Software Requirements Specification

for

DocuMed

Version <1.0>

Prepared by

Group 11

Group Name: Ctrl+C Ctrl+V

Aarsh Walavalkar	220013	aarshaw22@iitk.ac.in
Tanush Goel	221131	tanushg22@iitk.ac.in
Ankit Kaushik	220158	kankit22@iitk.ac.in
Bhaumik Chawda	220292	bhaumik22@iitk.ac.in
Devansh Agarwal	220340	devanshag22@iitk.ac.in
Devansh Bansal	220341	bdevansh22@iitk.ac.in
Nilay Agarwal	220714	nilayag22@iitk.ac.in
Purav Jangir	220837	puravj22@iitk.ac.in
Shaurya Sharma	221007	shauryas22@iitk.ac.in
Shah Divit Ritesh	220995	divits22@iitk.ac.in
Ruthvik Tunuguntla	220924	ruthvikjs22@iitk.ac.in

Course: CS253

Mentor TA: Somesh Vas

Date: 26-01-2024

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Tanush Goel Ruthvik Tunuguntla Devansh Bansal Devansh Agarwal Ankit Kaushik Nilay Agarwal Shaurya Sharma Shah Divit Ritesh Purav Jangir Bhaumik Chawda Aarsh Walavalkar	First Draft	26/01/24

1. Introduction

1.1. Product Scope

Often when one goes to visit a doctor, one is asked his past medical history and any medications one might be taking at the moment. Most of us either don't remember our past conditions or worse, misreport our history which can lead to complications later. Our app aims to do that by keeping a record of a patient's past medical history and prescriptions along with all the lab reports, scans and diagnostics. The patient can then choose to reveal only the relevant information to the doctor ensuring privacy. This also simplifies the process of getting second opinions and changing hospitals as the entire medical history is always available to the patient.

1.2. Intended Audience and Document Overview

The Software Requirement Document (SRS) is a crucial document in the software development life cycle, and it is intended for various readers involved in different stages of the project. The primary audience includes:

- ***Developers:** The SRS serves as a detailed guide for developers, providing them with insights into the system's functional and non-functional requirements. It outlines the features, constraints, and specifications that need to be implemented, helping the development team understand the scope and expectations.*

- ***Customers/Users:** For customers or end-users, the SRS is a crucial document that communicates what the software will do, its features, and how it will benefit them. It sets the foundation for aligning customer expectations with the final product, ensuring that the delivered software meets their needs and requirements.*

- ***Testers:** Testers rely on the SRS to create test cases and validate that the software functions as intended. The document provides a basis for developing test scenarios and ensures that the testing process aligns with the specified requirements. It becomes a reference point for evaluating the software's correctness and performance.*

The Software Requirements Specification (SRS) is a crucial document in software development, highlighting its essential role in project success through clear requirements. It starts with an overview of the software's purpose and functionalities, followed by detailed descriptions of user interfaces to offer stakeholders insights into user interactions. The document concludes with use cases, illustrating expected software behavior in real-world scenarios. This structured approach guides stakeholders, fostering a collective understanding of the software's objectives and requirements, establishing a strong foundation for the development process.

1.3. Definitions, Acronyms and Abbreviations

Abbreviations/Acronyms	**Definitions**
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
JS	Java Script
REST	Representational State Transfer
SQL	Structured Query Language
API	Application Programming Interface
Doctor	Refers to both Doctors and Diagnostic Labs

1.4. Document Conventions

The document follows the IEEE formatting requirements

- Arial font size 11 is used.
- Document is single-spaced.
- 1'' margins are maintained.
- Bullet point ordering has been used as a listing type setting tool.

1.5. References and Acknowledgments

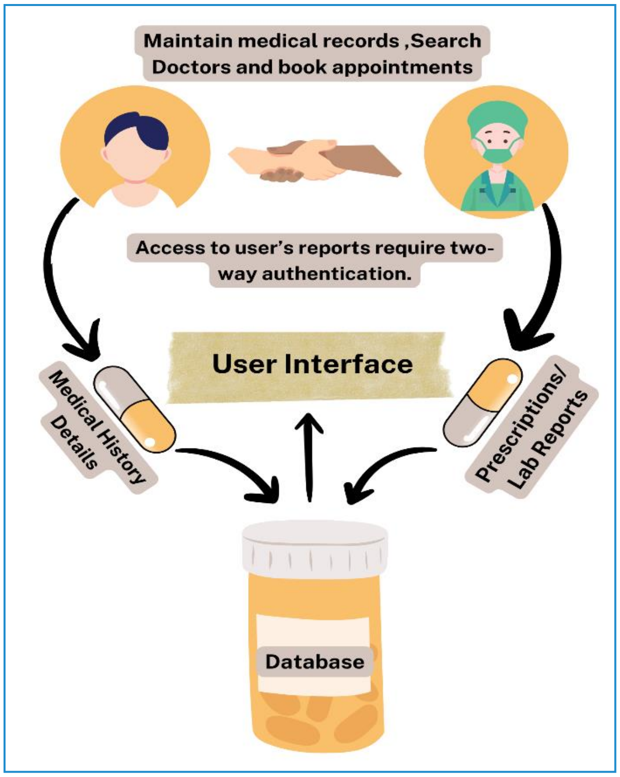
- *User Interface components were designed using Canva.*
- *Prof. Indranil Saha, for providing the SRS template document and teaching the required concepts.*
- *Our TA, Somesh Vas, for his valuable inputs from time to time in creating this document.*

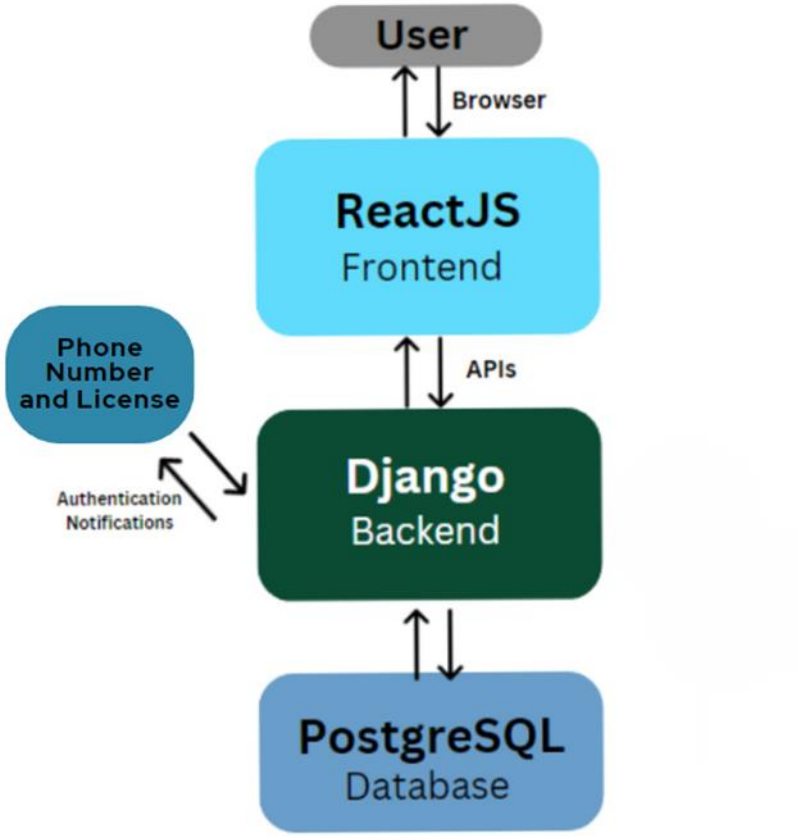
2. Overall Description

2.1. Product Overview

The envisioned software system emerges as a response to the prevalent challenges in the healthcare domain, where accurate and accessible patient information is crucial for effective medical care. Positioned as a standalone product, this application addresses the need for a reliable solution to manage patient medical records comprehensively. In the context of evolving healthcare practices, this software acts as an independent entity, rather than an extension or replacement of existing systems. The system's architecture encompasses major components such as a centralized database for storing patient medical history, prescriptions, lab reports, and diagnostics. A simple diagram illustrates the interconnections of these components, emphasizing the seamless flow of information within the system. The software is designed with adaptability in mind, ensuring compatibility with diverse healthcare infrastructures. This self-contained product aims to enhance patient-doctor interactions, mitigate the risk of inaccurate reporting, and facilitate smoother transitions between healthcare providers. The emphasis on privacy, selective information disclosure, and streamlined access to medical records positions this software as an innovative and indispensable tool in modern healthcare environments.

We plan on implementing the software where the user accesses the software through the browser and the software uses a Tech Stack which uses HTML, CSS, Javascript and ReactJS for the frontend requirements, Django for the backend requirements and PostgreSQL for database requirements.


Maintain medical records ,Search Doctors and book appointments
Access to user's reports require two-way authentication.
User Interface
Medical History Details
Prescriptions/ Lab Reports
Database



2.2. Product Functionality

- ***Registration and login:*** *The user on opening the application for the first time will be able to register himself/herself on the platform and login using his credentials the next time onwards.*
- ***Maintaining medical conditions and past diagnoses of the user:*** *The user can digitally maintain his medical conditions, known allergies, physical attributes and data of past diagnoses using the software, which will help doctors analyse the patient's condition better and offer suitable treatment and care.*
- ***Keeping records of patient's prescriptions, lab reports and medical certificates:*** *Once issued by a doctor, these documents are stored on the application in digital form in a secure and convenient way, while maintaining the privacy of the user.*
- ***Viewing available doctors for appointment and helping doctors list themselves up for the same:*** *The patients will be able to view the list of available doctors for a particular ailment using the software, and doctors will also be able to register themselves on the portal for patients to browse.*

2.3. Design and Implementation Constraints

Hardware Requirements:

- *The application requires a working computer or mobile network with internet access for it to function properly.*
- *The mobile device must have at least 2 GB of free RAM and minimum 0.1 Mbps of internet speed for the application to open.*

Software Requirements:

- *Our software is designed to work on Windows, Linux and MAC OS.*
- *It needs a web browser such as Google Chrome, Mozilla Firefox, Bing, Internet Explorer or Opera to work.*
- *The software and its dependencies may require an operating system which runs on 64-bit architecture and may also require the latest version of Android/IOS/Google Chrome for proper functioning.*

2.4. Assumptions and Dependencies

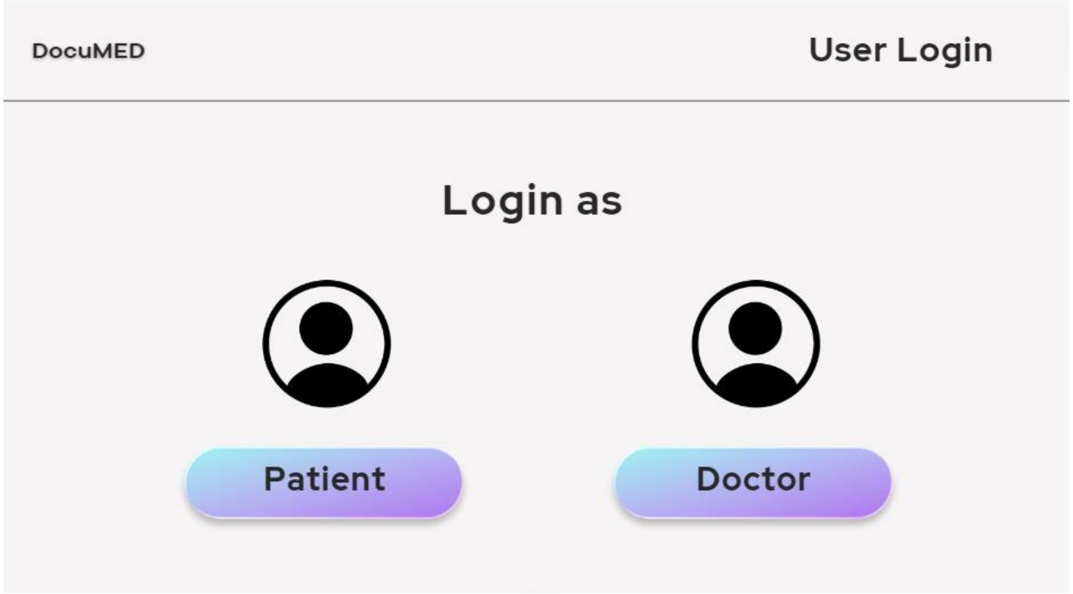
- *Users must be proficient in the use of mobile / web applications and have a basic understanding of their working.*
- *Users require a working email-ID for them to register and login. The email will be used for verification and authentication, as well as to receive updates from the software.*
- *Doctors must be required to check the website at regular times to cater to all patient requirements, as well as to update their working hours and to upload / edit specific patient related documents and prescription/ailment details.*
- *The web app shall be hosted on a web hosting service (in any cloud platform).*
- *The database used will be stored and accessed through a cloud platform.*

3. Specific Requirements

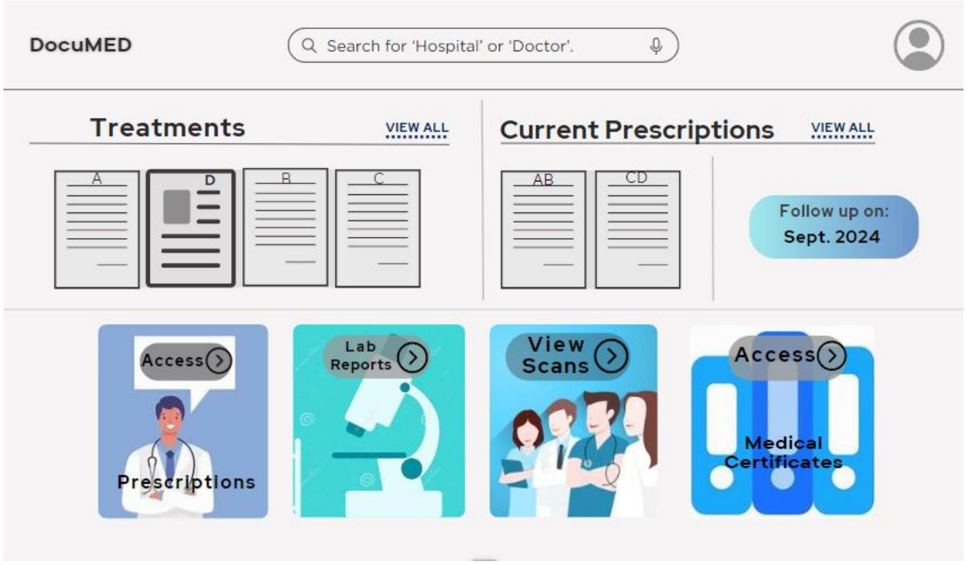
3.1. External Interface Requirements

3.1.1. User Interfaces

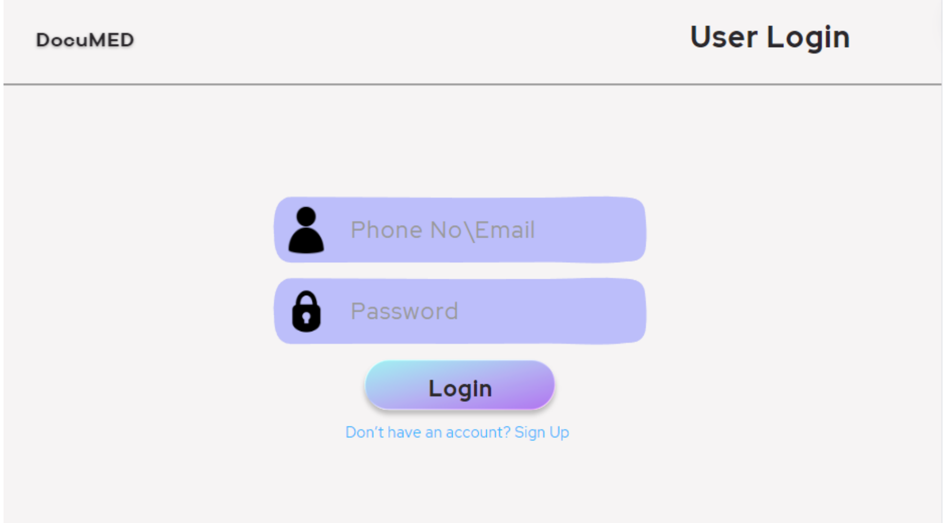
1. ***Entry Page:*** *- Upon accessing the website, a new user is presented with two options: "Doctor" and "Patient." The user selects their desired role and proceeds to sign in with the requisite credentials. For new users, the registration process is mandatory to create a new account.*



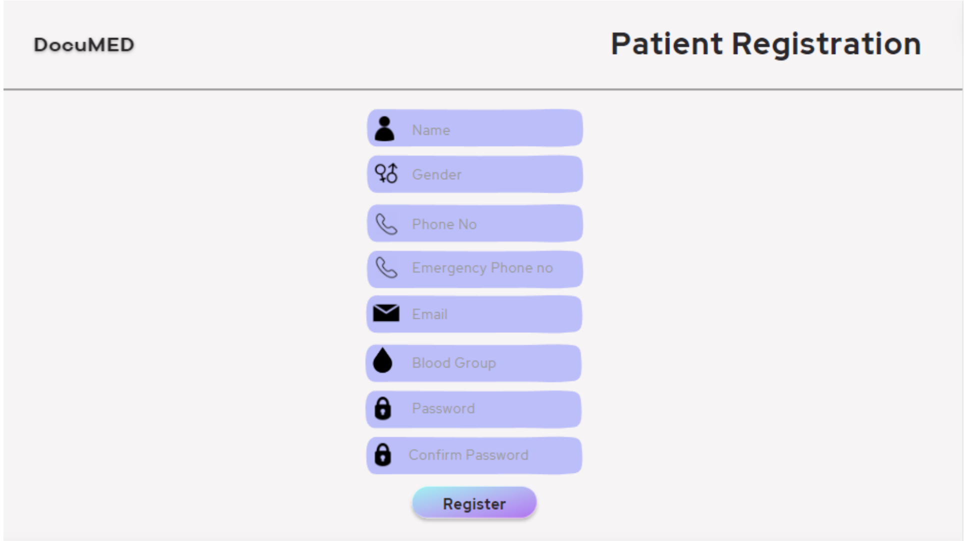
2. ***Home Page (Patient)**: - The homepage prominently features our product's logo in the top-left corner. In the top-right corner, users can find a profile icon, offering convenient access to profile editing functionality. Positioned directly under the logo is the "Treatments" icon, providing users with quick access to information about their current medical treatments. To the right of the "Treatments" icon, there are two icons, the "current medications" and for "follow ups" with a doctor. Within the middle-top section, a search bar empowers users to efficiently search for healthcare professionals, facilitating access to patient documents. Four additional icons are featured: "Lab Reports," "Diagnostic Reports/Prescriptions," "Scans," and "Medical Certificates."*



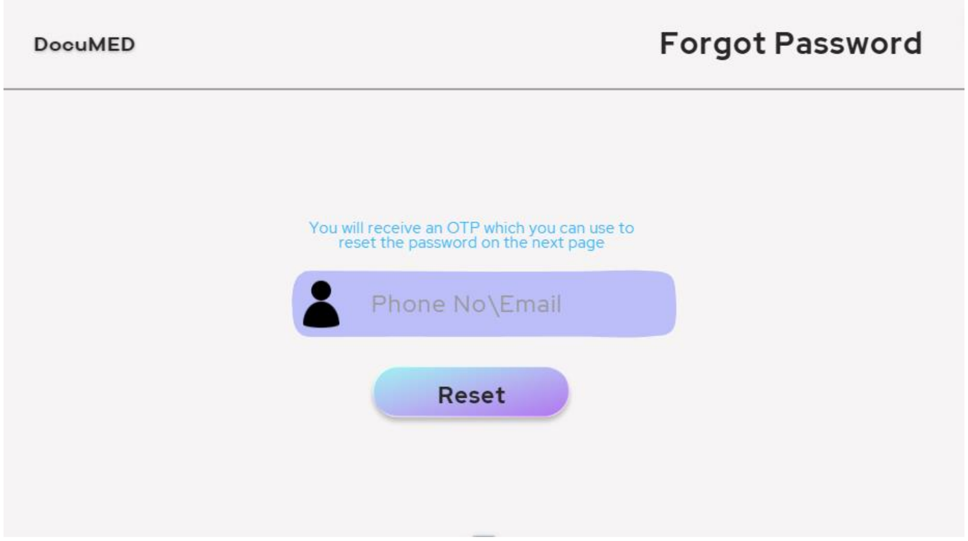
3. ***Sign In:** - The sign-in window prompts users to enter their phone number/email address and password. Upon correctly entering the credentials, users are directed to the home page. In the event of incorrect credentials, users are prompted to re-enter their password. Alternatively, they have the option to reset their password through the "Forgot Password" feature. This ensures a secure and user-friendly authentication process.*



4. ***Sign Up (Patient):*** *- The Sign-Up window for patients requires patients to provide their name, gender, blood group, phone number, emergency phone number, and email address and set a password for future logins. Additionally, patients are presented with the option to sign in if they already possess an account. Later patients are required to upload all their previous reports and enter the medications they are currently taking.*



5. ***Forgot Password:*** *- This window allows users to modify their password by providing either their phone number or registered email address. Upon entering the correct credentials, a confirmation email, inclusive of a link for password modification, will be sent to the user's registered email address.*

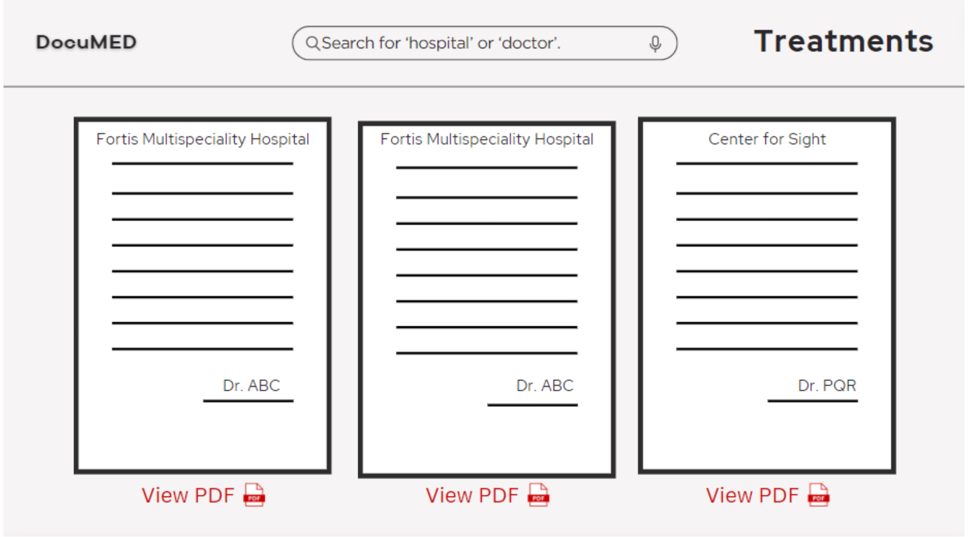


You will receive an OTP which you can use to reset the password on the next page

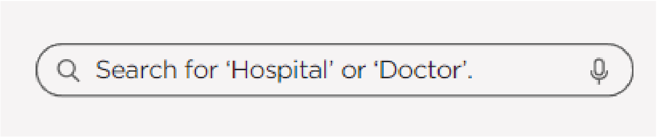
6. ***Document View(patient): -*** *This segment facilitates the ability for a patient to access and review all documents of a specified category in a chronological sequence. The standardized format for this segment will be consistent across four document types: laboratory reports, diagnostic reports/prescriptions, scans, and medical certificates.*

(Interface similar to Treatments below)

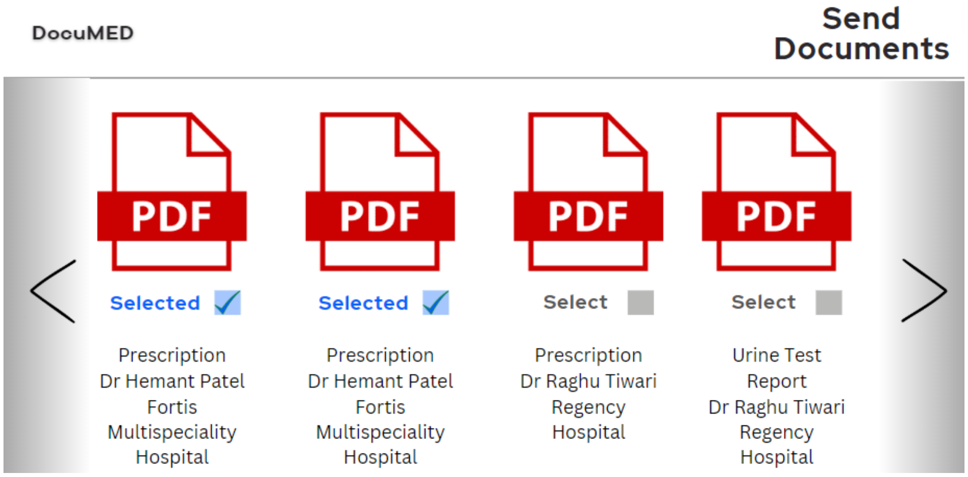
7. ***Treatments(patient): -*** *This segment facilitates the ability for a patient to access and review all documents uploaded by a specific doctor of all categories (laboratory reports, diagnostic reports/prescriptions, scans, and medical certificates) in a chronological sequence.*


DocuMED
Search for 'hospital' or 'doctor'.
Treatments
Fortis Multispeciality Hospital
Dr. ABC
View PDF
Fortis Multispeciality Hospital
Dr. ABC
View PDF
Center for Sight
Dr. PQR
View PDF

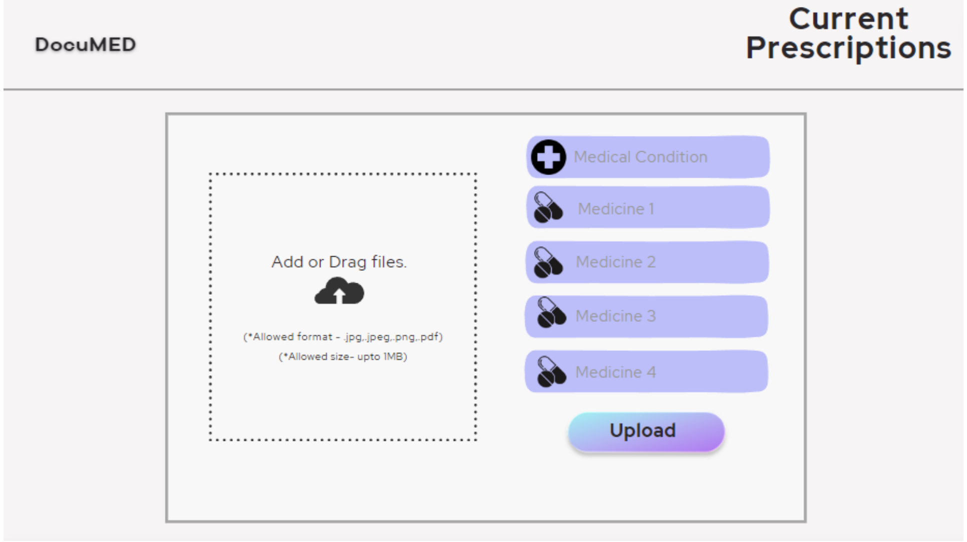
8. ***Search(patient): -*** *The Doctor Search and Interaction Module is designed to empower users with the capability to efficiently search for healthcare professionals based on specific criteria. It allows users to find doctors by their name or affiliated hospital, and subsequently engage with and grant them access.*



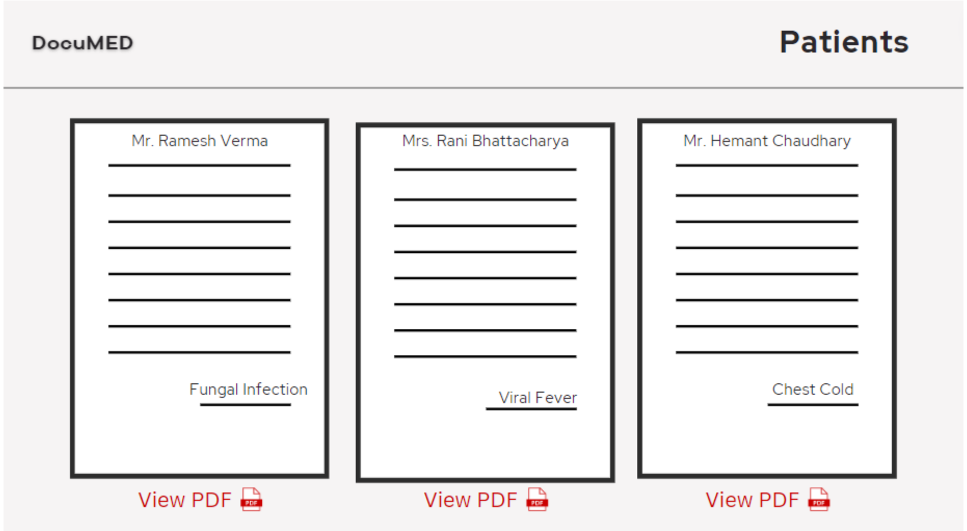
9. ***Provide Access(patient):*** *This option provides the patient with an option to allow his/her doctor to access the required documents/reports/diagnostics for better treatment. For protecting the patient's privacy, this will require two-way authentication between patient and doctor to prevent attacks from unauthorised users.*


DocuMED
Send Documents
PDF
Selected ✓
Prescription Dr Hemant Patel Fortis Multispeciality Hospital
PDF
Selected ✓
Prescription Dr Hemant Patel Fortis Multispeciality Hospital
PDF
Select □
Prescription Dr Raghu Tiwari Regency Hospital
PDF
Select □
Urine Test Report Dr Raghu Tiwari Regency Hospital

10. ***Add current medications(patient): -*** *The system shall furnish an intuitive and user-friendly interface, enabling patients to diligently input details pertaining to the medications they are presently prescribed. This shall encompass comprehensive data fields encompassing: Medication Name, Dosage, Frequency, Route of Administration, Commencement Date and Conclusion Date (if applicable).*



11. ***Document View(doctor): -*** The Document View interface for doctors is activated upon the doctor's selection of a patient. Within this interface, the doctor gains access to the comprehensive medical history of the selected patient. Subsequently, the doctor possesses the capability to update the upcoming follow-up date and upload documents of any of the four specified types, namely lab reports, diagnostic reports/prescriptions, scans, and medical certificates.

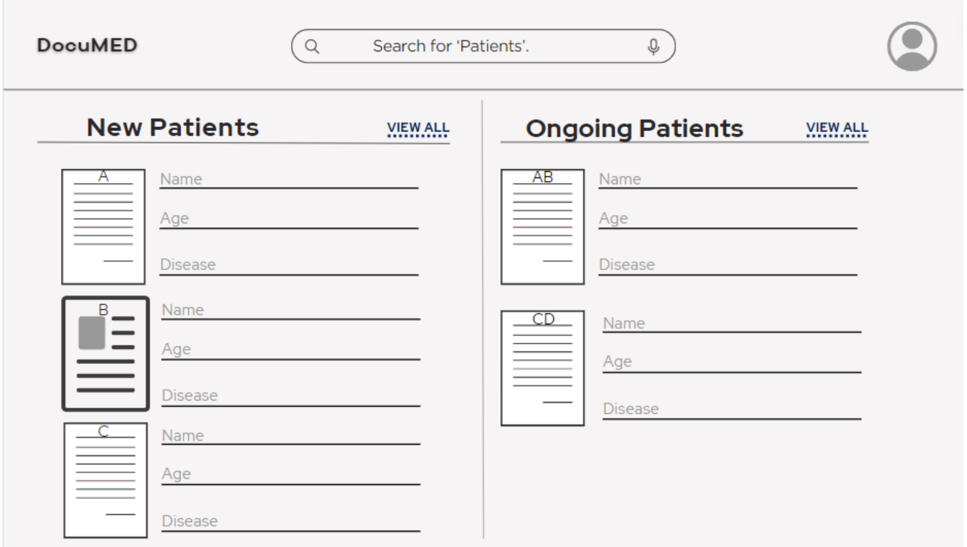


Mr. Ramesh Verma — Fungal Infection

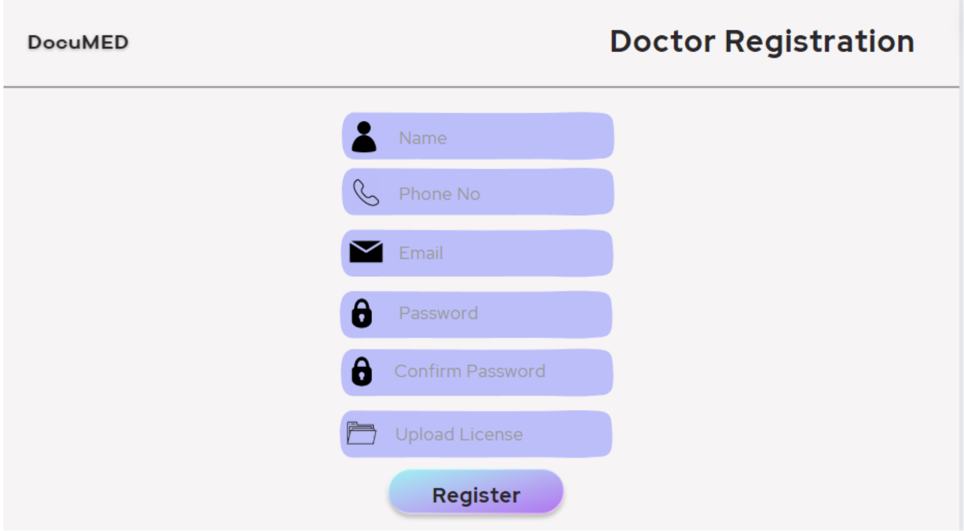
Mrs. Rani Bhattacharya — Viral Fever

Mr. Hemant Chaudhary — Chest Cold

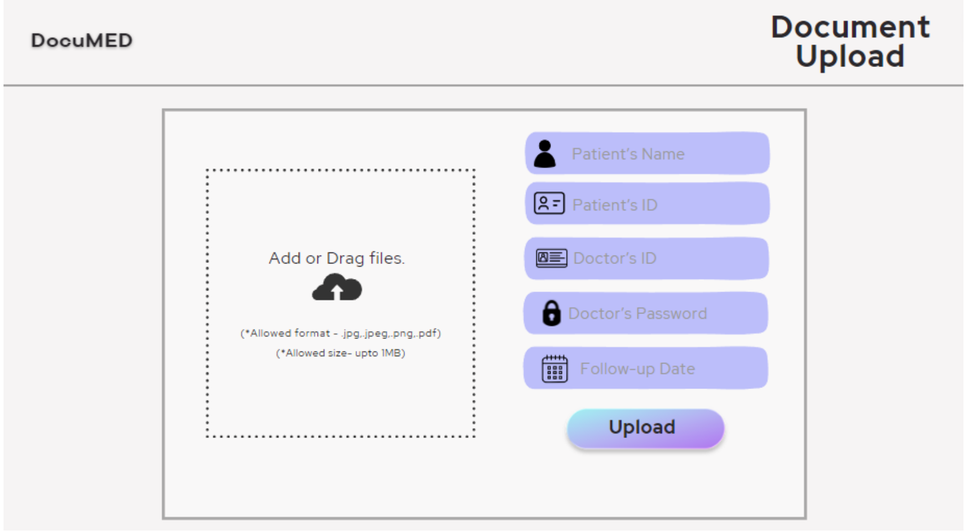
12. ***Home Page (Doctor)****: The Doctor User Interface features a robust search bar, doctors can swiftly locate specific patients based on names, identification numbers, or relevant keywords. The interface is organized into two distinct columns, "Ongoing Patients" and "New Patients," streamlining patient categorization. The "Ongoing Patients" column provides a quick overview of patients currently under the doctor's care, displaying essential details such as patient name, identification number, and a brief health summary. Simultaneously, the "New Patients" column facilitates prompt identification of recently added cases. The user interface prioritizes intuitive design, ensuring doctors can seamlessly click on any patient within these columns to access comprehensive patient profiles, thereby enabling a thorough examination of medical histories and records.*



13. ***Sign Up (Doctor):*** *- The Sign-up window for doctor necessitates providing essential details, including name, gender, phone number, emergency phone number, email address, specialization, associated hospital, working days, and working hours. Additionally, doctors are given the option to sign in if they already have an account.*



14. ***Add Prescription(doctor): -*** *Upon selecting a patient's profile, the doctor shall be capable of supplementing the patient's medical records with documents of the four types: lab reports, diagnostic reports/prescriptions, scans, and medical certificates. The document addition process shall adhere to a standardized format specified for each document type.*



15. ***Include a scheduled follow-up date:*** *In order to facilitate effective communication between the patient and the doctor, as well as streamline the process of scheduling subsequent appointments for routine check-ups, this feature allows for the designation of a specific date and time for a follow-up appointment, with the discretion for selection residing with the doctor.*

While adding prescription, doctor will be able to put a follow-up date
(see above for diagram)

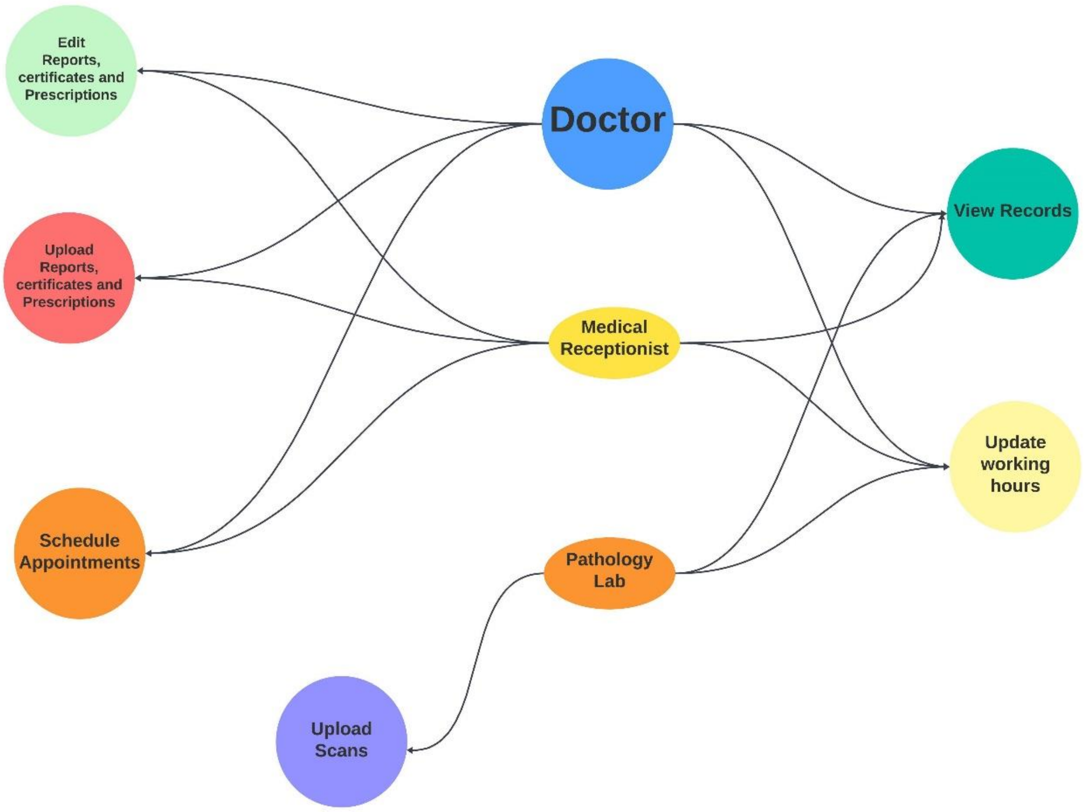
3.1.2. Hardware Interfaces

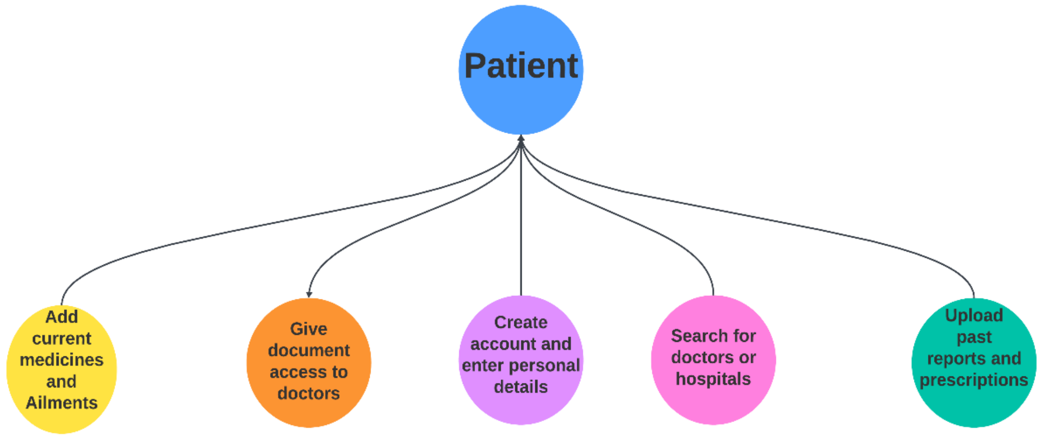
We shall be hosting the web app and working on the databases using a cloud platform. The users will be able to access the web app through their laptops, PC, mobiles etc.

3.1.3. Software Interfaces

- *The server side components including the database will be hosted in a Linux-based environment*
- *The client side components must be functional on modern web browsers such as Google Chrome, Safari, Mozilla Firefox etc.*
- *The database management system will be PostgreSQL.*

3.2. Functional Requirements




Patient
Add current medicines and Ailments
Give document access to doctors
Create account and enter personal details
Search for doctors or hospitals
Upload past reports and prescriptions

3.2.1. F1: Users choose between doctor or patient

user_type(is_doctor, is_patient)

3.2.2. F2: Doctors could create an account.

doctor_id = new_doctor(doctor_name, password, phone no., email_id, license_image)

3.2.3. F3: Doctors could add other details about their practice.

doctor_details(doctor_id, doctor_specialization , hospital_associated_with, working_days, working_hours)

3.2.4. F4: Doctors will need to verify their license.

verify_license(doctor_id, license_image)

3.2.5. F5: Doctors could change their account password.

update_password(doctor_id, password, new_password)

3.2.6. F6: Doctors could edit their working days and working hours.

update_timing(doctor_id, working_days, working_hours)

3.2.7. F7: Doctors can view patient documents

view_patient_docs(patient_id, is_accessible)

3.2.8. F8: Patient could create a new account.

patient_id = new_account(name, gender, blood_group, phone_no, emergency_phone_no, email, password)

3.2.9. F9: Patient can change their personal details.

update_details(patient_id, phone_no, emergency_phone_no, email, blood_group)

3.2.10. F10: Patient can change their account password.

update_password(patient_id, old_password, new_password)

3.2.11. F11: Patient can upload their past reports in PDF/Image format.

upload_reports(patient_id, past_report)

3.2.12. F12: Patient can upload their past prescriptions in PDF/Image format.

upload_prescriptions(patient_id, past_prescription)

3.2.13. F13: Patient can add their current medicines and ailments.

add_ongoing_medical_status(patient_id, current_medicines, current_ailments)

3.2.14. F14: Doctors can add Medical Certificate and Disability Certificate

add_certificate(doctor_id, patient_id, certificate_type)

3.2.15. F15: Doctors can add X-Rays, MRI and Ultrasound, CT scan and PET-CT

add_scans(doctor_id, patient_id, scan_type)

3.2.16. F16: Patients can view their documents and reports

view_self_docs(patient_id, is_accessible)

3.2.17. F17: Patients can search for doctors using either name or hospital or doctor_id

find_doctor(doctor_name, hospital_associated_with, doctor_id)

3.2.18. F18: Patients can give selective document access to a particular doctor

docs_access(patient_id, doctor_id, docs_id)

3.2.19. F19: Doctors can edit prescriptions for patients

edit_prescription(doctor_id, patient_id, is_accessible)

3.2.20. F20: Doctors can schedule next appointment

next_appointment(doctor_id,patient_id)

3.2.21. F21: Patients and Doctors can download the documents

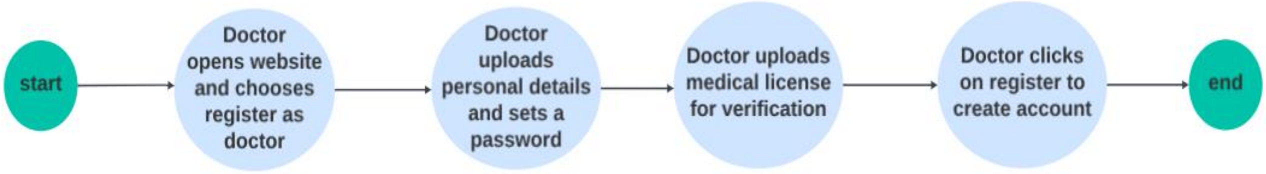
download_docs(patient_id)

3.2.22. F22: Patient and Doctor Login

verify_credentials(phone no, password)

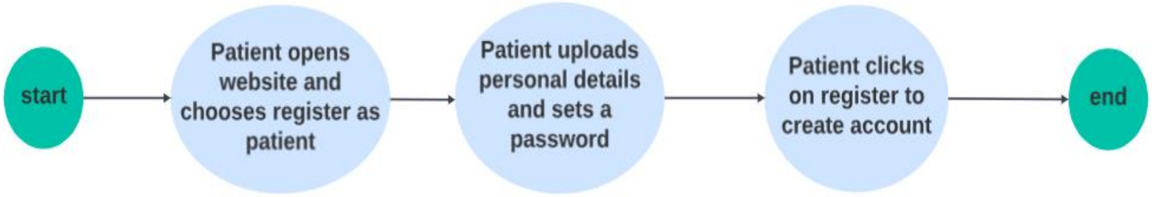
3.3. Use Case Model

3.3.1. Use Case #1 (Doctor Registration (U1))


start
Doctor opens website and chooses register as doctor
Doctor uploads personal details and sets a password
Doctor uploads medical license for verification
Doctor clicks on register to create account
end

Author	*Devansh Agrawal*
Purpose	*To register doctors on the platform*
Requirements Traceability	*Name of Doctor, Doctor's License, Generate password for him, other details about him like*
Priority	*High*
Preconditions	*Doctor must have a medical license*
Post conditions	*Doctor is registered on the platform*
Actors	*Doctor*
Exceptions	*Hospital's administrative staff*
Includes	*None*
Notes/Issues	*None*

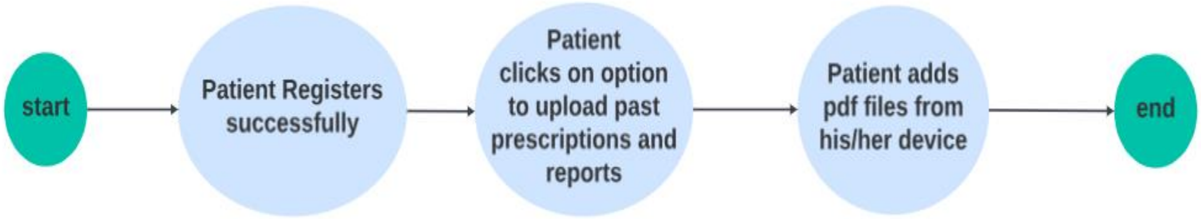
3.3.2. Use Case #2 (Patient Registration (U2))


start
Patient opens website and chooses register as patient
Patient uploads personal details and sets a password
Patient clicks on register to create account
end

Author	*Devansh Agrawal*
Purpose	*Create a patient entry in the database for registration*
Requirements Traceability	*Patient Name, Patient generic details, patient history*
Priority	*High*

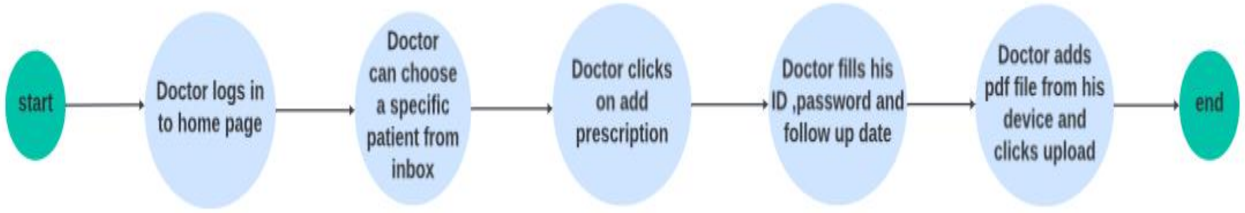
Preconditions	*Does not have an existing account*
Post conditions	*Patient now has registered on the platform with all general information and history*
Actors	*Patient*
Exceptions	*None*
Includes	*U3*
Notes/Issues	*None*

3.3.3. Use Case #3 (Uploading Past Prescriptions/Reports (U3))


start
Patient Registers successfully
Patient clicks on option to upload past prescriptions and reports
Patient adds pdf files from his/her device
end

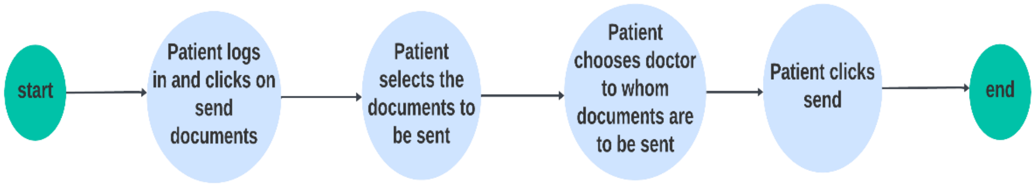
Author	*Devansh Bansal*
Purpose	*Uploading prior information while creating the account on the portal*
Requirements Traceability	*Documents of the patient*
Priority	*Medium*
Preconditions	*Documents available in proper format*
Post conditions	*All the uploaded documents will now reside in the database and easier to access*
Actors	*Patient*
Exceptions	*None*
Includes	*None*
Notes/Issues	*None*

3.3.4. Use Case #4 (Adding Prescriptions/Ailments (U4))


start
Doctor logs in to home page
Doctor can choose a specific patient from inbox
Doctor clicks on add prescription
Doctor fills his ID ,password and follow up date
Doctor adds pdf file from his device and clicks upload
end

Author	*Devansh Bansal*
Purpose	*Adding ongoing treatments and prescriptions in the patient database*
Requirements Traceability	*Names of the medicines and the ailments*
Priority	*Medium*
Preconditions	*Doctor has patient's permission for his treatment*
Post conditions	*All new prescriptions will be updated on the database*
Actors	*Doctor*
Exceptions	*Administrative staff may be able to add it*
Includes	*None*
Notes/Issues	*None*

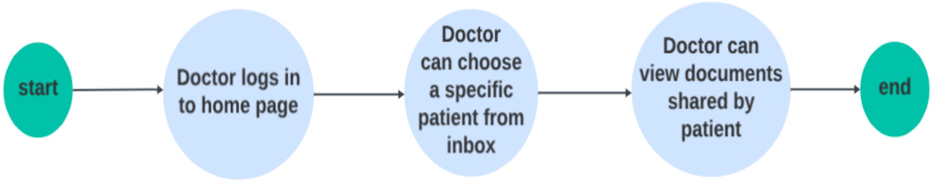
3.3.5. Use Case #5 (Send Documents to Doctor(U5))


start
Patient logs in and clicks on send documents
Patient selects the documents to be sent
Patient chooses doctor to whom documents are to be sent
Patient clicks send
end

Author	*Shaurya Sharma*
Purpose	*Send the doctor your medical history*
Requirements Traceability	*Documents in history,*
Priority	*Medium*
Preconditions	*None*

Post conditions	*Doctor gets access to all the documents that you have selected to be sent*
Actors	*Patient*
Exceptions	*None*
Includes	*None*
Notes/Issues	*None*

3.3.6. Use Case #6 (View Documents (U6))


start
Doctor logs in to home page
Doctor can choose a specific patient from inbox
Doctor can view documents shared by patient
end

Author	*Shaurya Sharma*
Purpose	*Doctor is able to view the documents sent to him via the patient*
Requirements Traceability	*Document*
Priority	*Medium*
Preconditions	*None*
Post conditions	*Doctor has view access to the patient's history*
Actors	*Patient, Doctor*
Exceptions	*None*
Includes	*U5*
Notes/Issues	*None*

3.3.7. Use Case #7 (Doctor adding report (U7))

Author	*Shaurya Sharma*
Purpose	*New reports in their specific formats to be updated in patient's database*
Requirements Traceability	*Report*
Priority	*Medium*
Preconditions	*Doctor has access to the patients documents*
Post conditions	*New reports are updated in the database*
Actors	*Doctor*
Exceptions	*Administrative staff may be able to add it on doctor's behalf*
Includes	*None*
Notes/Issues	*None*

4. Other Non-functional Requirements

4.1. Performance Requirements

Unstable Networks: *Application must be able to identify unstable internet connections. It must also display cooldown time and/or relevant error message instead of being stuck on the loading page in such cases.*
Intelligent search: *The search engine for the application must be able to show related results instead of simply matching keywords with the document names/types.*
Query time: *Average query time for the search of the document must be 200 ms on stable internet connections and must not exceed 5 sec anytime.*
Document Upload Size: *The documents being uploaded should not have a size more than 25MB. This limit is set taking into consideration the large file size of certain reports like Digital Radiography, Digital Mammography, Computed Radiography etc.*
Document Upload Time: *Under a stable and fast internet connection, upload time of a 25MB file should not exceed 15 seconds.*
Flexibility: *The website must work on all platforms (both mobile and PC OS).*

4.2. Safety and Security Requirements

Safety Requirements*:*

Mandatory Login: *Upon first access of the software, it is mandatory for the user to create his/her unique user ID and password so as to maintain security with respect to the data he/she will be uploading henceforth.*

Patient Privacy: *All of the patients' collected data can be viewed only upon login with a unique ID and password of the patient.*

Doctor Privacy: *Doctors' data can be viewed and accessed only upon login with a unique ID and password of the doctor.*

Document Upload: *The documents uploaded by an authorized person (doctor/nurse) shall be verified by a green tick to ensure their authenticity. Documents uploaded by the patient himself would not be verified but can be stored for future reference.*

Security Requirements:

Confidentiality: *The patient can choose what documents are to be displayed to the doctor and this can be viewed by the doctor only after entering his/her unique ID and password.*

Integrity: *It is made sure that the server or the software cannot harm the files, data or any personal information of the user, thus maintaining integrity.*

Availability*: The software and its data should never be restricted for authorized users*

4.3. Software Quality Attributes

***4.3.1 Reliability**: We plan to implement the following to ensure the reliability of our software*

- *Robust design through enhanced code quality*
- *Testing and beta testing*
- *Thorough up to date documentation*
- *Continuous integration and deployment*
- *Hence ensuring that the application will be accessible most of the time and will be dynamically updated would ensure the reliability of our software.*

***4.3.2 Performance:** Ensuring the performance of our software is crucial for delivering a positive user experience. Here are some methods we plan to incorporate:*

- *Using ReactJs for frontend, Django for backend and PostgreSQL for databasing requirements.*

***4.3.3 Interoperability**: Implies that the application will not interfere with any other application and will work seamlessly across various browsers. The following are the methods we will incorporate:*

- *Cross Browser compatibility, to allow the software to run across most of the used browsers by continuous testing.*
- *The software functions simply, whenever visiting the doctor he can select the documents to whom he would like to give access*

4.3.4 Usability:

- ***For Patients:***
  Easy to navigate graphical user interface for smooth functionality of the software.

- ***For Doctors:***
  Provide easy navigation to access, upload, and manage medical documents.

4.3.5 Testability:

- *Develop a comprehensive testing plan to verify the functionaility, security, and performance of the application.*
- *Include testing scenarios for different use cases, such as document uploads, access control, and data sharing.*

5. Other Requirements

Database Requirements:

- *The database should be designed with security in mind and should be able to protect sensitive data such as patient information and diagnostics.*
- *The database should be able to recover from failures or errors and should have a robust recovery system in place.*
- *The database should be designed to handle a high volume of read and write operations from multiple users concurrently.*

Internationalization Requirements:

- **Character Encoding:**
  Use UTF-8 or UTF-16 encoding for text data to support a wide range of characters. Ensure that the database, file handling, and communication protocols also support Unicode.
- **Language-Independent Code:**
  Write code that is independent of any specific language. Avoid hardcoding text strings directly into the code. Instead, use resource files or external configuration files.
- **Keyboard Input:**
  Allow for different keyboard layouts and input methods. Make sure that keyboard shortcuts are sensible across different languages and input methods.

Legal Requirements:

- *A backup or data preservation mechanism should be put in place. Data capacity should be planned to meet the storage requirement as per the mandated rules / laws.*
- *Should be able to harness any telecommunications-related connectivity like the Internet, LAN, WAN, WAP, CDMA, GSM or even Cloud Computing that will permit the various Electronic Medical Record of an individual to be integrated into a single lifelong electronic health record.*
- *The application shall comply with the provisions outlined in the Personal Data Protection Bill. Patient data, including personal and health-related information, shall be handled, stored, and secured in accordance with the specified privacy regulations. The system shall implement robust security measures to protect the confidentiality and integrity of patient data.*
- *The system shall incorporate mechanisms for obtaining informed consent from patients for the collection, storage, and usage of their health-related data. The application shall transparently communicate the purposes for which data is collected and seek explicit consent from users.*

Appendix A – Data Dictionary

Patient Class:

Element Name	***Description***	***Attributes***	***Operations***
Patient	*Patient will use the website to maintain personal medical history and send required documents to the doctor*	*Patient_ID=string* *Email_ID=string* *Password=string* *Name=string* *Phone number=string* *Gender=string*	*1)update_details()* *2) update_password()* *3)upload_reports()* *4)add_ongoing_medical_status()* *5) view_self_docs()* *6)upload_prescriptions()* *7) find_doctor()* *8) docs_access()* *9) download_docs()*

Doctor/Lab Class:

Element Name	***Description***	***Attributes***	***Operations***
Doctor/Lab Personal	*Doctor/Lab Personnel will be able to look at the previous medical history of the patient, update the current prescription and upload lab reports and scans.*	*Doctor_ID=string* *Email_ID=string* *Password=string* *Name=string* *Phone number=string*	*1)new_docotr()* *2) doctor_details()* *3) verify_license()* *4) update_password()* *5) update_timing()* *6)view_patient_docs()* *7) add_certificate()* *8) add_scans()* *9) edit_prescription()* *10) next_appointment()* *11) download_docs()*

Appendix B - Group Log

- *Offline meetings were hosted on a regular basis to discuss the software ideas, consistently monitor the completion of task assigned to every team member.*

- *Apart from this, a WhatsApp Group was also created to discuss daily updates over the documentation, along with brainstorming over some sudden change of thoughts.*

Date and Time duration of the Meet	*Meet Type*	*Ongoing discussions in the meet*
15/01/2024 *12:30 noon-2:00 pm*	*Offline Meet*	*Discussion over designing the software upon the chosen ideas along with finalizing the computer stack for building the software.*
18/01/2024 *1:15 pm-2:40 pm*	*Offline Meet*	*Discussion over how to create the software whilst listing over all the points to be mentioned in the SRS document related to the software. Tasks were equally distributed to each team member.*
21/01/2024 *20:00pm-21:30pm*	*Offline Meet*	*Meet to discuss and review over each team members' update on how much of everybody's task was completed. Finalizing some topics that were left to be discussed for the future.*
23/01/2024 *1:00 pm-3:30 pm*	*Offline Meet*	*Final meet amongst the team members as a robust draft of the SRS documentation was ready from the team's side and was ready to be discussed with the assigned TA.*
25/01/2024 *19:30 pm-21:00 pm*	*Offline Meet*	*Meet of the team with TA to proofread he documentation after having a brief discussion of the idea behind our software with him. Finalizing the SRS after some inputs and modifications given by the TA.*