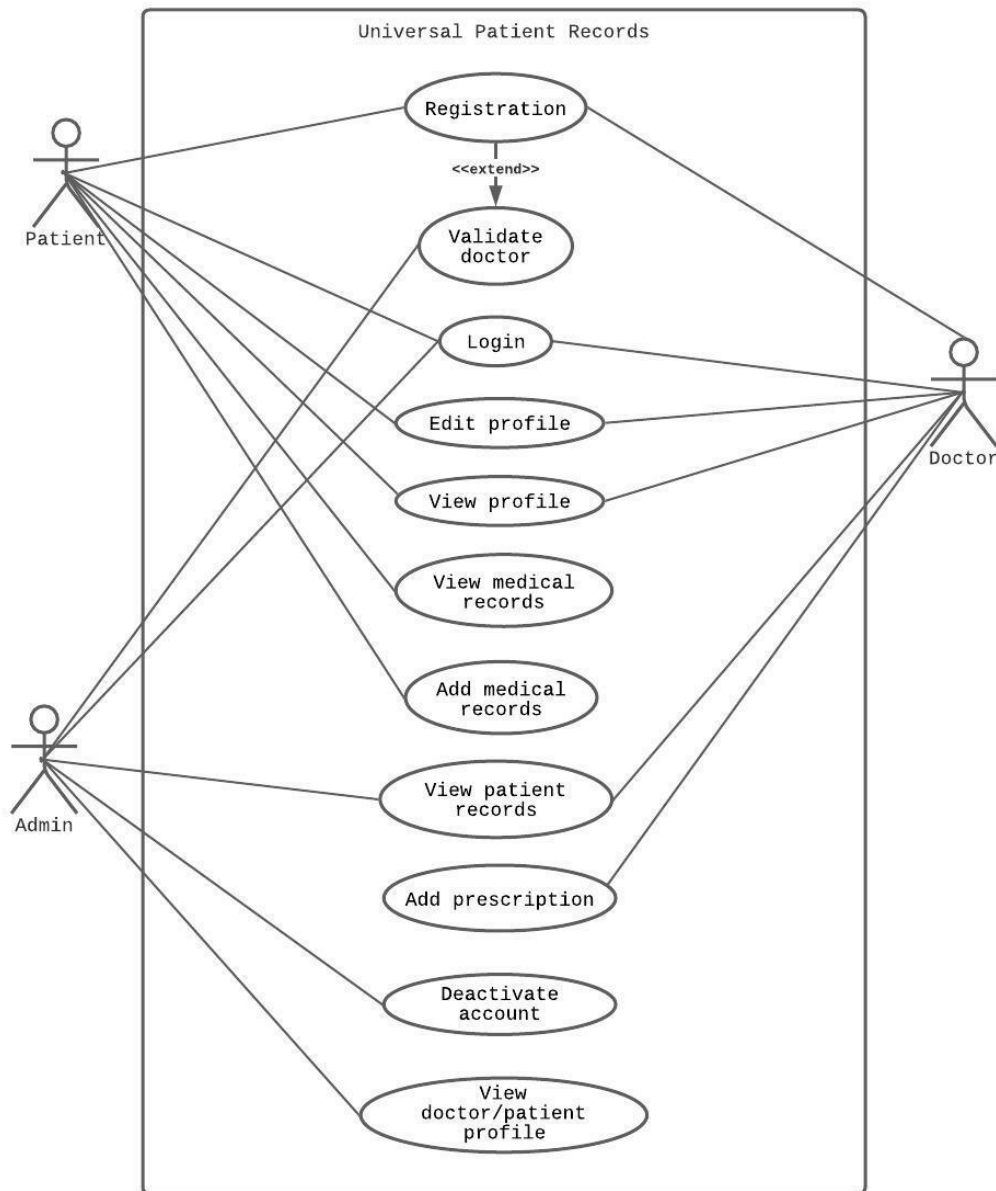
1. Admin will verify the doctor that has registered to the system and only after his/her approval doctor's account will be created else doctor's account will not be created.
2. Admin can deactivate the patient/doctor accounts.
3. Admin can view the profile of doctors and patients.

### Non-Functional Requirements:

1. Reliability: While inserting any data into the field the system must ensure that data is not altered before storing it in the database.
2. Design Constraints: The system should be developed using HTML 5, React, in the front-end side, and on the server-side Django will be used, and for the Database, MySQL is used. The system will be developed as a web application that can be used with all the latest browsers.
3. Availability: The system can be used or accessed at any time of the day. It'll be operational for the whole day.

4. Efficiency: The system must be efficient enough to handle multiple requests from doctors and patients from multiple locations.
5. Usability: The User-Interface must be simple and clean so that it becomes easier for every patient and doctor to use.
6. Capacity - The database must be capable enough to hold all the data like the patient's personal information and medical records along with doctor details.
7. Regulatory - The system will work within the boundaries of the law.

## Use Case Diagram



## Use Case Description

### 1. Registration (U1)

1.1 Introduction: User will be able to register themselves with the application

#### 1.2 Actors:

- 1.2.1 Patient
- 1.2.2 Doctor

#### 1.3 Post-conditions:

- 1.3.1 Account created for Patient
- 1.3.2 Account created for Doctor

#### 1.4 Basic flow:

- 1.4.1 User navigates to **sign up**
- 1.4.2 User enters the details
- 1.4.3 System checks if user exists
- 1.4.4 Account created

#### 1.5 Alternate flow:

- 1.5.1 For patient
  - 1.5.1.1 User will have to enter patient centric details
  - 1.5.1.2 Account will be created for patient
- 1.5.2 For doctor
  - 1.5.2.1 Admin manually verifies the doctor
  - 1.5.2.2 Account will be created for doctor

#### 1.6 Exceptional flow:

- 1.6.1 If user already has an account sign up will be unsuccessful
- 1.6.2 If verification for doctor is unsuccessful then account for doctor will not be created
- 1.6.3 If the user clicks “cancel” in the middle of the process the account will not be created.

#### 1.7 Special requirements:

- 1.7.1 Password should have at least 8 characters, at least 1 non alphanumeric character. It can contain a-z, A-Z, 0-9, @, #.
- 1.7.2 Patient and Doctor must have a valid Aadhar id.
- 1.7.3 Doctors should have a valid medical practitioner’s license.

#### 1.8 Use case relationships:

- 1.8.1 extends “validate doctor” (U2)

### 2. Validate doctor (U2)

2.1 Introduction: Doctor will be verified

2.2 Actors:

2.2.1 Admin

2.3 Pre-conditions:

2.3.1 Doctor should have entered all the necessary details.

2.4 Post-conditions:

2.4.1 The doctor will be eligible for an account

2.5 Basic flow:

2.5.1 starts when doctor registers into the application

2.5.2 the admin verifies the account using doctor's license number

2.5.3 doctor is verified

2.6 Exceptional flow:

2.6.1 if doctor hasn't entered all the necessary details the verification process fails

2.6.2 if so problem has been found during manual background check by admin verification fails

2.7 Use case relationships:

2.7.1 extended from registration(U1)

### 3. Login (U3)

3.1 Introduction: Users will be able to log into their respective accounts with username and password.

3.2 Actors:

3.2.1 Patient

3.2.2 Doctor

3.2.3 Admin

3.3 Pre-conditions:

3.3.1 User should have a verified account

3.4 Post-conditions:

3.4.1 User will be able to use services provided by the system

3.5 Basic flow:

3.5.1 Starts when the user enters username and password and clicks the "Login" button.

3.5.2 The user credentials are verified by the system

3.5.3 The user is logged in

### 3.6 Alternate flow:

#### 3.6.1 Reset password

- 3.6.1.1 User clicks on reset password
- 3.6.1.2 The user will receive a code via email
- 3.6.1.3 User enters the code
- 3.6.1.4 User submits the code
- 3.6.1.5 User will be asked for new password
- 3.6.1.6 User enters new password
- 3.6.1.7 User submits the password
- 3.6.1.8 User logs in

### 3.7 Exceptional flow:

- 3.7.1 If username or password is incorrect the login will be unsuccessful
- 3.7.2 If password reset is unsuccessful then login will be unsuccessful
- 3.7.3 If no account for the user is found then login will be unsuccessful

## 4. Edit profile (U4)

### 4.1 Introduction: User can edit profile in the system

### 4.2 Actors:

- 4.2.1 Patient
- 4.2.2 Doctor

### 4.3 Pre-conditions:

- 4.3.1 User should be logged in

### 4.4 Post-conditions:

- 4.4.1 User profile will be updated

### 4.5 Basic flow:

- 4.5.1 User navigates to **Edit Profile page**
- 4.5.2 User selects the field to be updated
- 4.5.3 Changes the field
- 4.5.4 System updates the field

### 4.6 Exceptional flow:

- 4.6.1 If a user tries to change “aadhar number”, the profile will not be edited.
- 4.6.2 If user cancels the process the profile will not be updated



## 5. View profile (U5)

5.1 Introduction: User will be able to view their profile

5.2 Actors:

5.2.1 Patient

5.2.2 Doctor

5.2.3 Admin

5.3 Pre-conditions:

5.3.1 User should be logged in

5.4 Post-conditions:

5.4.1 User will be able to view their profile details

5.5 Basic flow:

5.5.1 starts when user navigates to "profile"

5.5.2 User will be able to view their profile details

## 6. View medical records (U6)

6.1 Introduction: Patient will be able to view their medical records

6.2 Actors:

6.2.1 Patient

6.3 Pre-conditions:

6.3.1 User should be logged in

6.4 Post-conditions:

6.4.1 User will be able to view their medical records

6.5 Basic flow:

6.5.1 Starts when user navigates to "View Medical Records"

6.5.2 User will be able to view their medical records

## 7. Add medical records (U7)

7.1 Introduction: Patient will be able to add their medical records

7.2 Actors:

7.2.1 Patient

7.3 Pre-conditions:

7.3.1 user should be logged in

7.4 Post-conditions:

7.4.1 User's medical records will be updated with the new data

#### 7.5 Basic flow:

- 7.5.1 Starts when user navigates to “Add Medical Records”
- 7.5.2 User can type or attach the documents for medical records
- 7.5.3 User confirms the changes
- 7.5.4 Medical records are updated

#### 7.6 Exceptional flow:

- 7.6.1 If user discards the changes instead of confirming them, the medical records will not be updated
- 7.6.2 If user submits the changes without making any, the medical records will not be updated

### 8. View patient records (U8)

8.1 Introduction: Describes how doctor/admin can view record of patient

#### 8.2 Actors:

- 8.2.1 Doctor
- 8.2.2 Admin

#### 8.3 Pre-conditions:

- 8.3.1 doctor/admin should be logged into their accounts

#### 8.4 Post-conditions:

- 8.4.1 doctor/admin will be able to view patient records

#### 8.5 Basic flow:

- 8.5.1 Doctor/Admin clicks on view patient record
- 8.5.2 Doctor/Admin enter's the patient ID
- 8.5.3 Doctor/Admin can view the patient records

#### 8.6 Exceptional flow:

- 8.6.1 If the doctor/admin enters a wrong patient ID the patient records will not be retrieved.

### 9. Add prescription (U9)

9.1 Introduction: doctor will be able to add prescription for a patient

#### 9.2 Actors:

- 9.2.1 Doctor

#### 9.3 Pre-conditions:

- 9.3.1 Doctor should be logged into their account

#### 9.4 Post-conditions:

9.4.1 Patient records will be updated with the new prescription entered by the doctor

9.5 Basic flow:

9.5.1 Doctor clicks on Add Prescription button

9.5.2 Doctor enters the details in the form

9.5.3 Doctor confirms the changes

9.5.4 The Patient's medical records will be updated with the new prescription entered by the doctor

9.6 Exceptional flow:

9.6.1 If the doctor cancels the changes then the prescription will not be added into the patient's medical records.

10. Deactivate account (U10)

10.1 Introduction: Admin will be able to Deactivate the accounts of a particular doctor/patient.

10.2 Actors:

10.2.1 Admin

10.3 Pre-conditions:

10.3.1 Admin should be logged into their account

10.4 Post-conditions:

10.4.1 Patient/doctor's account will be Deactivate

10.4.2 Patient/doctor will not be able to log into their accounts

10.5 Basic flow:

10.5.1 Admin navigates to deactivate option in the patient/doctor's profile

10.5.2 Admin confirms deactivation

10.5.3 patient/doctor account will be deactivated

10.6 Exceptional flow:

10.6.1 If the patient/doctor account don't exist the account will not be deactivated

10.6.2 If Admin doesn't confirm the deactivation the account will not be deactivated

## 11. View patient/doctor profile (U11)

11.1 Introduction: Admin will be able to view the patient/doctor's profile

11.2 Actors:

11.2.1 Admin

11.3 Pre-conditions:

11.3.1 Admin should be logged into their account

11.4 Post-conditions:

11.4.1 Admin will be able to view the patient/doctor profile details

11.5 Basic flow:

11.5.1 Admin navigates to view patient/doctor profile

11.5.2 Admin searches for patient/doctor by entering their userID

11.5.3 Admin will be able to view their profile details

11.6 Exceptional flow:

11.6.1 If the user doesn't exist then the admin will not be able to view their details.

## Schema

