Software Requirements Specification

for

DigiNotary

Version 1.0 approved

Prepared by Kashish Hemrajani, Khushal Dhanuka

Sardar Patel Institute of Technology, Mumbai

5th September, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
DigiNotary	5th September, 2022	First Version	1.0

1. Introduction

1.1 Purpose

The document presents a detailed description of `DigiNotary' . It will explain the purpose and features of the digitally designed notary system. The document will extend to showcasing the constraints of the website and how users will be uploading and consequently accessing their legal documents from the website, as well as how the system will react to external stimuli.

This document is intended for both the stakeholders and the developers of the system.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

The digital notary system designed here is intended for readers such as developers, project managers, users, testers and documentation writers. The rest of the SRS is organized by first defining the product perspective and functions,the operating environment and implementation constraints which is followed by the four crucial system features which includes the description of the feature and the stimulus and the response to the various user actions.

The reading sequence is as follows:

- Developers : The developer can first read through the product functions, the operating environment and design and implementation constraints which are crucial for him.There after he can learn about the software and hardware interfaces required for the development of the product and also the assumptions and dependencies of the system.
- Project Managers : The project manager has to know about the product perspective, functions and design. The implementation constraints and operating environment including additional dependencies part can be read by him towards the end.
- Testers : The testers can start reading the SRS with the system design and features and can further read about the user interface and the hardware and software interfaces required by the system
- Users : The user of the application first goes through the system features to learn about the product functionality and then through other requirements and product perspective or scope as per his interest.

1.4 Product Scope

- A notary (or notary public) is an official appointed by the state to administer oaths, take acknowledgments, certify documents and take depositions. The signature and seal or stamp of a

notary public is necessary to attest to the oath of the truth of a person making an affidavit and to attest that a person has acknowledged that he/she executed a deed, power of attorney, or other documents, and is required for making entries in public records.

- This software system has been digitally designed to give an official the ability to carry out the notarizing function electronically, which otherwise mandated a personal visit to the notary offices. This system will ensure that time taken for the entire process is reduced considerably and the documents are safe in a digital environment.
- The system works in a series of steps:
  - The applicant duly fills in the application and uploads documents to be notarized.
  - The admin then digitally verifies the documents and uploads them on the blockchain.
  - The Organization which requires access to the notarized documents (User), will request the admin for access to the same after entering the pin that the applicant provides.
  - A One Time Password (OTP) is sent to the applicant which completes the 2-factor authentication, following which the Organization is granted access to the notarized documents by the user.
  - In case of a query, the applicant or user may connect with the admin via a chatbot.

The objective of this project is to take this whole system over to a digital and very strongly secure platform that eliminates the problems in the existing system. With the use of Blockchain Technology, we would be able to achieve our desired goal in the following ways:

- No document loss can occur as a copy of the document is saved in every block in the blockchain.
- The system is controlled and managed by the people within the system. Thus, the chances of tampering with any document are completely eliminated. A particular change in a document is only accepted if and only if it is present in the majority number of nodes.

This system is a digital process and can be obtained at the click of a button. Thus minimizing efforts for carrying the hardcopies and efficient utilization of time.

1.5 References

The idea of the project has been referenced from:
https://www.forbes.com/sites/forbestechcouncil/2019/11/12/a-blockchain-based-digital-notary-what-you-need-to-know/

The style guides and references used in the designing of the user interface have been referenced from:
https://livenotary.com/

The market competitors and the online references used in the development of the ideas for the product have been referenced from :
https://www.g2.com/categories/e-notary
https://www.investopedia.com/best-online-notary-services-5085059

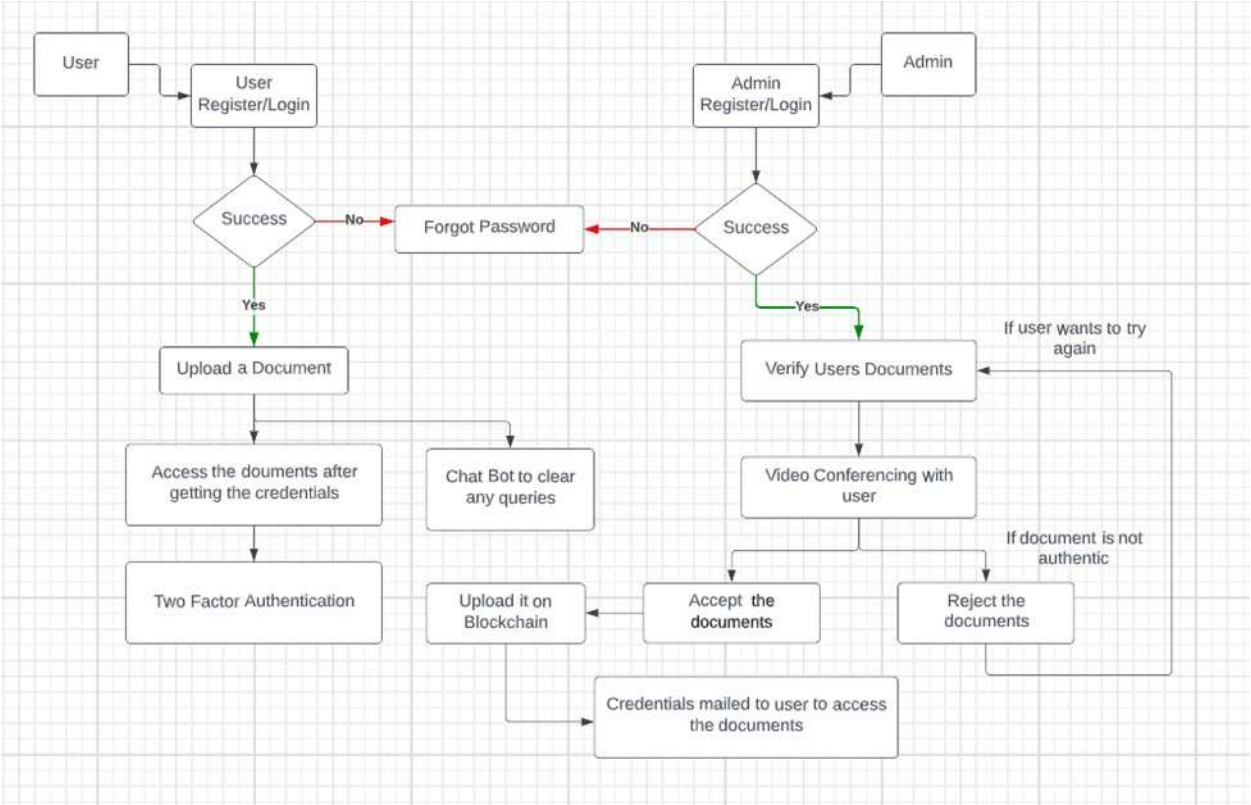
2. Overall Description

2.1 Product Perspective

Existing softwares and some important features they provide:-

Software	Features (Pros)	Drawbacks (Cons)
Notarize	Available 24/7 and 365 days per year Access via mobile app or tablet.	A bit more expensive than other options for notarizing
NotaryCam	Optimum for multiple users as it allows multiple signers from different locations Available 24/7 and 365 days per year NotaryCam boasts a 99.8% customer satisfaction rate for more than 1,000,000 online notarizations.	Few customer complaints about delayed appointments were observed Relatively more expensive ($79 price for notarizations outside of the U.S.)
OneNotary	Very fast; allows user to schedule 15 minute notary appointments Provides 100% money back guarantee	Less established company Notarized document available only for seven days

2.2 Product Functions



2.3 User Classes and Characteristics

User	Description	Features
Applicant	People who are going to upload their documents(both existing and new users)	Uploading legal documents, video conference with admin for authenticity check, chat with bot in case of query
Organization	Those requesting access to the documents uploaded by the customer	access to applicant's documents
Admin	Personnel authorized with managing the information and regulating user access	Check authenticity of documents and either accepts or rejects them, grants access of documents to organization, chat system

2.4 Operating Environment

For blockchain, we will be using Solidity as our Programming Language and would need a Windows 10 as it provides built-in Linux Subsystem.

For server side NodeJS would be used as backend, which would be interacting with the database to handle user requests. Any operating system supporting these languages would be appropriate for developing the website.

2.5 Design and Implementation Constraints

- Versions of OS: Only supported versions.
- Version of browser: All versions which support HTML, CSS and JavaScript.
- Language support: English.
- A stable internet connection is recommended.

2.6 User Documentation

No documentation yet

2.7 Assumptions and Dependencies

- Firebase: For authentication of users (https://firebase.google.com/docs/auth)
- Generate-password: extensive library for generating random and unique passwords. (https://www.npmjs.com/package/generate-password)
- User should have a functional webcam to interact with the admin.

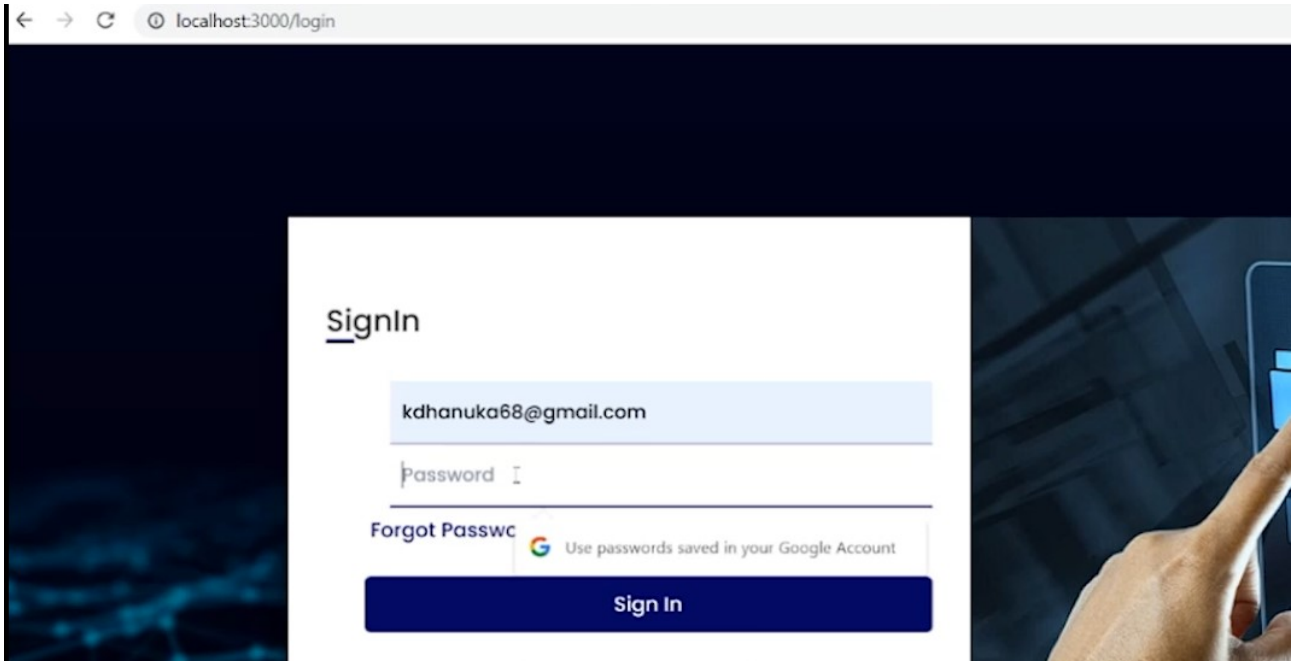
Assumptions: User has active internet connection to use the website and its features, and also for the video conferencing with the admin

3. External Interface Requirements

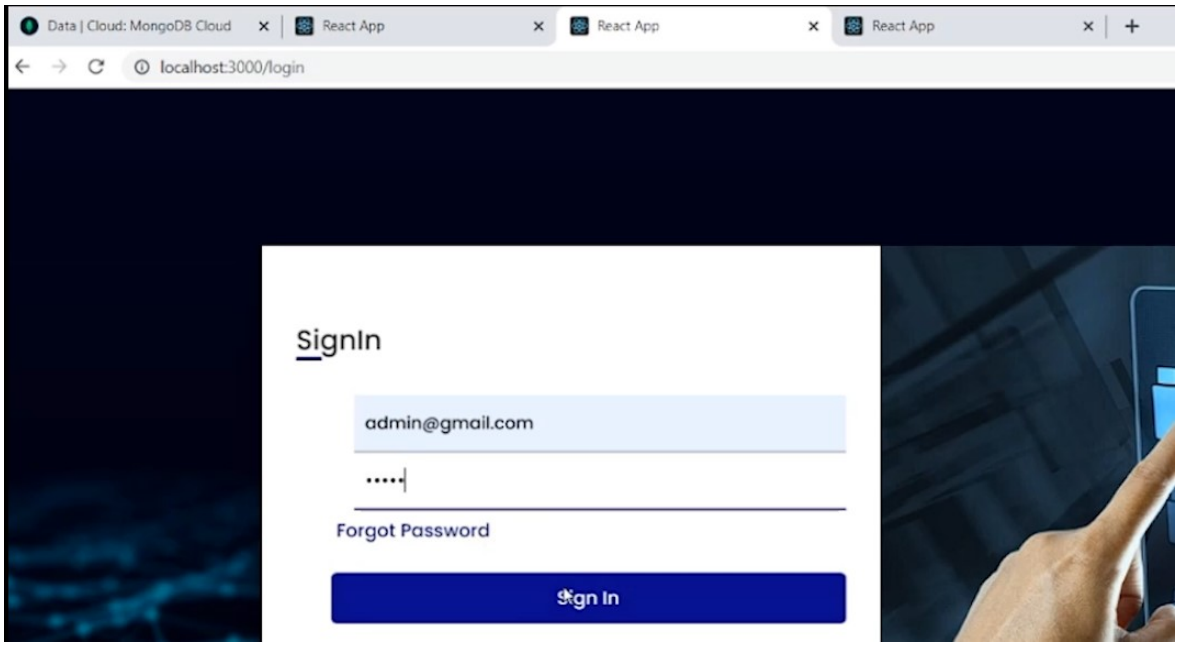
3.1 User Interfaces

Login Page

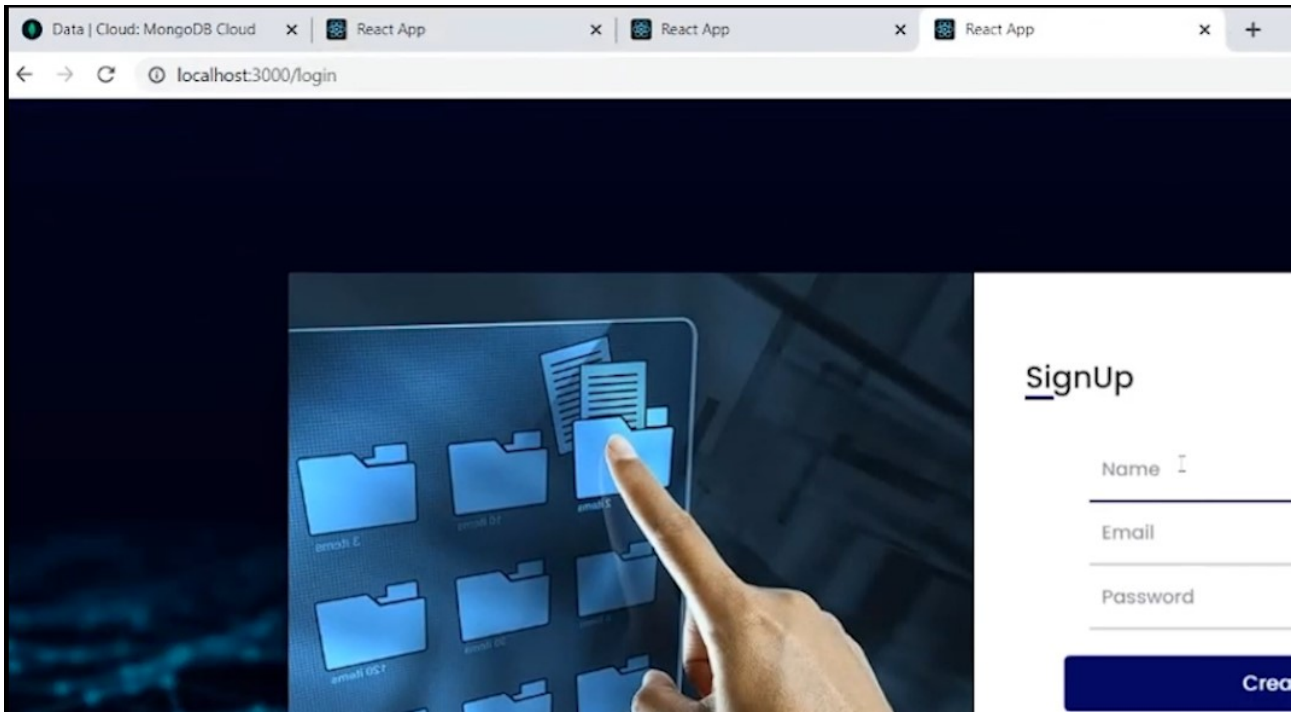
USER



ADMIN

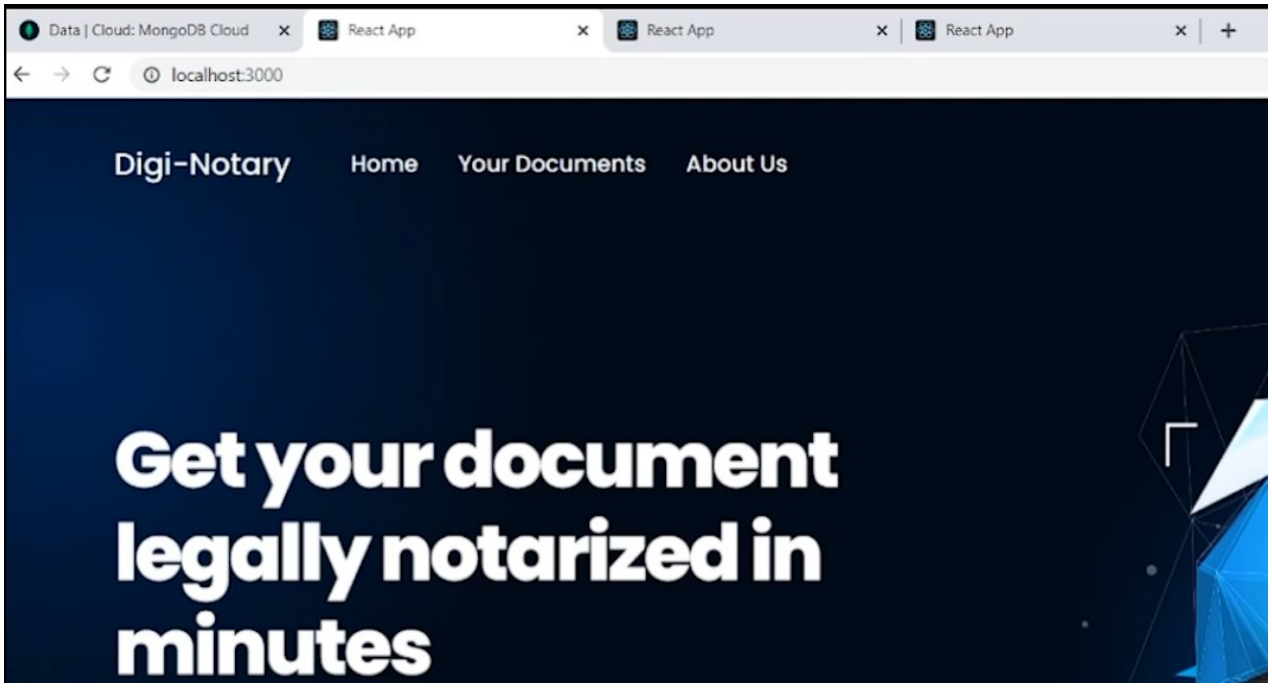


NEW USER

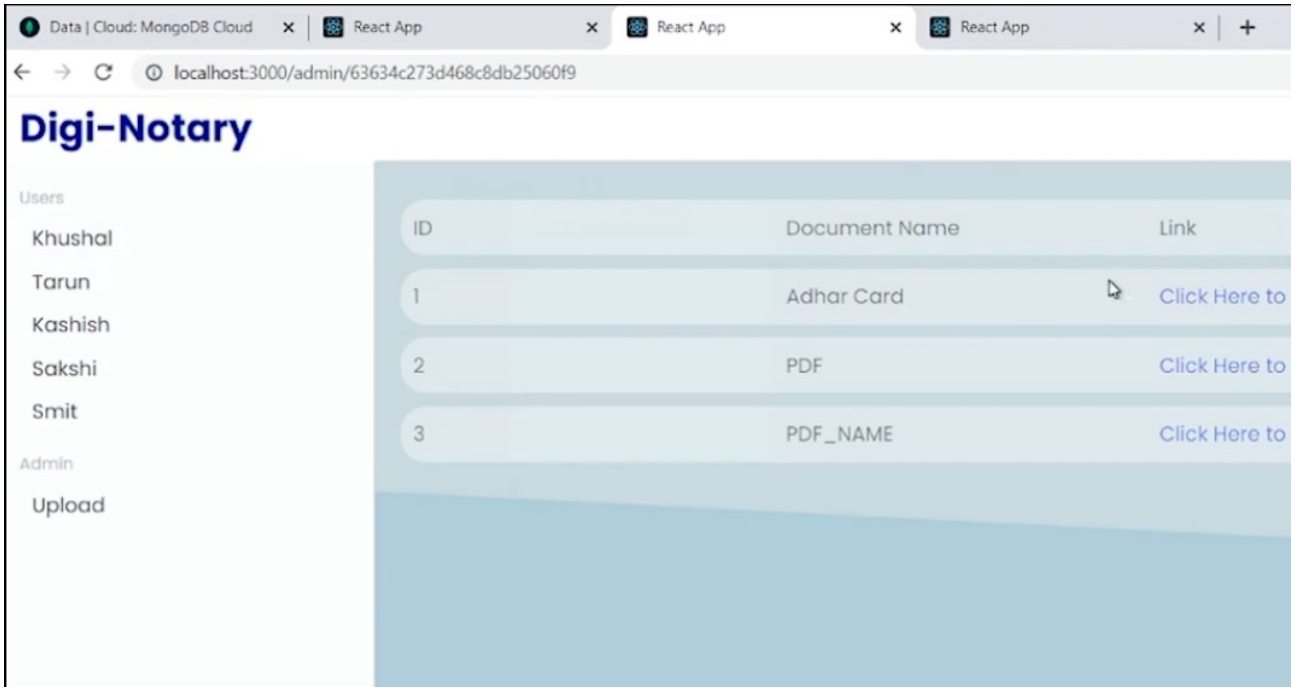


Landing Page

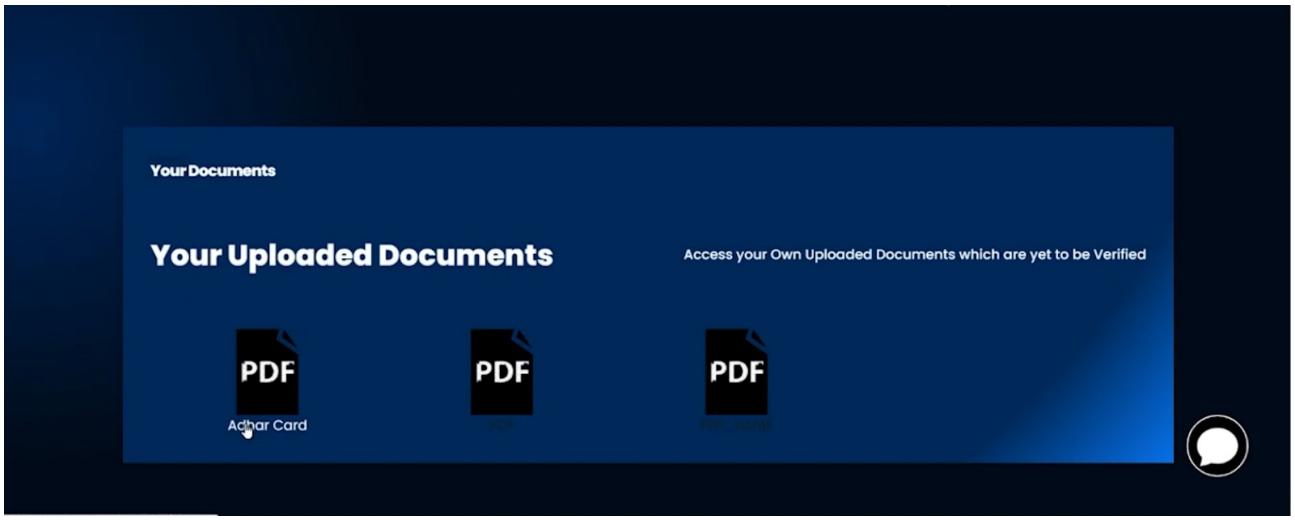
USER



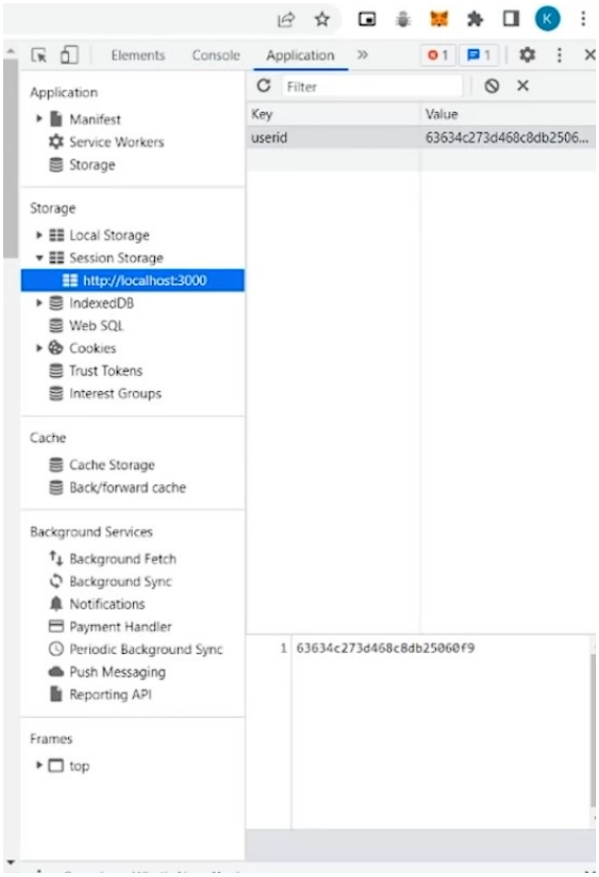
ADMIN



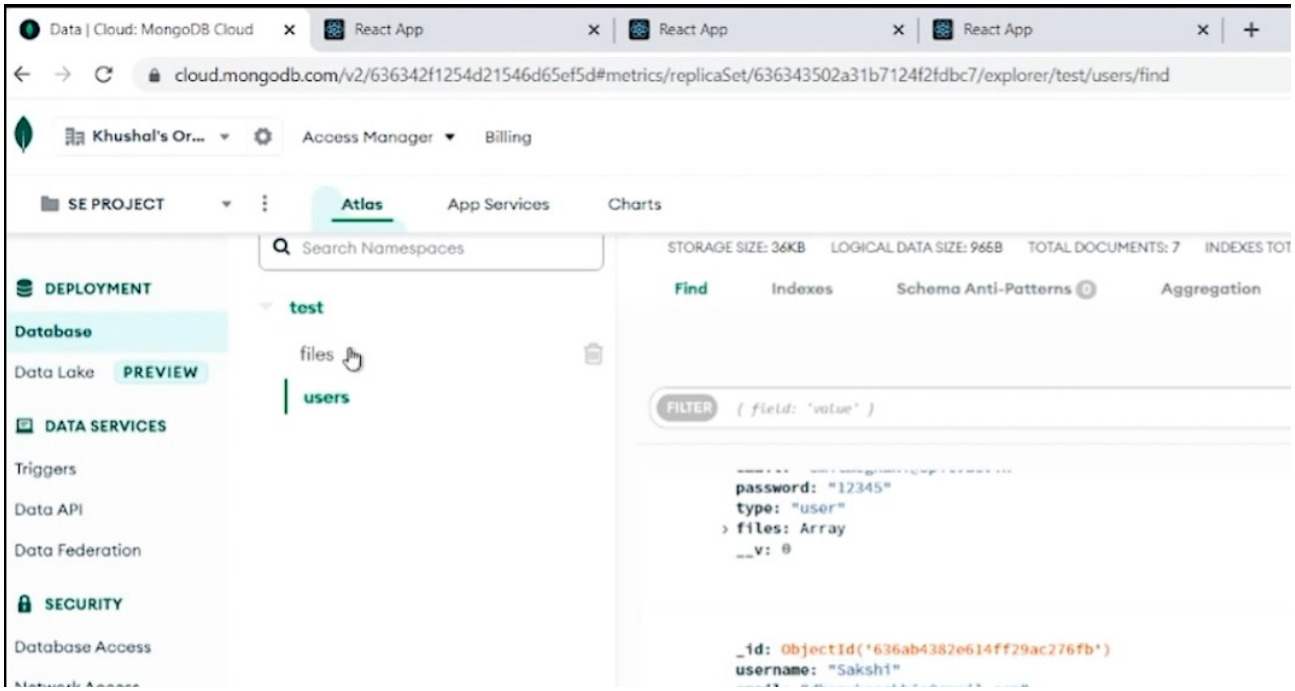
Your Documents Page (User):



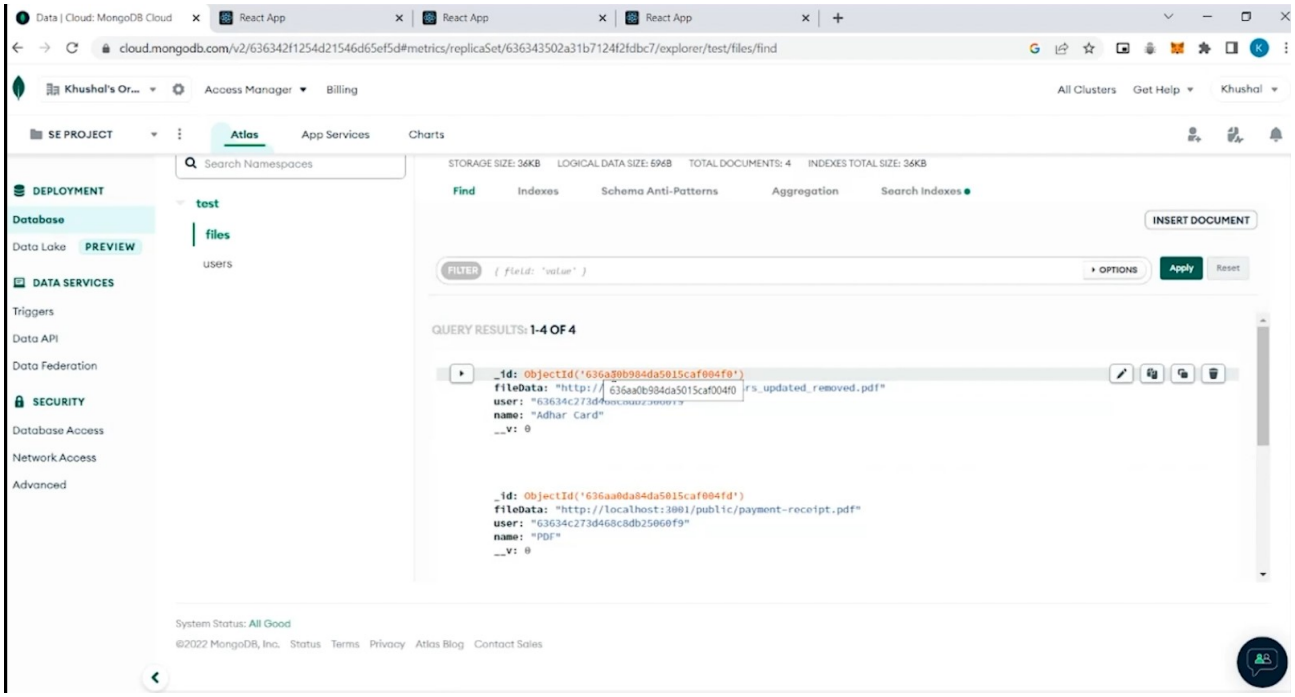
User Session:



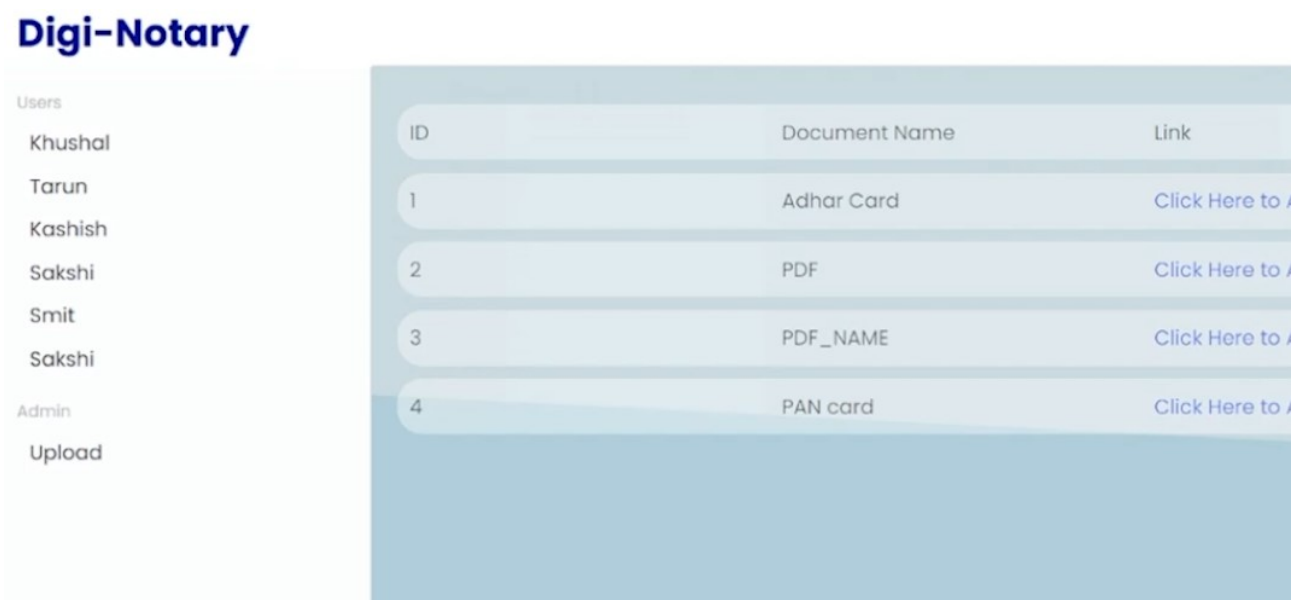
User Database:



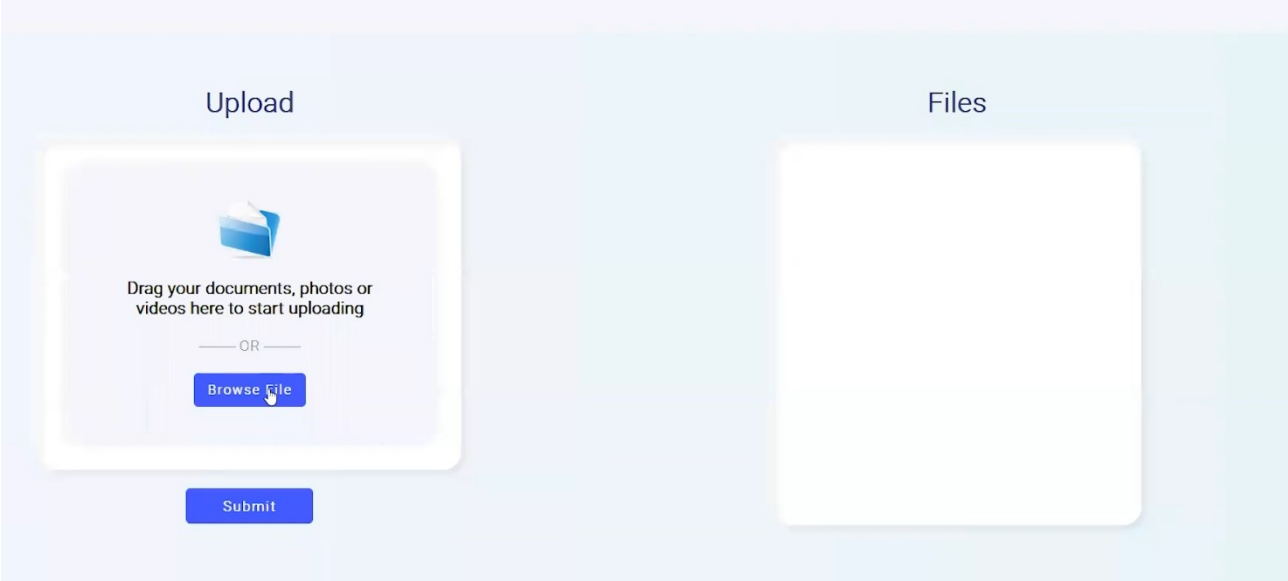
Files Database:



Chat Server:

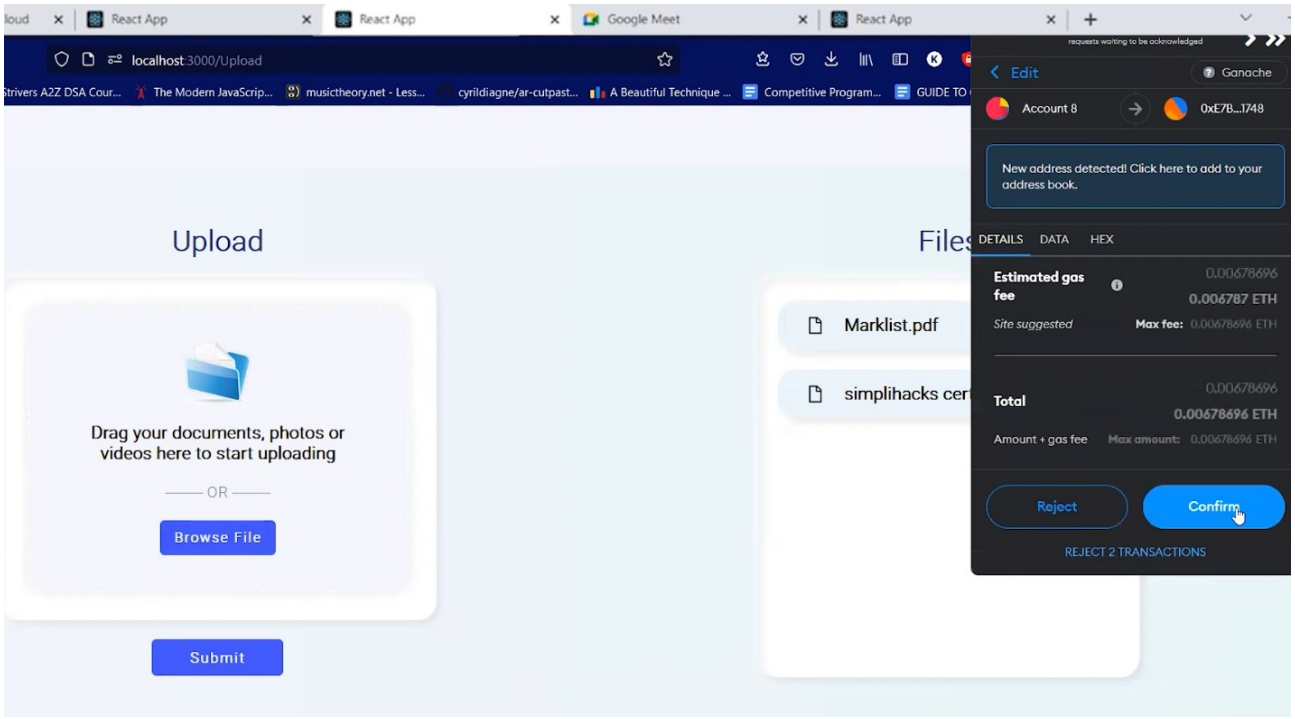


Admin uploads verified documents to blockchain:



Files Uploaded:

Confirm Transaction:



3.2 Hardware Interfaces

The application will run on all PCs, laptops and mobile devices which meet the browser and operating system requirements, and have a stable internet connection.

3.3 Software Interfaces

Libraries and frameworks used:

Node.js: Node.js is a back-end JavaScript runtime environment that runs on a JavaScript Engine and executes JavaScript code outside a web browser.

Express.js: Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.

Database

MongoDB: MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

3.4 Communications Interfaces

SMTP: Used for sending the verification link to the users email while registering.

HTTP: Used for basic communication between client and server. Client sends a request to which it may get a response or trigger a backend change through this protocol.

4. System Features

4.1 Uploading the Documents

4.1.1 Description and Priority

Applicant Side:

- Uploading the necessary legal documents which they must get verified by the concerned admin
- Communication with admin via chat bot in case any query arises

Admin Side:

- Receiving the uploaded documents from the admin panel
- Chat with the user to resolve any queries

4.1.2 Stimulus/Response Sequences

- Stimulus: The applicant proceeds to upload the document
  Response: The system responds with a browse file option from where the applicant can select the documents which are to be uploaded.

- Stimulus: Applicant sends a query to the admin via chat functionality
  Response: Admin responds to the user and clarifies the arisen issue

- Stimulus: File size exceeds specified limit
  Response: User is prompted to upload a compressed file

4.1.3 Functional Requirements

REQ-1: Applicant must possess a account to upload to the document
REQ-2: After uploading the document, user has to wait for the commencement of the verification process from the admin end
REQ-3: A database is mandatory to store the documents
REQ-4: Chat box history must be maintained

4.2 Video Conferencing

4.2.1 Description and Priority

Admin Side:

- To ensure the authenticity of the documents, video conferencing is done between the user and admin
- Clicks the “start call” button and he/she enters the meet via the provided link

Applicant Side:

- To prove authenticity of their documents, they communicate with the admin via video conference
- Clicks the button and he/she enters the meeting
- Chat with the user to resolve any queries

4.2.2 Stimulus/Response Sequences

- Stimulus: Admin clicks on the button for video conferencing
  Response: A meeting is created, and the admin enters into it

- Stimulus: User and admin click on "leave meeting" button
  Response: User and admin are disconnected when verification process is complete

4.2.3 Functional Requirements

REQ-1: The admin panel must provide the same meeting link to both the applicant and the concerned admin
REQ-2: The webcam of the user must be fully functional in order to verify the documents without difficulty
REQ-3: The applicant must possess stable internet connection so that the admin can efficiently verify the documents

4.3 Accessing the Document

4.3.1 Description and Priority

Admin Side:

- After receiving the documents, admin has to verify the authenticity of the documents and upload it on the blockchain
- After completion of the uploading process, they must send the credentials to the concerned user via email.

Applicant Side:

- Receives the credentials from the admin and provides it to the requesting organization, when asked for.
- Gives permission to the organization to access the associated documents so that no 3rd party member can access it unethically

4.3.2 Stimulus/Response Sequences

- Stimulus: Admin receives uploaded document from applicant
  Response: Verifies whether the document is authentic
- Stimulus: Admin suspects no foul play involved in the uploaded document, i.e. it is authentic
  Response: The document is accepted and put onto the blockchain

- ❖ Stimulus: Organization requests for access to a particular legal document
  Response: As the document is on the blockchain, the access process commences, and the organization requests the applicant for the pin to view the necessary documents

- Stimulus: Admin suspects that the document is fraudulent
  Response: The given document is rejected, and the user must re-upload and undergo the process again or visit a notary office in person for the document verification process

4.3.3 Functional Requirements

REQ-1: Admin should send the credentials to the user otherwise they won't be able to access the documents
REQ-2: Documents should be authentic to be accessed by the user or the organization

4.4 2-Factor Authentication

4.3.1 Description and Priority

Organization Side:

- To access the documents, they must request it from the user
- They send a link to user via email asking for the request

User Side:

- To give access of the documents, they click the link
- Access is provided to the organization

4.3.2 Stimulus/Response Sequences

- Stimulus: To gain access of the documents, organization sends the link to the user via their email
  Response: User must accept the access by clicking on the link received via email

4.3.3 Functional Requirements

REQ-1: Users Email-Id should be functional so that they can give access to the organization.
REQ-2: Users must check the organization who is requesting the documents before giving the access for the same.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The website will be made using NodeJS and would need a web browser with the following minimum versions (Chrome 6.0+,Safari 10.1+, / iOS Safari 10.1+,Edge 12+, Firefox ESR+, Opera) and Windows10 or higher operating system. All users with the minimum specifications can access the website without any problems.

5.2 Safety Requirements

To ensure that no 3rd Party organization can unethically access our documents, we have integrated 2-way authentication, wherein access to the users' documents will be provided only after consent is given. This is ensured by providing an OTP via email.

5.3 Security Requirements

- Blockchain provides a better secured platform where no one can tamper with the documents as they are stored in a linked-list format; if an attempt is made to do so,the other nodes will reject this request.
- To ensure that no information is compromised , Firebase authentication will be used so that a secure token is generated which is refreshed every hour so that security of the user is not compromised.
- Also the database will be secured so that there is no harm to security regarding details of any applicant or organization.
- Google authentication will also be provided which involves security of Gmail.
- The admin panel should be secured too because it will maintain all the data and so these routes must be protected as well.

5.4 Software Quality Attributes

The product will have an easy to use user interface for uploading and accessing the documents. It has adapted to the modern era of everything becoming online thereby making it easy for the user to access it. It is reliable as there is no loss of document, and it is secured using blockchain. It is portable and can run on any machine with a browser running.

5.5 Business Rules

The website will act as a platform for various user classes mentioned above. The applicant, i.e. the one who would be uploading the documents would be entitled to communicating with the admin via video conferencing for checking the authenticity of the uploaded documents and solve their queries, if any, by means of chat bot. The organization that is interested in accessing these legal documents would request access for the same. Ensuring that the documents are shared with the concerned organization only is supported by 2 factor authentication to prevent any fraudulent activities.

6. Other Requirements

Appendix A: Glossary

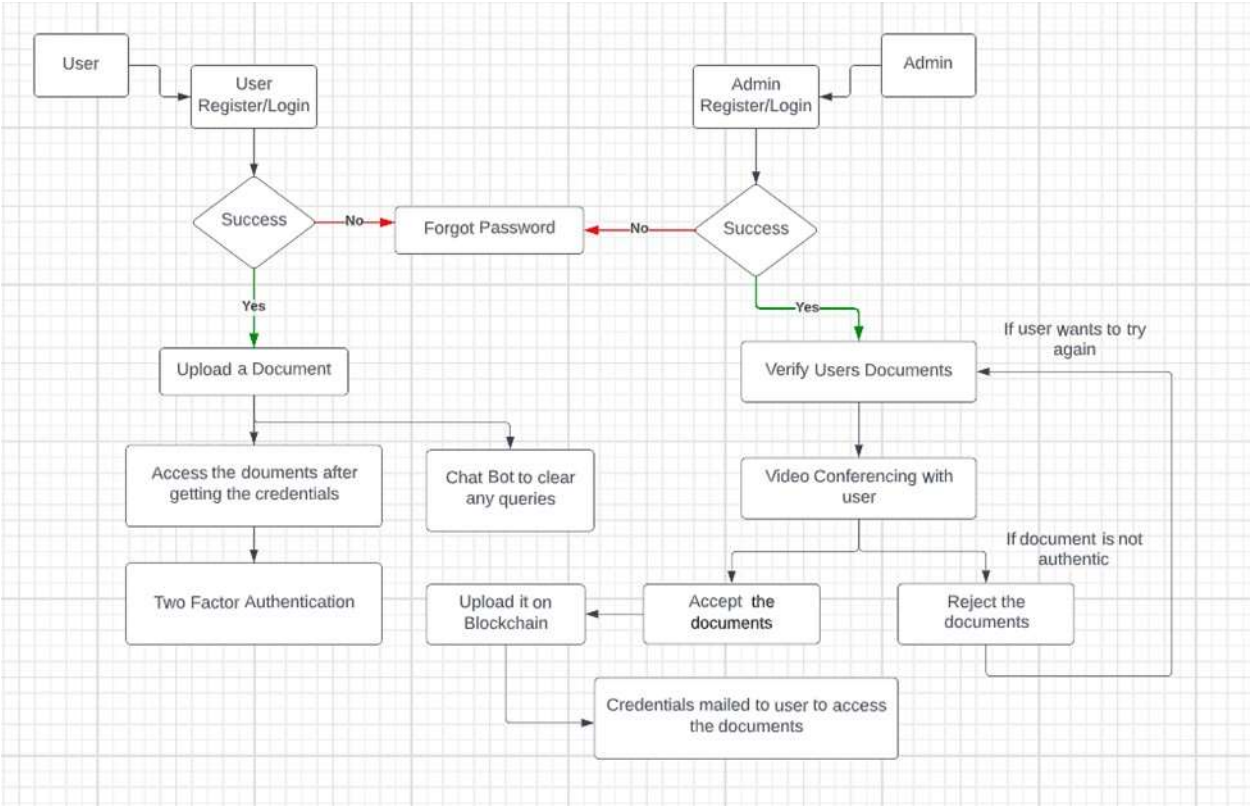
Firebase: Firebase is a platform developed by Google for creating mobile and web applications. It provides hosting, a realtime database, cloud firestore and many more services.

JS: JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. JavaScript is used to program the behavior of web pages

Node JS: Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine (i.e. V8 engine) and executes JavaScript code outside a web browser, which was designed to build scalable network applications. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

Express JS: Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js

Appendix B: Analysis Models



Appendix C: To Be Determined List

- Deciding details for user documentation
- Deciding the platform for the meet

IMPLEMENTATION LINK:
https://drive.google.com/drive/folders/1te0FHXXV0ycYM8cqqSGbiU1S7TLi7CFS?usp=share_link