HealthKard

Submitted in partial fulfillment of requirements
for the degree of

Bachelors in Technology

by

Umang Thadani
Roll No: 1914061,

Anurag Singh Name
Roll No: 1914058,

Dhruv Solanki
Roll No: 1914059,

Aayush Kapoor
Roll No: 1914066

Guide:
Prof. Era Johri

Department of Information Technology
K.J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch 2019-2023

K.J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to the University of Mumbai)

Certificate

This is to certify that this Synopsis entitled **HealthKard** submitted at the end of semester VII of LY B. Tech is a record for partial fulfillment of requirements for the degree of Bachelor of Technology in Information Technology.

Guide

Head of Department

Principal

Date:

Place: Mumbai-77

K.J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

This is to certify that this Synopsis entitled **HealthKard** submitted at the end of semester VII of LY B. Tech is a record for partial fulfillment of requirements for the degree of Bachelor of Technology in Information Technology.

Internal Examiner

External Examiner

Date:

Place: Mumbai-77

K.J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written thesis submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student **1914061**	**Signature of the Student** **1914058**
Signature of the Student **1914059**	**Signature of the Student** **1914066**

Date:

Place: Mumbai-77

Abstract

At the current stage, there is no unified framework in India that facilitates the storage of health records for all citizens. Although private companies/hospitals do solve this problem to some extent by providing e-health record services, they only cater to their own patients and customers. Therefore, there is a need to develop the foundations necessary for supporting digital health infrastructure to maintain health data in a decentralised and secure way. A few major advantages to this project will be ease of access, user consent for every sophisticated transaction, and portability across national borders.

In the healthcare system, a Blockchain network can be utilised to store and share patient data amongst hospitals, diagnostic labs, drug companies, and doctors. HealthKard (Blockchain applications) can precisely detect serious errors, including potentially deadly ones, in the medical industry. In the healthcare sector, it can therefore enhance the efficiency, security, and transparency of sharing medical data. Medical institutions can acquire insight and improve the analysis of patient information with the use of this application.

HealthKard supports a distinctive data storage pattern at the highest level of security, which can help eliminate the concern about data manipulation in the healthcare industry. It offers adaptability, connectivity, accountability, and authentication for access to data. Health records must be kept secure and private for a number of reasons. Blockchain helps prevent specific dangers and provides decentralised data protection for the healthcare industry.

A unique health identity will be generated via Smart Contracts, in the form of an NFT that will store the user's personal as well as health-related information, which is constant throughout to maintain the non-fungible nature of NFTs. This unique ID will help us map users' health records. Also, this will enable accessing personal health records, based on international standards, easily accessible to individuals and healthcare professionals and services providers, based on individual's informed consent.

Contents

List of Figures

Nomenclature

GUI	Graphical User Interface
API	Application Programming Interface
SDLC	Software Development Life Cycle
SRS	Software Requirements Specifications
SPMP	Software Project Management Plan
SDD	Software Design Document
UX	User Experience
UI	User Interface
JSON	JavaScript Object Notation
URI	Universal Resource Identifier
IPFS	Interplanetary File System

Chapter 1

Introduction

1.1 Product Overview

HealthKard aims to implement the following modules:

- Creation of a unique Health ID using Aadhaar Number
- Storage of Electronic Health Records (EHRs) mapped to Health Identity in the blockchain
- Integration of different sectors in the medical industry
- Encourage better administration of the health sector by utilizing health data analytics

1.1.1 Problem Definition

The purpose of this digital and decentralized health identity is to store all your health records in one place to make it convenient for users by avoiding the hassles of carrying physical reports or multiple digital reports, each from a different service.

The system is based on Blockchain and the idea of NFTs so that the unique health identity cannot tamper with and that the health records are stored in a secure manner without centralized control over your sensitive records. We will provide a comfortable User Interface to find patients, request access to health records, and ensure patients are able to grant access with his/her consent and let users store their Electronic Health Records (EHRs)

1.1.2 Motivation

Currently, there is no unified framework in India that facilitates the storage of health records of all citizens. Although private companies/hospitals do solve this problem to some extent by providing e-health record services, they only cater to their own patients/-customers. Therefore, there is a need to develop the foundations necessary for supporting digital health infrastructure to maintain health data in a decentralized and secure way. A few major advantages to this project will be ease of access, user consent for every sophisticated transaction, and portability across national borders.

Chapter 2

Literature Survey

1] The authors demonstrate the efficiency of employing blockchain technology for secure health information transfer and storage. An illustration of the health care system's decentralized architecture is provided in the article. A smart contract algorithm has been created for the healthcare industry. The study's findings demonstrate the usefulness of blocking technology for storing patient electronic medical records. According to system analysis, this technology's applications create a secure and naturally decentralized framework for efficient digital interaction.

2] The authors of this research use blockchain technology to create and assess a safe Digital Health Passport (DHP). To produce or view digital health certificates, users must first register with BigchainDB. Holders and verifiers have read-only access while the issuer has read-write access to upload test findings and ensure tamper-free. Holders can send the test results to verifiers via QR codes. This research study uses BigchainDB as the blockchain database in its execution. Then, DHP is assessed in light of the specifications for smart healthcare and the salient characteristics of blockchain technology. The evaluation and discussions present that the DHP is secured, immutable and traceable.

3] This article offers a cutting-edge solution for health identity, EHRs, and privacy using blockchain technology. This article offers a cutting-edge solution for health identity, EHRs, and privacy using blockchain technology. Individuals' identifying information and their health records are handled separately in this design of the blockchain-based health record system. An identity blockchain system houses the identity data. Through this system, healthcare providers, insurers, and governmental organizations can confirm a person's identity. The medical records are kept on a blockchain system for medical records.

4] This article describes the architecture of a Blockchain subsystem to manage access through Digital Identity to the Electronic Medical Record of patients. When a clinician has to access a patient's EHR data, the suggested solution is intended to address such a situation. The technique states that there should be two phases to the subsystem's development. The Smart Contract Part and the External Part are these two components. The first one will contain the module that controls patient data access and authorizations, as well as the policies that will direct how the subsystem functions. The external component will make it easier for users to access the application through a graphical user interface.

5] This paper describes a decentralized identity management system built on blockchain technology that enables patients and healthcare professionals to identify and authenticate themselves securely and transparently across various eHealth domains. Health identifiers allow patients and healthcare professionals to be identified in a unique way (healthIDs). A healthcare regulator certifies the identity traits, they are listed on the blockchain, and the identity owner stores them. The paper established smart contracts on a blockchain operated by the Ethereum consortium to facilitate authentication and identity processes.

6] In order to provide data privacy, data accessibility, and data interoperability for the healthcare-specific scenario, this article investigates the likelihood of portraying medical records. To access Smart Contracts supported by EHRs as information intermediaries, it created a substantial information infrastructure. Blockchain technology's decentralized structure helps make the EHR accessible across a wider network. By offering immutable, authentic, and accessible medical information, privacy, and quicker payments, the Blockchain enables the healthcare sector to undergo significant change.

7] In the proposed study, a blockchain-based architecture for the Electronic Health Record (EHR) would be framed. The suggested system consists of two steps, the first of which is block formation, and the second of which is data security utilizing the Elliptic Curve Cryptography (ECC) Algorithm. The system was developed using Ethereum. The patient, doctor, and administrator are the participants in the proposed system. Each participant has a position to perform, such as a patient, doctor, or administrator, as well as the ability to update, add, delete, and see information. Throughput, latency, and execution time are employed as the system's evaluation criteria.

8] Using KNN and CNN algorithms, this paper presented broad disease predictions based on patient symptoms. The first dataset used is a list of diseases and their symptoms that was acquired from the UCI machine learning website. The dataset is next subjected to preprocessing for cleaning, which involves eliminating commas, punctuation, and white spaces. Additionally, it serves as a training dataset. Then the feature was retrieved and chosen. Then, using classification methods like KNN and CNN, the data is categorized. We make reliable disease predictions using machine learning.

9] On three distinct disease databases (Heart, Breast cancer, and Diabetes) available in the UCI repository, authors have used several categorization methods for disease prediction. The UCI machine learning library makes the datasets used for creating the predictive models in this study available for download. The CSV-formatted data is imported and prepared for use. Machine learning algorithms like Logistic Regression, Decision Trees, Random Forest, Support Vector Machine(SVM), and Adaptive Boosting are used to predict the aforementioned diseases after data munging and attribute selection. Their accuracy is compared to determine which model is the best fit for that disease dataset.

10] In addition to providing statistics for improved planning of healthcare-related concerns at any level of administrative hierarchy, the proposed architecture offers a powerful platform for alert generation. A framework is presented that can be used to address public health-related activities like surveillance, registries, and keeping immunization records while addressing all the gaps we have found in the literature that are crucial for developing countries. This framework can be used in the context of acquisition and transmission of EHR from multiple sources. When adopted, the proposed framework offers a very useful platform for producing alerts and alarms as well as statistics for more effective planning of healthcare-related issues.

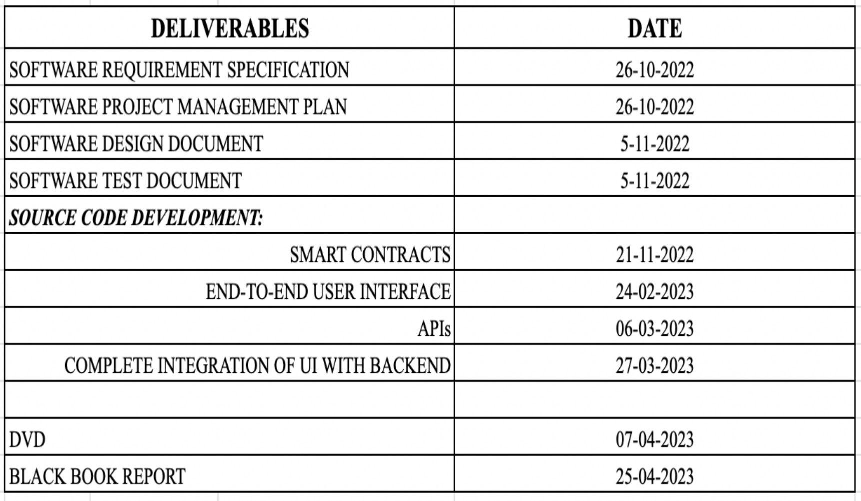
Chapter 3

Software Project Management Plan

3.1 Project Deliverables

After 12 weeks from the start of the project, the first module will be delivered. The entire product will be delivered within 7 months. The following are the project deliverables: Software Requirements Specification (SRS), Software Project Management Plan (SPMP), Software Design Document (SDD), Working Software Product with User Interface and Functions, DVD, and Black Book Report.

Figure 3.1: Deliverables



DELIVERABLES	DATE
SOFTWARE REQUIREMENT SPECIFICATION	26-10-2022
SOFTWARE PROJECT MANAGEMENT PLAN	26-10-2022
SOFTWARE DESIGN DOCUMENT	5-11-2022
SOFTWARE TEST DOCUMENT	5-11-2022
SOURCE CODE DEVELOPMENT:	
SMART CONTRACTS	21-11-2022
END-TO-END USER INTERFACE	24-02-2023
APIs	06-03-2023
COMPLETE INTEGRATION OF UI WITH BACKEND	27-03-2023
DVD	07-04-2023
BLACK BOOK REPORT	25-04-2023

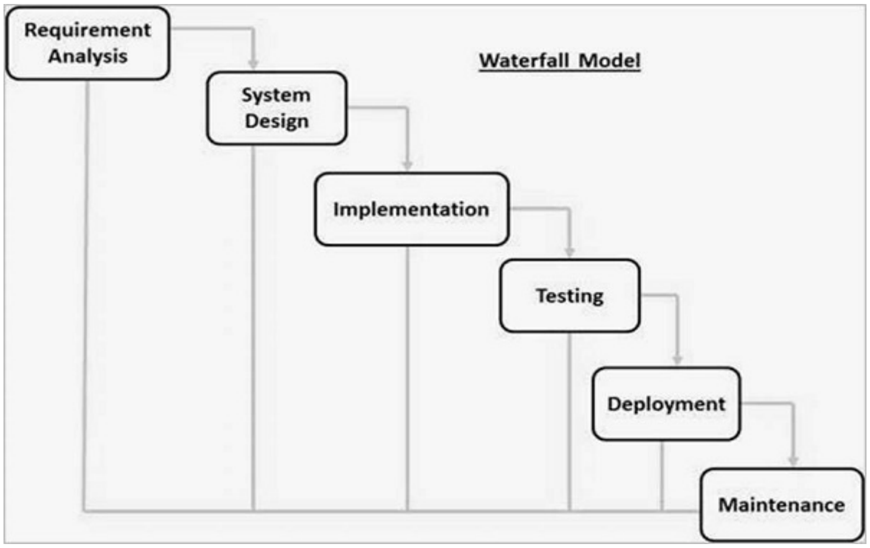
3.2 Project Organization

3.2.1 Software Process Model

The waterfall model depicts the software development process in a linear sequential flow. Due to this, it is also referred to as a linear-sequential life cycle model, which indicates that any development process steps can start after the previous one has finished. The stages are always done in this order. The model is called a waterfall because it progresses from one phase to the next logically.

In our project, we will first develop a unique health identity will be generated via Smart Contracts, in the form of an NFT that will store the user's personal as well as health-related information, which is constant throughout to maintain the non-fungible nature of NFTs since we cannot have a functional UI without this implementations. Then we will develop static webpages in React.js. And in final build phase we will integrate all. In our project, software requirements and technology stack is already decided, it will suitable to follow Waterfall model.

Figure 3.2: Waterfall Model



3.2.2 Tools and techniques

1. SRS, SPMP, SDD, and STD Document: Overleaf
2. Blockchain: Solidity and Metamask

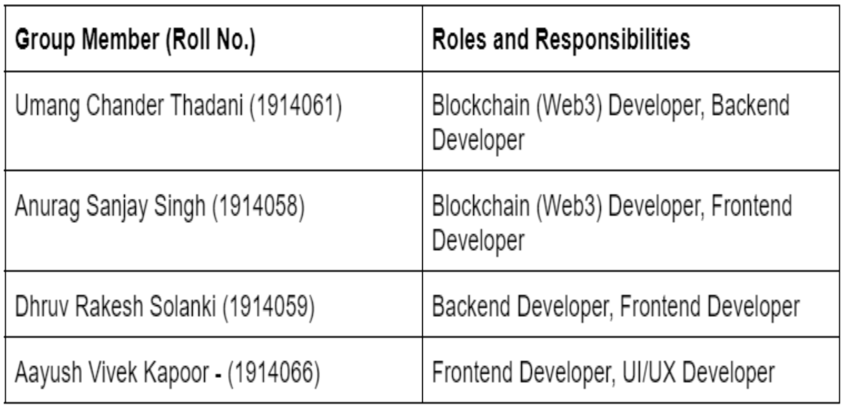
3. Back End: Django REST Framework

4. Database: PostgreSQL

5. Front End: React.js and Tailwind CSS

3.2.3 Roles and Responsibilities

Figure 3.3: Roles and Responsibilities



Group Member (Roll No.)	Roles and Responsibilities
Umang Chander Thadani (1914061)	Blockchain (Web3) Developer, Backend Developer
Anurag Sanjay Singh (1914058)	Blockchain (Web3) Developer, Frontend Developer
Dhruv Rakesh Solanki (1914059)	Backend Developer, Frontend Developer
Aayush Vivek Kapoor - (1914066)	Frontend Developer, UI/UX Developer

3.3 Project Management Plan

3.3.1 Tasks

1. Requirement Anaylsis

Description

Following elicitation, requirement analysis is an important and necessary activity. To create consistent and unambiguous requirements, we analyse, refine, and scrutinise the gathered requirements. This activity goes over all of the requirements and may show a graphical representation of the entire system. The project's understandability is expected to improve significantly following the completion of the analysis. We can also use the customer interaction to clear up any points of confusion and determine which requirements are more important than others.

Deliverables and Milestones

Completion of SRS Document and Synopsis

Resources and Dependencies

Stakeholders must be present to give inputs in order to obtain desired results, Whitepapers and websites.

Risks and Contingencies

Absence of stakeholders may lead to ambiguity in requirements.

2. Identification of Tasks and Subtasks

Description

Project management is heavily reliant on tasks. It is critical to know how to manage tasks effectively and efficiently if you want to complete your project efficiently and effectively.

Deliverables and Milestones

Tasks and subtasks identified and Risks evaluated with Task Sheet

Resources and Dependencies

SRS Document.

Risks and Contingencies

We must ensure that no tasks are left behind unidentified.

3. Schedule Preparation

Description

Another important aspect of project management is scheduling. Scheduling is important when it comes to all of the tasks and activities that make up the project as a whole. Project management can be done in a variety of ways, ranging from the more traditional to the most cutting-edge software. The goal of project management is to achieve the project's goals successfully, on time and on budget, whether internally or externally for a client.

Deliverables and Milestones

Completion of SPMP Document and Gantt chart

Resources and Dependencies

SRS and Task Sheet

Risks and Contingencies

We must ensure that schedule is foolproof.

4. Identifying Goals and Priorities

Description

We can handle a small problem at a time, but for a larger problem, divide the problems and conquer the problem means breaking the problem down into smaller pieces and capturing each piece separately.

Deliverables and Milestones

Basis for SDD is ready

Resources and Dependencies

SRS and SPMP Document

5. Constructing Designs

Description

The first step in the SDLC (Software Design Life Cycle) is software design, which shifts the focus from the problem domain to the solution domain. It tries to specify how to meet the requirements outlined in the SRS.

Deliverables and Milestones

Completion of SDD

Resources and Dependencies

SRS and SPMP Document, and Goals must be identified

Risks and Contingencies

We must ensure that all designs are constructed carefully.

6. UI Development

Description

This task involves creation of different modules for different webpages along with the styles and UX so that the UI looks clean and user friendly

Deliverables and Milestones

Complete Front End is ready

Resources and Dependencies

1. React.js
2. SRS, SPMP and SDD Document

Risks and Contingencies

There may be some instances where the UI crashes or has a loophole for unauthorized data access

7. UI Testing

Description

Testing the user interface (UI) is an important part of the software testing process. To ensure that applications have the desired functionalities and are user-friendly, they must be tested.

Deliverables and Milestones

Testing is completed on the UI part.

Resources and Dependencies

UI must be ready.

Risks and Contingencies

We must ensure that all test cases are considered.

8. Functional Development

Description

All requirements as mentioned in the SRS Document are implemented.

Deliverables and Milestones

Software Development is completed.

Resources and Dependencies

1. Django REST Framework
2. PostgreSQL
3. SRS, SPMP and SDD Document

Risks and Contingencies

There might be times when the product fails to deliver its purpose. The solution is to try and make the software fail-proof

9. Functional Testing

Description

The last step involved in wrapping up the Software is testing all of its functionality.

Deliverables and Milestones

Testing is completed on the entire product.

Resources and Dependencies

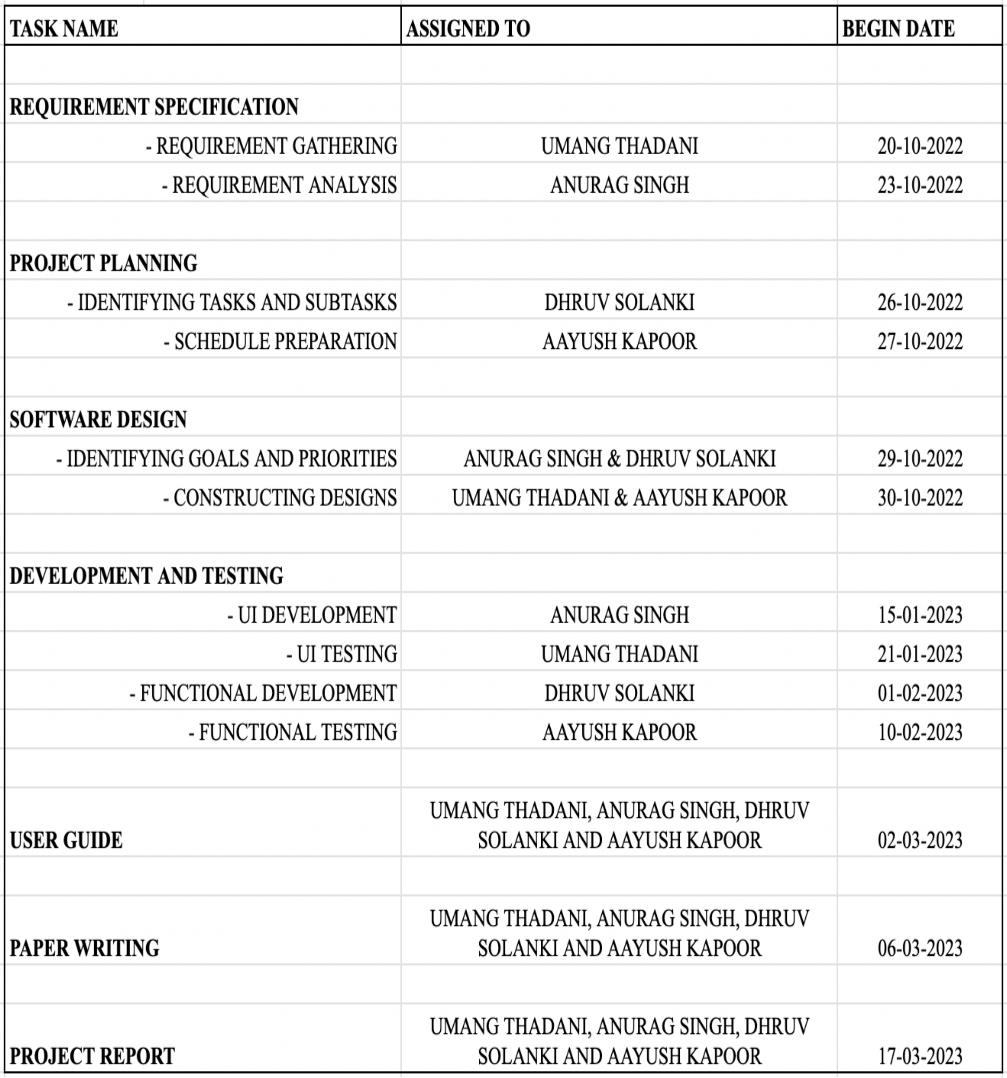
1. Software Development must be completed.
2. pytest: Testing framework

Risks and Contingencies

We must ensure that all test cases are considered.

3.3.2 Assignments

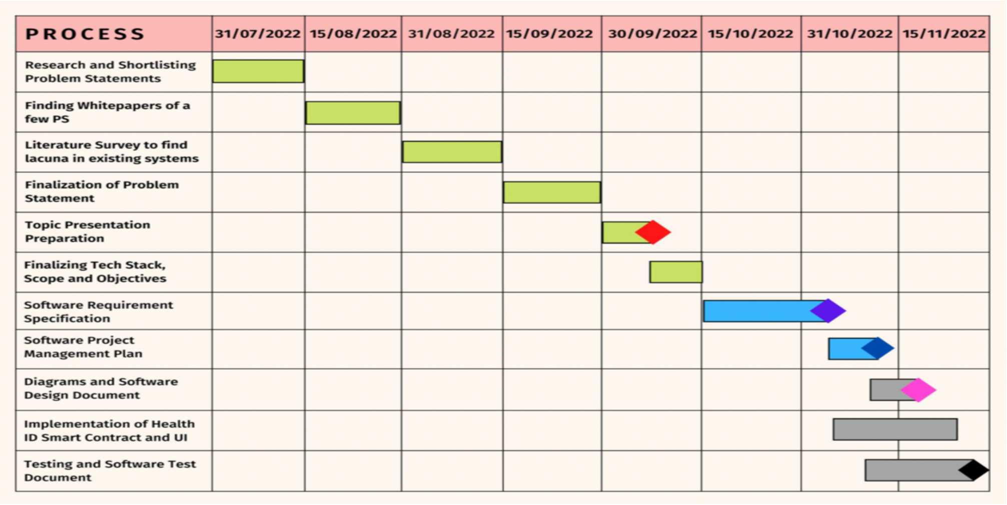
Figure 3.4: Assignments



TASK NAME	ASSIGNED TO	BEGIN DATE
REQUIREMENT SPECIFICATION		
- REQUIREMENT GATHERING	UMANG THADANI	20-10-2022
- REQUIREMENT ANALYSIS	ANURAG SINGH	23-10-2022
PROJECT PLANNING		
- IDENTIFYING TASKS AND SUBTASKS	DHRUV SOLANKI	26-10-2022
- SCHEDULE PREPARATION	AAYUSH KAPOOR	27-10-2022
SOFTWARE DESIGN		
- IDENTIFYING GOALS AND PRIORITIES	ANURAG SINGH & DHRUV SOLANKI	29-10-2022
- CONSTRUCTING DESIGNS	UMANG THADANI & AAYUSH KAPOOR	30-10-2022
DEVELOPMENT AND TESTING		
- UI DEVELOPMENT	ANURAG SINGH	15-01-2023
- UI TESTING	UMANG THADANI	21-01-2023
- FUNCTIONAL DEVELOPMENT	DHRUV SOLANKI	01-02-2023
- FUNCTIONAL TESTING	AAYUSH KAPOOR	10-02-2023
USER GUIDE	UMANG THADANI, ANURAG SINGH, DHRUV SOLANKI AND AAYUSH KAPOOR	02-03-2023
PAPER WRITING	UMANG THADANI, ANURAG SINGH, DHRUV SOLANKI AND AAYUSH KAPOOR	06-03-2023
PROJECT REPORT	UMANG THADANI, ANURAG SINGH, DHRUV SOLANKI AND AAYUSH KAPOOR	17-03-2023

3.3.3 Timetable

Figure 3.5: Project Schedule



Chapter 4

Software Requirements Specification

4.1 Intended Audience and Requirement Specification

This SRS is for developers, project managers, users, and testers. Further, the discussion will provide all the external, functional, and non-functional requirements of ”HealthKard - A Digital Decentralized Health Identity”.

4.2 External Interface Requirements

4.2.1 Hardware Interfaces

The system will be lightweight and be used on any hardware with at least 1GB of RAM. No specific hardware components will be used.

4.2.2 Software Interfaces

The software will incorporate best-in-class frameworks libraries for Front-end and Back-end and software requirements are stated with respect to support for these frameworks. Following are the software interfaces that will be used in HealthKard

- MetaMask v10.20.0 for authenticating users into our application
- Ethereum blockchain to store user data and health records in a secure, tamper-proof manner.

- Multimedia documents and other relevant information will be stored on Interplanetary File System (IPFS)

4.2.3 Communications Interfaces

Communication interfaces are those that belong to communication functions. We plan to make use of the following communication interfaces:

- The software will be developed and tested for Chrome 105+, Edge 105+, Safari 15.6+, and Firefox 105+
- Web3Modal will be used to call MetaMask and authenticate users
- We will use IPFS-HTTP Client to communicate with Interplanetary File System through our web application
- HTML forms will be used to gather user data, which will be sent to the Back-end through Axios and Rest APIs and related frameworks.
- Email will be used to answer any user queries, that will make use of SMTP protocol

4.3 Functional Requirements

4.3.1 MetaMask Authentication for authenticating the user

MetaMask is an extension for accessing Ethereum-enabled distributed applications, or ”DApp” in your browser. It also lets the user create and manage their own identities, so when a DApp wants to perform a transaction and write to the blockchain, the user gets a secure interface to review the transaction, before approving or rejecting it. It also helps warn you when you navigate to sites known to have engaged in phishing or with names that are suspiciously similar to popular phishing targets. We will allow users to authenticate to our application using MetaMask.

4.3.2 Unique Health ID Generation

A unique health identity will be generated via Smart Contracts, in the form of an NFT that will store the user’s personal as well as health-related information, which is constant throughout to maintain the non-fungible nature of NFTs. This unique ID will help us map users' health records. Also, this will enable accessing personal health records, based on international standards, easily accessible to individuals and healthcare professionals and services providers, based on individual's informed consent.

4.3.3 Health and Consultation History

The user's health history, previous consultation reports, doctor's remarks, and diagnoses will be saved and secured on the blockchain which makes it immutable. This will be mapped uniquely to the User's Health ID and can only be accessed through the individual's informed consent.

4.4 Nonfunctional Requirements

4.4.1 Usability

Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease of use during the design process.
Usability is defined by 5 quality components:

- Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- Efficiency: Once users have learned the design, how quickly can they perform tasks?
- Memorability: When users return to the design after a period of not using it, how easily can they re-establish proficiency?
- Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction: How pleasant is it to use the design?

4.4.2 Security

Security is a non-functional requirement assuring all data inside the system or its part will be protected against malware attacks or unauthorized access. Our application will only let the users authenticate to the system via their own MetaMask wallet while doctors and health experts can only view a patient's data after their consent.

4.4.3 Correctness

Our system will always display the correct and latest health records of the user since all of their data is stored on the blockchain which guarantees security, and immutability and acts as a single source of truth.

4.4.4 Maintainability

Maintainability defines the time required for a solution or its component to be fixed, changed to increase performance or other qualities, or adapted to a changing environment. Like reliability, it can be expressed as a probability of repair during some time. Since the backbone of our application uses the Ethereum blockchain, which has a 99.95% up-time, our systems will also be running 24/7 without any maintenance downtime. Also, we plan to maintain modular code so that the project can be taken forward or modified with changing requirements without any hassles.

4.4.5 Legal

Our health record storage will adopt the open standards by all national digital health stakeholders i.e. the NITI Aayog, IMHA, etc.

Chapter 5

Software Design Description

5.1 Introduction

5.1.1 Design Overview

HealthKard aims to develop the foundations necessary for supporting digital health infrastructure for maintaining health data in a decentralized and secure way. A few major advantages to this project will be ease of access, user consent for every sophisticated transaction, and portability across national borders.

HealthKard aims to implement the following modules:

- Creation of a unique Health ID using Aadhaar Number.
- Storage of Electronic Health Records (EHRs) mapped to Health Identity in the blockchain.
- Integration of different sectors in the medical industry.
- Encourage better administration of the health sector by utilizing health data analytics.

5.2 System Architectural Design

5.2.1 Client-Server Architecture

We choose client-server architecture for our system. There are two types of clients: Normal Users and Health Experts, each having a different interface. Both of these types of users interact with the same server with a common database and blockchain. Therefore, we separate concerns for application program and data management in our system.

5.2.2 Model-View-Controller

This is our second choice of architectural design. We chose Client-Server over MVC due to the fact that there is minimal dynamic content and need for dependency mechanism since the entire data needs to be changed, if any changes are applicable, only when the page reloads.

5.2.3 System Interface Description

Ethtreum Blockchain

We use the Ethereum Blockchain to save users' health records linked to the respective users' MetaMask account. The users' MetaMask Account is further linked to their Aadhar Card number.

File System

We use a modular file system where the Front End and Back End of the application are separated into different modules to improve maintainability.

Hardware Interfaces

1. Processor: x86 or x64
2. RAM: 512 MB (minimum), 1 GB (recommended)
3. Hard disk: up to 512 MB of available space may be required.

Software Interfaces

1. Operating System: Any OS that can support a gunicorn server.
2. Front End Stack: ReactJS, Recoil, MUI, TailwindCSS.
3. Back End Stack: Django REST Framework, PostgreSQL.
4. Browser Requirements: Any modern web browser having MetaMask extension.

Communication Protocols

1. The client side and the server endpoint will communicate using standard HTTP (Hyper Text Transfer Protocol) which is a generic stateless protocol.
2. The email system will use the SMTP protocol.

5.3 Detailed Description of Components

5.3.1 Authentication

1. Allow new users to login/register.
2. Verify their identity using the Aadhar APIs.
3. Change Password and Logout
4. Create profile for Health Expert.

5.3.2 Health Card creation

1. Link users' MetaMask to their Aadhar Card.
2. Save users' Health Records over the Blockchain.
3. Allow health experts to access users' data only after their consent.

5.3.3 Epidemic Analysis

1. Collecting anonymous health symptoms.
2. Analyzing anonymous health data and predicting the spread of a disease in a region.

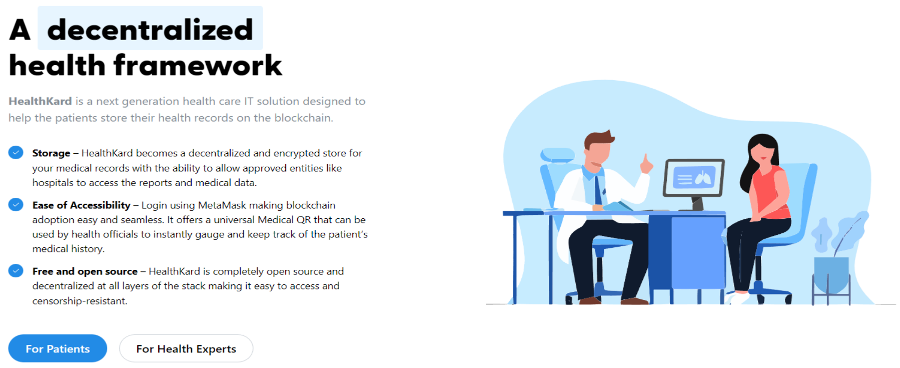
5.4 User Interface Design

5.4.1 Landing Page

A landing page is any web page that a customer can land on, but in marketing, it's usually a standalone page that serves a single and focused purpose, separate from your homepage or any other page.

Screen Images

Figure 5.1: Landing Page



A decentralized health framework

HealthKard is a next generation health care IT solution designed to help the patients store their health records on the blockchain.

- ✓ **Storage** – HealthKard becomes a decentralized and encrypted store for your medical records with the ability to allow approved entities like hospitals to access the reports and medical data.
- ✓ **Ease of Accessibility** – Login using MetaMask making blockchain adoption easy and seamless. It offers a universal Medical QR that can be used by health officials to instantly gauge and keep track of the patient's medical history.
- ✓ **Free and open source** – HealthKard is completely open source and decentralized at all layers of the stack making it easy to access and censorship-resistant.

Objects and Actions

Header It comprises of the navigation bar with hyperlink to the following pages:

1. About

2. Logo

3. Patient Login

4. Expert Login

Hero Image: It is just a simple image used to beautify the Design

Features: It showcases the features of our web application

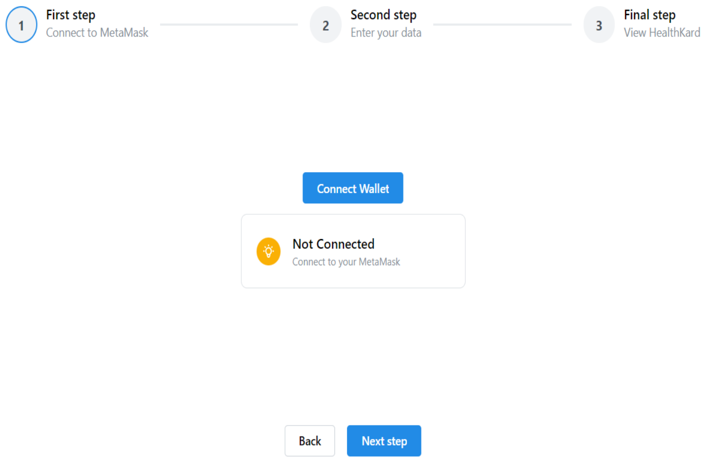
About: It is a short description of our web application

5.4.2 Patient Login

The page for the user to login using his MetaMask and create his health card.

Screen Images

Figure 5.2: Patient Login Page



Objects and Actions

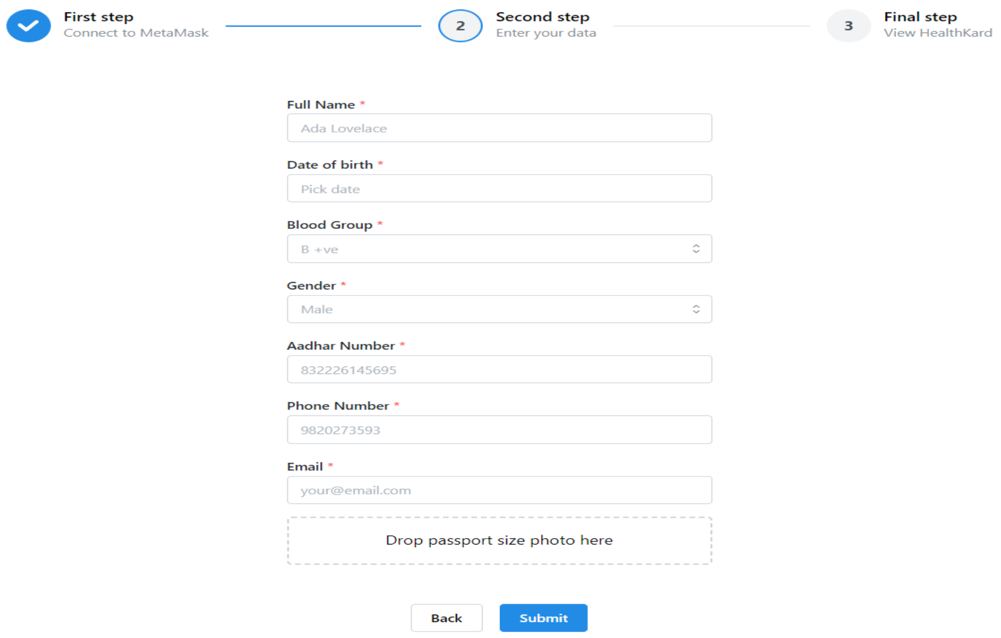
Connect Wallet It comprises of the current connection status and asks the user to connect if not connected.

5.5 Enter Patient Details

This page asks the user to enter his details to be submitted on the card.

5.5.1 Screen Images

Figure 5.3: Patient Form



Full Name *
Ada Lovelace

Date of birth *
Pick date

Blood Group *
B +ve

Gender *
Male

Aadhar Number *
832226145695

Phone Number *
9820273593

Email *
your@email.com

5.5.2 Objects and Actions

Enter Patient Data

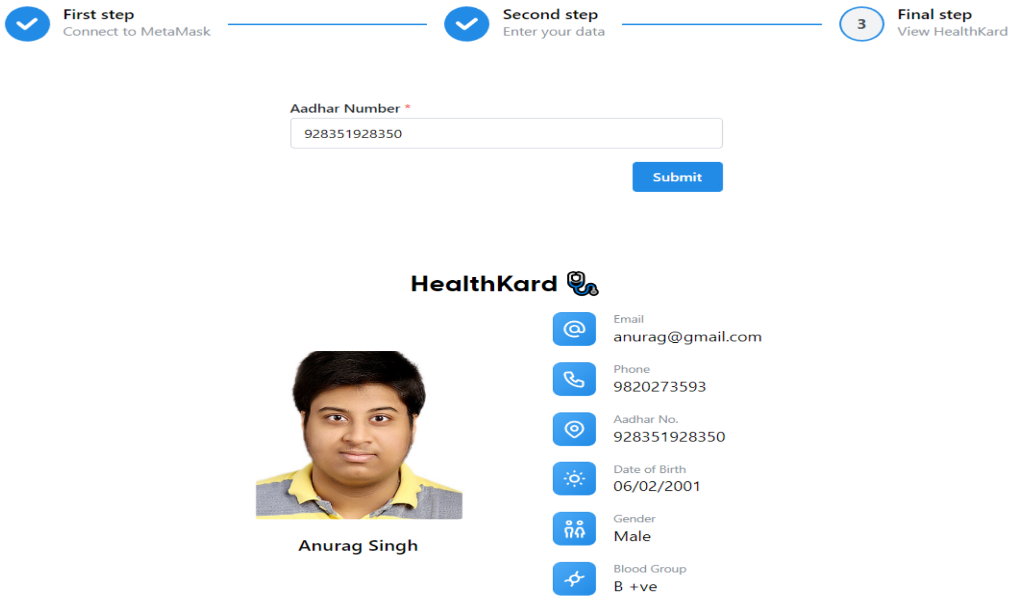
Enter the patient data and photograph with a submit button to upload to IPFS.

5.6 View Patient Healthcard

This page asks the user's Aadhar Card and shows the user's Health card.

5.6.1 Screen Images

Figure 5.4: Patient Card



Aadhar Number *
928351928350

Anurag Singh

Email
anurag@gmail.com

Phone
9820273593

Aadhar No.
928351928350

Date of Birth
06/02/2001

Gender
Male

Blood Group
B +ve

5.6.2 Objects and Actions

Enter Patient Data

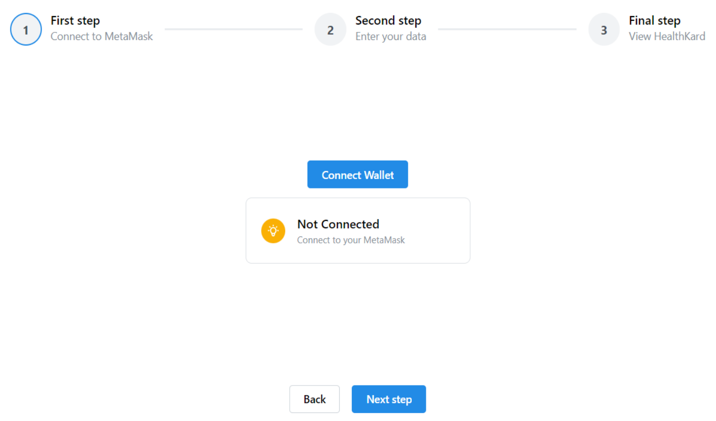
View the patient's health card by fetching details from the blockchain.

5.7 Health Expert Login

The page for the user to login using his MetaMask and create his health card.

5.7.1 Screen Images

Expert Login Page



Not Connected
Connect to your MetaMask

5.7.2 Objects and Actions

Connect Wallet

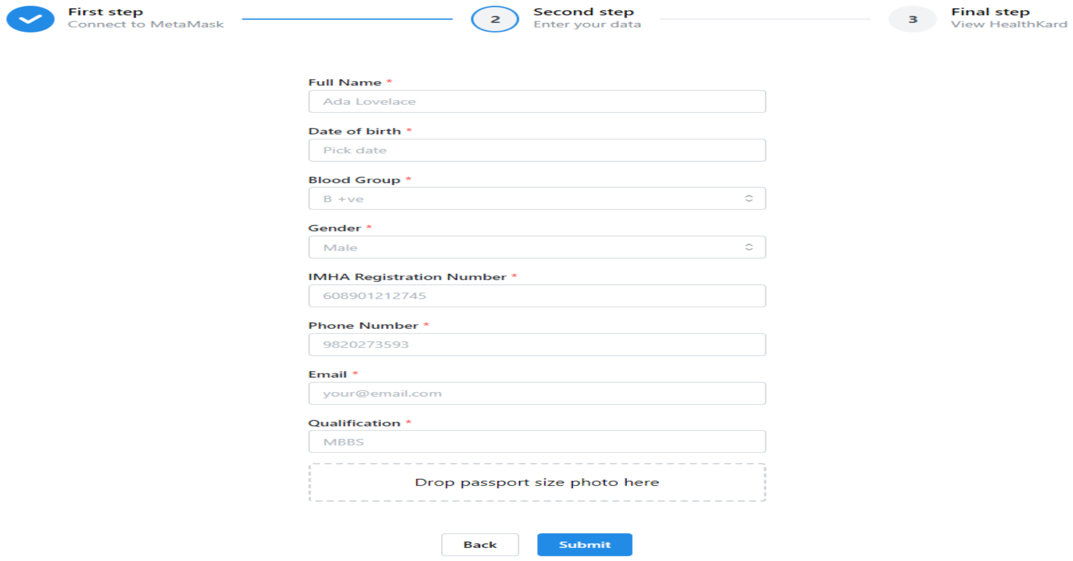
It comprises of the current connection status and asks the user to connect if not connected.

5.8 Enter Health Expert Details

This page asks the user to enter his details to be submitted on the card.

5.8.1 Screen Images

Figure 5.5: Expert Form



First step
Connect to MetaMask

2 Second step
Enter your data

3 Final step
View HealthKard

Full Name *
Ada Lovelace

Date of birth *
Pick date

Blood Group *
B +ve

Gender *
Male

IMHA Registration Number *
608901212745

Phone Number *
9820273593

Email *
your@email.com

Qualification *
MBBS

Drop passport size photo here

Back Submit

5.8.2 Objects and Actions

Enter Patient Data

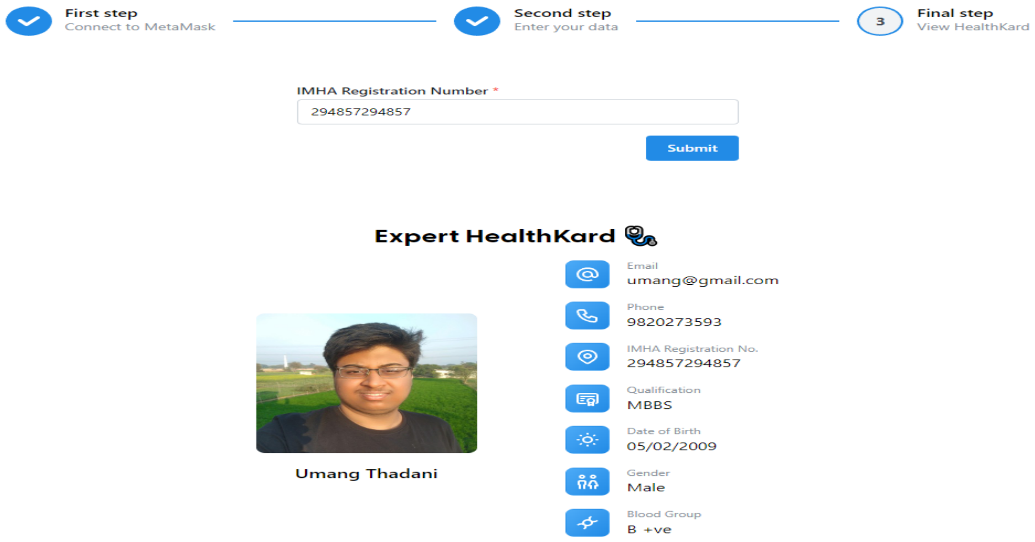
Enter the patient data and photograph with a submit button to upload to IPFS.

5.9 View Health Expert Healthcard

This page asks the user's IMHA Registration Number and shows the user's Health card.

5.9.1 Screen Images

Figure 5.6: Expert Card



IMHA Registration Number *
294857294857

Expert HealthKard

Umang Thadani

Email
umang@gmail.com

Phone
9820273593

IMHA Registration No.
294857294857

Qualification
MBBS

Date of Birth
05/02/2009

Gender
Male

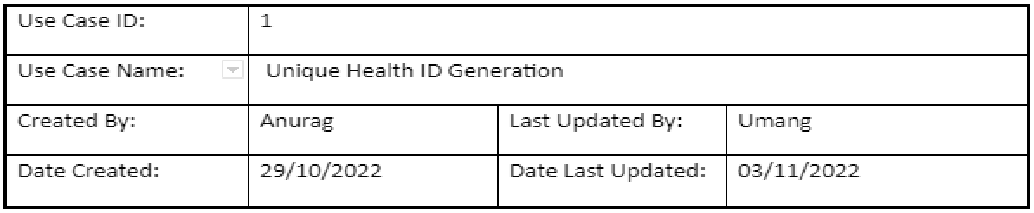
Blood Group
B +ve

5.9.2 Objects and Actions

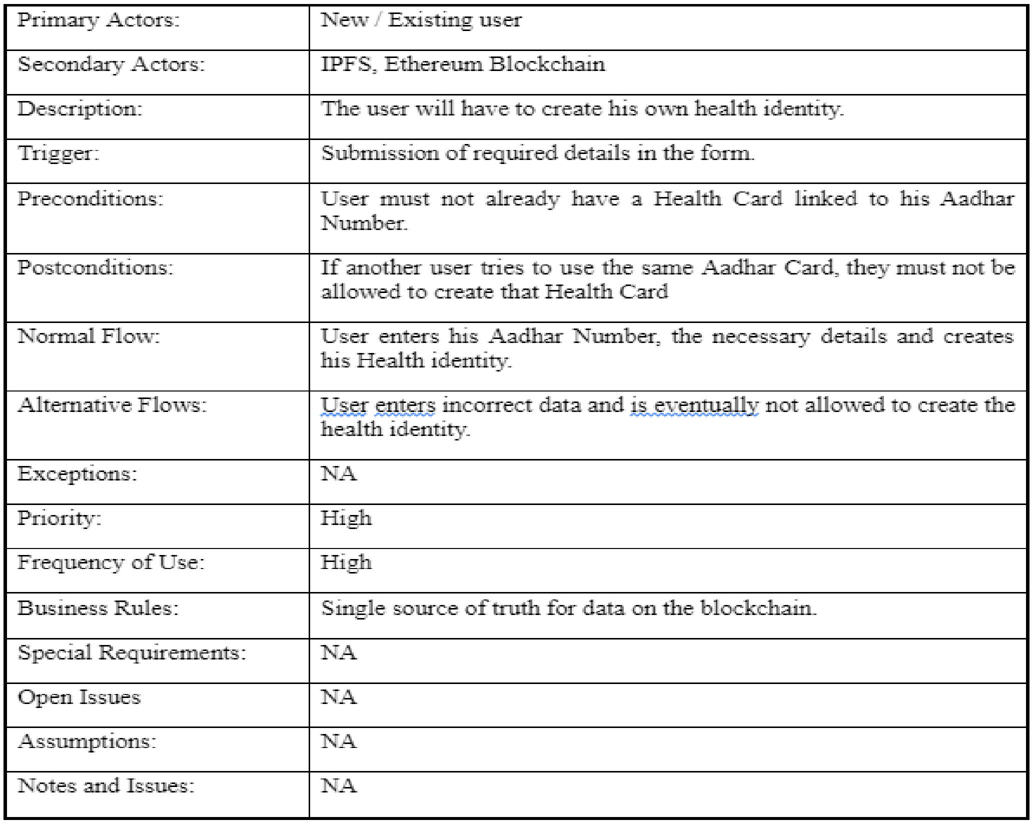
Enter Patient Data

Enter the patient data and photograph with a submit button to upload to IPFS.

5.10 System Architecture



Use Case ID:	1		
Use Case Name:	Unique Health ID Generation		
Created By:	Anurag	Last Updated By:	Umang
Date Created:	29/10/2022	Date Last Updated:	03/11/2022



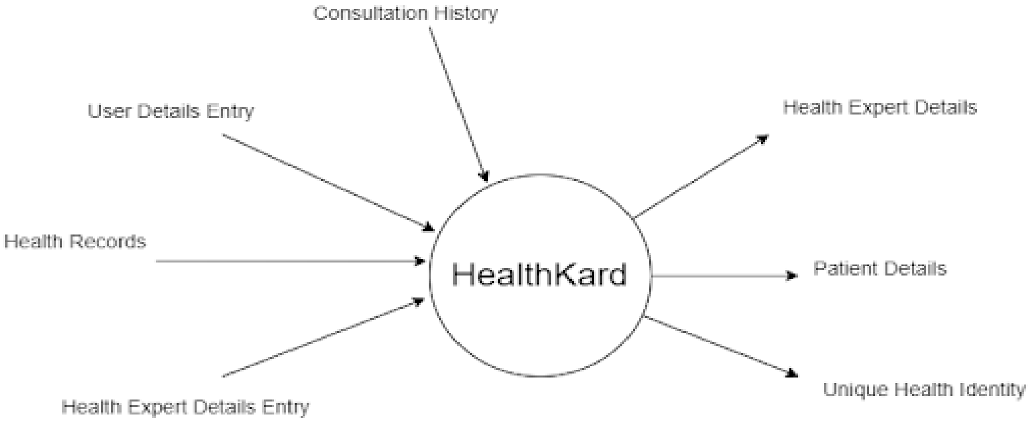
Primary Actors:	New / Existing user
Secondary Actors:	IPFS, Ethereum Blockchain
Description:	The user will have to create his own health identity.
Trigger:	Submission of required details in the form.
Preconditions:	User must not already have a Health Card linked to his Aadhar Number.
Postconditions:	If another user tries to use the same Aadhar Card, they must not be allowed to create that Health Card
Normal Flow:	User enters his Aadhar Number, the necessary details and creates his Health identity.
Alternative Flows:	User enters incorrect data and is eventually not allowed to create the health identity.
Exceptions:	NA
Priority:	High
Frequency of Use:	High
Business Rules:	Single source of truth for data on the blockchain.
Special Requirements:	NA
Open Issues	NA
Assumptions:	NA
Notes and Issues:	NA

5.11 Data Flow Specifications

DFD is created from the SRS document provided.

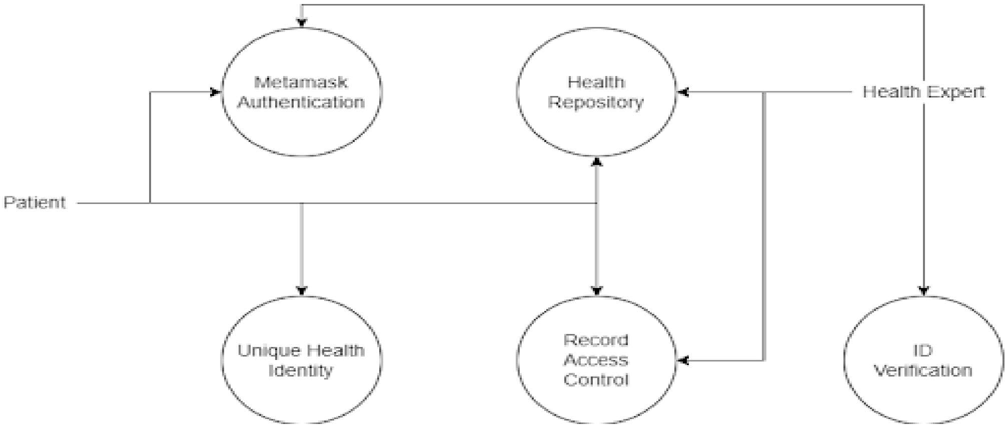
5.11.1 Level 0 DFD with description

Figure 5.7: Level 0 DFD



5.11.2 Level 1 DFD with description

Figure 5.8: Level 1 DFD



Chapter 6

Software Test Document

6.1 Test Approach

Test will be conducted to test the efficiency of the software. The reason for this test is to check for any errors and limitations. A list of various planned tests is as follows:

6.1.1 Unit Testing

Unit testing, is a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with the functional correctness of the standalone modules. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

6.1.2 White Box Testing

White Box Testing is a software testing technique in which the internal structure, design, and coding of software are tested to verify the flow of input-output and to improve design, usability, and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing. White box testing techniques analyze the internal structures, the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

6.1.3 Black Box Testing

Black box testing is a method of software testing that tests the functionality of an application as opposed to its internal structures of working. Specific knowledge of the application's code/internal structure and programming knowledge, in general, is not required. The tester is only aware of what the software is supposed to do, but not how i.e. When he/she enters a certain input, he/she gets a certain output; without being aware of how the output was produced in the first place. Test cases are built around specifications and requirements i.e. what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and design to derive test cases.

6.2 Test Plan

The objectives supported by the Test Plan are:

1. To test the project such that it meets the business and user requirements.
2. The Test Plan should find defects that may get created by the programmer while developing the software.
3. To make sure that the end result is correct.
4. Testing can be helpful in gaining confidence in and providing information about the level of quality.

The scope of the Test Plan are: The major functions that would be tested are:

1. Metamask Authentication
2. Unique Health ID Generation
3. Health and Consultation History

The following points are to be considered while testing:

1. Perform unit testing on individual modules

2. Integrate the modules to perform the integration testing after the successful completion of unit testing.

3. Then, the system as a whole will be tested and we will perform the system testing.

4. Different scenarios that can occur once the project will be deployed have been considered for ensuring effective testing of the software.

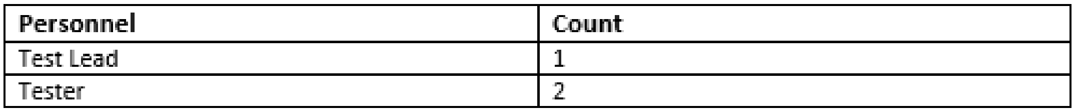
6.2.1 Features to be tested

Features	Scenarios
Metamask Authentication	• Login when the user mapped with the same address exists. • Create a new user in case a user with the same address does not exist. Verification through Aadhar has to be done in such a case.
Health Expert Verification	• If the user wants to register as a medical practitioner, he/she must be verified • Only one medical practitioner per ID must exist
Health ID Generation	• NFT must be generated when the user creates his/her profile • The details are not editable and NFT must be tamper-proof
Health Card Dashboard	• Show details relevant to only that particular user • Authorization is taken care of in this step
Patient Search	• Medical Experts should be able to search for patients using their Health ID. • Functional QR Scan and Search should be implemented.

Features	Scenarios
Patient Request	• Medical Experts should be able to request patients for access to patient's health records and the same should be reflected on the Patient's account • Patient's health records should not be visible to healthcare professionals without consent.
Manage Approval	• Patients should be able to grant/revoke access to medical practitioners to access health records. • The same should be reflected on the Practitioner's Dashboard.
Adding Medical Records	• Users should be able to add medical health records linked to their Unique Health ID. • Medical experts with access should also be able to add to patients' health records.
Editing Medical Records	• Users should be able to edit medical health records linked to their Unique Health ID. • Medical experts with access should also be able to edit patients' health records.

6.2.2 Testing Tools and Environment

Testing Staff



Personnel	Count
Test Lead	1
Tester	2

Testing Environment

Operating System

1. MAC Operating system
2. Microsoft Windows (Windows 8, 10, 11)
3. Linux Ubuntu

Devices

1. Laptop/Desktop
2. Mobile/Tablet

Browser requirement

1. Google Chrome
2. Microsoft Edge
3. Mozilla Firefox
4. Safari

Development

1. Network: Ethereum
2. Authentication: Metamask
3. File System: IPFS
4. Editor: Visual Studio Code
5. Documentation: Overleaf

6.3 Test Cases

Test Case ID	Module Name	Input	Expected Output	Actual Output	Status (Pass/-Fail)
1	Metamask Authentication	User Wallet Address	• Ensure that Metamask is connected without any issues	Metamask is successfully connected	Pass
2	Health Expert Verification	Pending	• If the user wants to register as a medical practitioner, he/she must be verified • Only one medical practitioner per ID must exist	Pending	Pending

Test Case ID	Module Name	Input	Expected Output	Actual Output	Status (Pass/-Fail)
3	Health ID Generation	• Name • Date of Birth • Blood Group • Gender • Aadhar Number • Phone Number • E-Mail • Photo	• NFT must be generated when the user creates his/her profile • The details are not editable and NFT must be tamper-proof	NFT is generated and cannot be tampered	PASS
4	Health Card Dashboard	Pending	• Show details relevant to only that particular user • Authorization is taken care of in this step	Pending	Pending

Test Case ID	Module Name	Input	Expected Output	Actual Output	Status (Pass/-Fail)
5	Patient Search	Pending	• Medical Experts should be able to search for patients using their Health ID. • Functional QR Scan and Search should be implemented.	Pending	Pending
6	Patient Request	Pending	• Medical Experts should be able to request patients for access to patient's health records and the same should be reflected on the Patient's account • Patient's health records should not be visible to healthcare professionals without consent.	Pending	Pending

Test Case ID	Module Name	Input	Expected Output	Actual Output	Status (Pass/-Fail)
7	Manage Approval	Pending	• Patients should be able to grant/revoke access to medical practitioners to access health records. • The same should be reflected on the Practitioner's Dashboard.	Pending	Pending
8	Adding Medical Records	Pending	• Users should be able to add medical health records linked to their Unique Health ID. • Medical experts with access should also be able to add to patients' health records.	Pending	Pending

Test Case ID	Module Name	Input	Expected Output	Actual Output	Status (Pass/-Fail)
9	Editing Medical Records	Pending	• Users should be able to edit medical health records linked to their Unique Health ID. • Medical experts with access should also be able to edit patients' health records.	Pending	Pending

Chapter 7

Conclusion

7.1 Conclusion & Future scope

The outcomes of the experiments suggest that our proposed strategy is capable of achieving good results. Interplanetary File System does help in saving important user data securely and makes it immutable for other users. More effort should be put into improving the verification process which have been restricted from the Aadhar APIs. Our future efforts will be focused on improving the system's performance and developing the health consultation logic for doctors as well as patients.

7.1.1 Future Scope

1. Health Consultations/Prescriptions

2. Epidemic Analysis

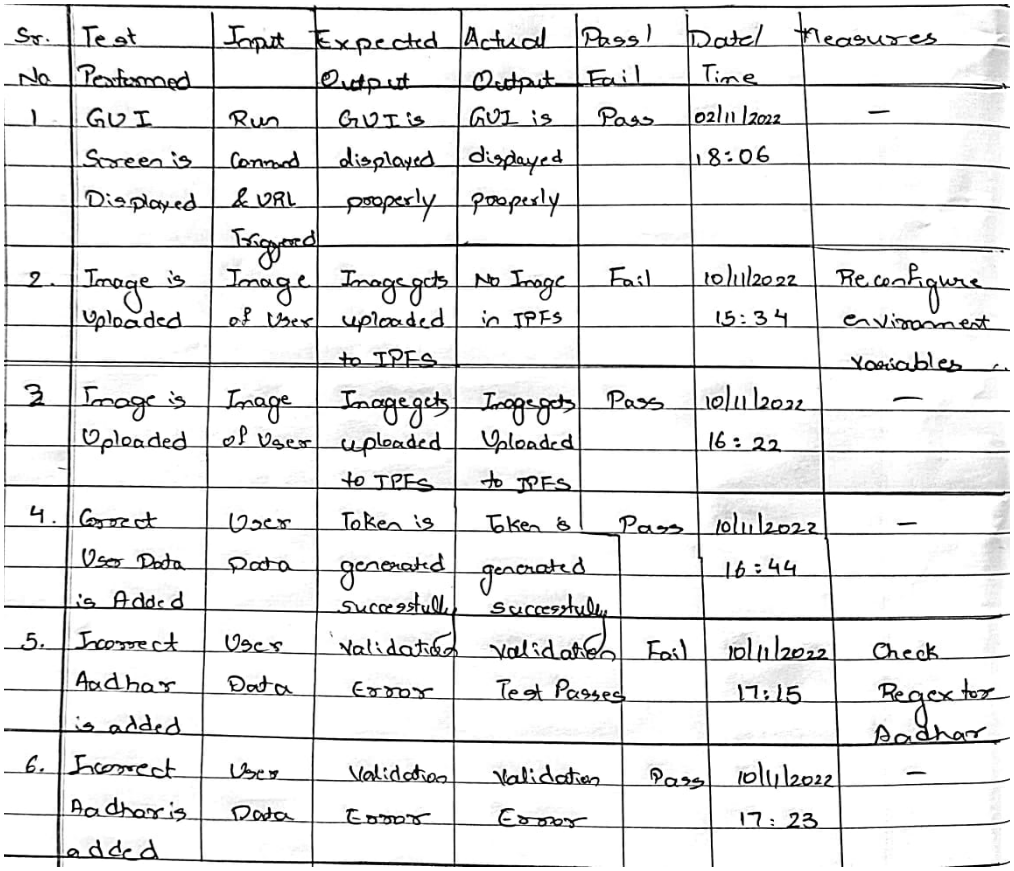
3. End-to-End Deployment

Appendix A

Appendix

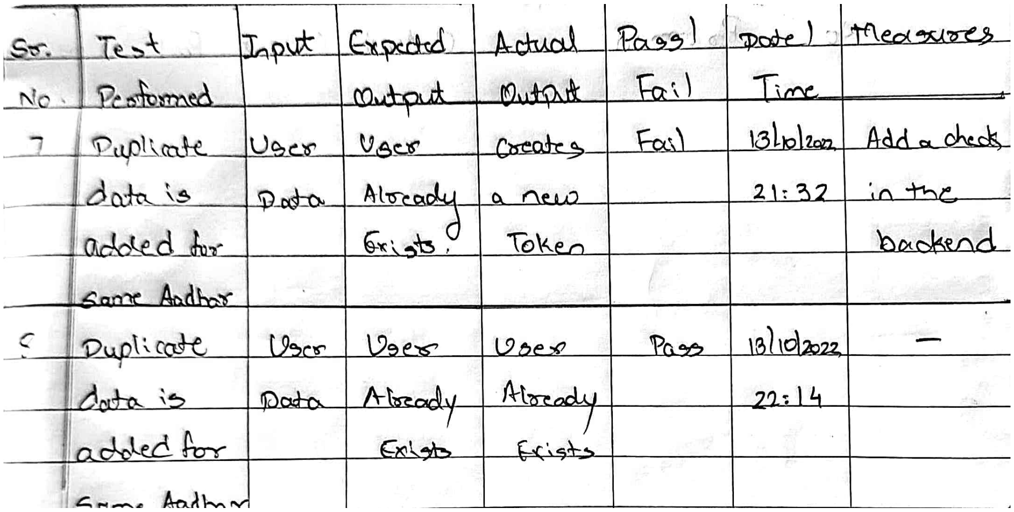
A.1 Test Logs

Figure A.1: Test Logs - 1



Sr. No	Test Performed	Input	Expected Output	Actual Output	Pass/ Fail	Date/ Time	Measures
1	GUI Screen is Displayed	Run Command & URL Triggered	GUI is displayed properly	GUI is displayed properly	Pass	02/11/2022 18:06	—
2.	Image is Uploaded	Image of User	Image gets uploaded to IPFS	No Image in IPFS	Fail	10/11/2022 15:34	Reconfigure environment variables
3	Image is Uploaded	Image of User	Image gets uploaded to IPFS	Image gets Uploaded to IPFS	Pass	10/11/2022 16:22	—
4.	Correct User Data is Added	User Data	Token is generated successfully	Token is generated successfully	Pass	10/11/2022 16:44	—
5.	Incorrect Aadhar is added	User Data	Validation Error	Validation Test Passes	Fail	10/11/2022 17:15	Check Regex for Aadhar
6.	Incorrect Aadhar is added	User Data	Validation Error	Validation Error	Pass	10/11/2022 17:23	—

Figure A.2: Test Logs - 2



Sr. No.	Test Performed	Input	Expected Output	Actual Output	Pass / Fail	Date / Time	Measures
7	Duplicate data is added for same Aadhar	User Data	User Already Exists.	Creates a new Token	Fail	13/10/2022 21:32	Add a check in the backend
8	Duplicate data is added for Same Aadhar	User Data	User Already Exists	User Already Exists	Pass	13/10/2022 22:14	—

Acknowledgement

We like to share our sincere gratitude to all those who help us in partial completion of this project. During the work, we faced many challenges due to our lack of knowledge and experience in the field of Blockchain Development but online documentations helped us to get over from all the difficulties and in final compilation of our idea to a shaped sculpture.

We would like to thank Era Johri ma'am for her governance and guidance, because of which our whole team was able to learn the minute aspects of the project work.

In the last we would like to thank the Department of Information Technology of K. J. Somaiya College of Engineering for providing us such an opportunity to learn from these experiences.

All of our team is thankful to the evaluation committee and the respective faculties. We are also thankful to our peers and our parents who have inspired us to face all the challenges and overcome all the hurdles faced throughout the project.

Thank you all.

Anurag Sanjay Singh
Dhruv Rakesh Solanki
Umang Chander Thadani
Aayush Vivek Kapoor