# SOFTWARE DESIGN DOCUMENT

**for**

# HEALTHKARD

**Prepared by :**
**Umang Thadani (1914061)**
**Anurag Singh (1914058)**
**Dhruv Solanki (1914059)**
**Aayush Kapoor (1914066)**

**Guide: Prof. Era Johri**

# Contents

# 1 Introduction

## 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.

# 2 System Architectural Design

## 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.

## 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.

## 2.3 System Interface Description

### 2.3.1 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.

### 2.3.2 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.

### 2.3.3 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.

### 2.3.4 Software Interfaces

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

### 2.3.5 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.

# 3 Detailed Description of Components

## 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.

## 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.

## 3.3 Epidemic Analysis

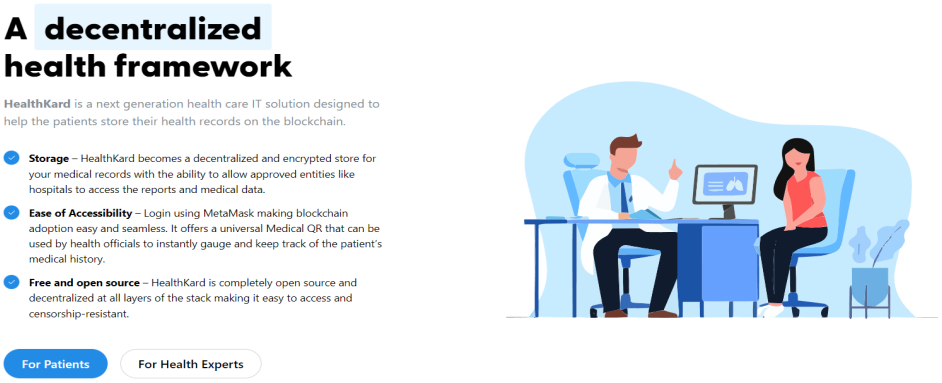
1. Collecting anonymous health symptoms.
2. Analyzing anonymous health data and predicting the spread of a disease in a region.

# 4 User Interface Design

## 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.

### 4.1.1 Screen Images




### 4.1.2 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

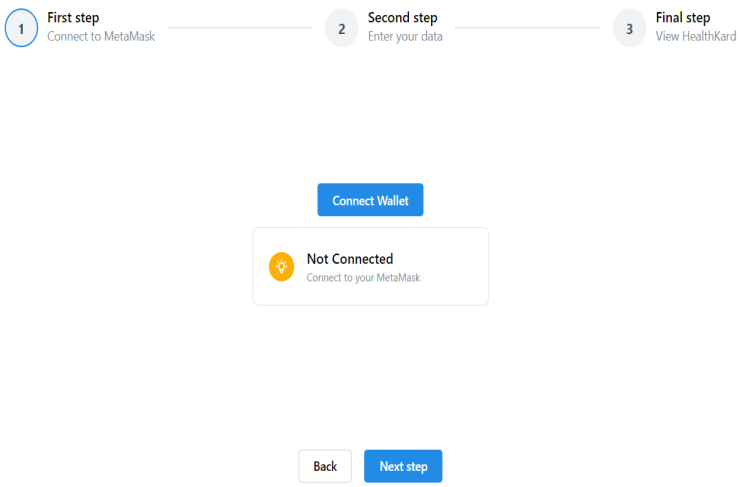
### About

It is a short description of our web application

## 4.2 Patient Login

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

### 4.2.1 Screen Images



### 4.2.2 Objects and Actions

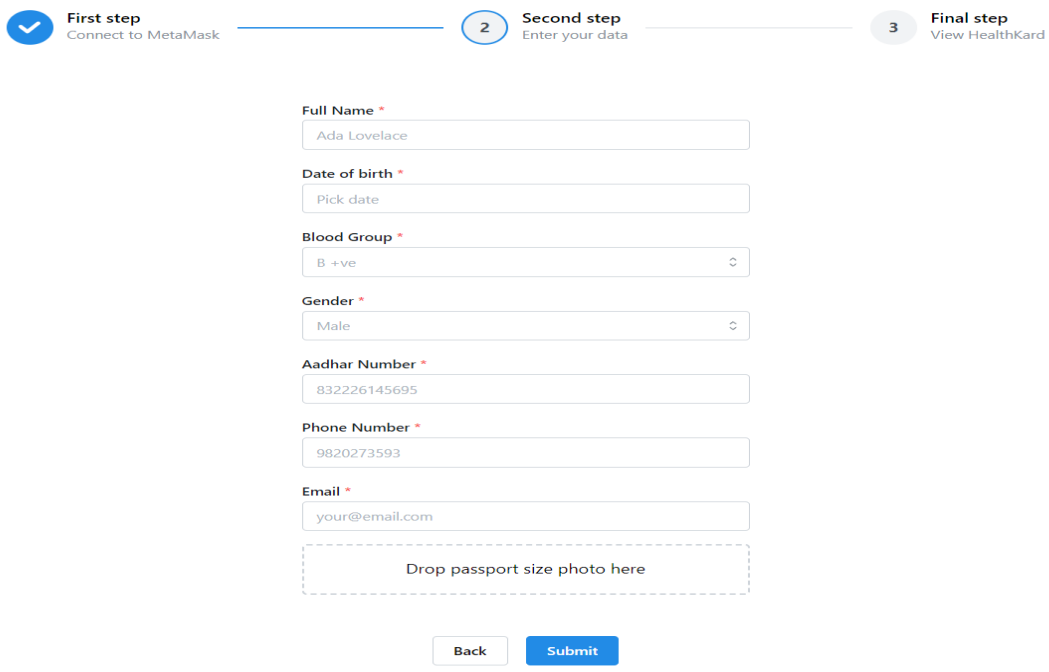
#### Connect Wallet

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

## 4.3 Enter Patient Details

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

### 4.3.1 Screen Images




### 4.3.2 Objects and Actions

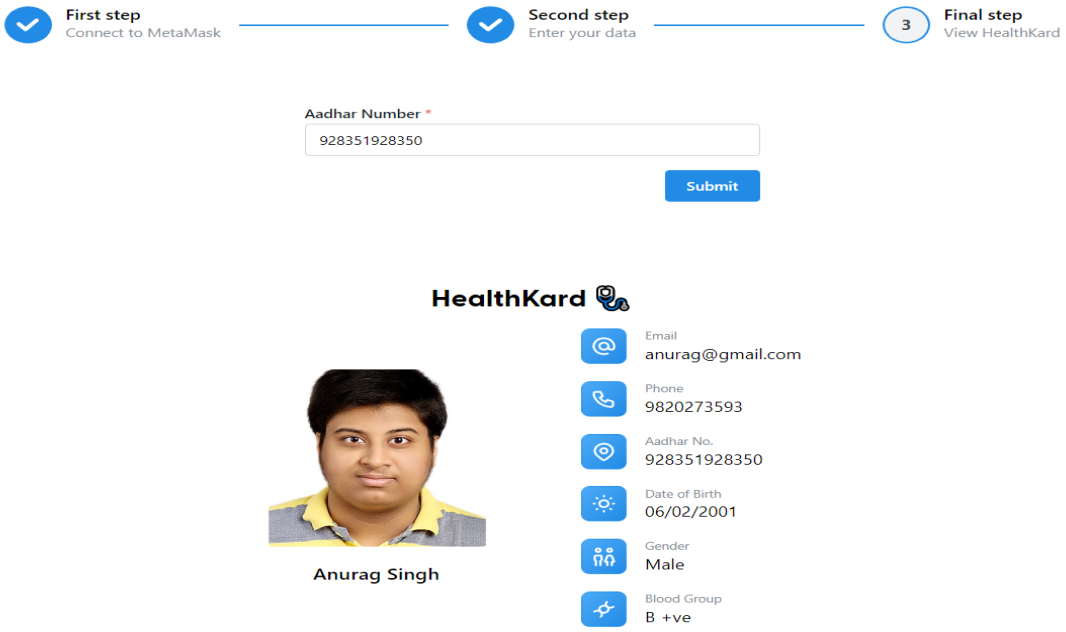
#### Enter Patient Data

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

## 4.4 View Patient Healthcard

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

### 4.4.1 Screen Images




### 4.4.2 Objects and Actions

#### Enter Patient Data

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

## 4.5 Health Expert Login

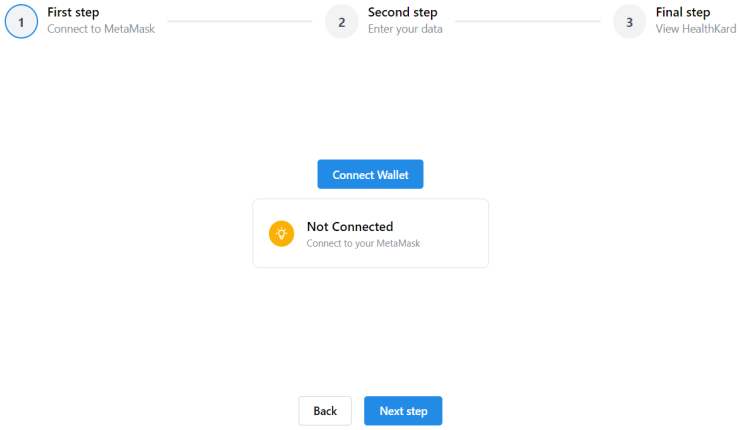
The page for the user to login using his MetaMask and create his health card.

### 4.5.1 Screen Images

### 4.5.2 Objects and Actions

#### Connect Wallet

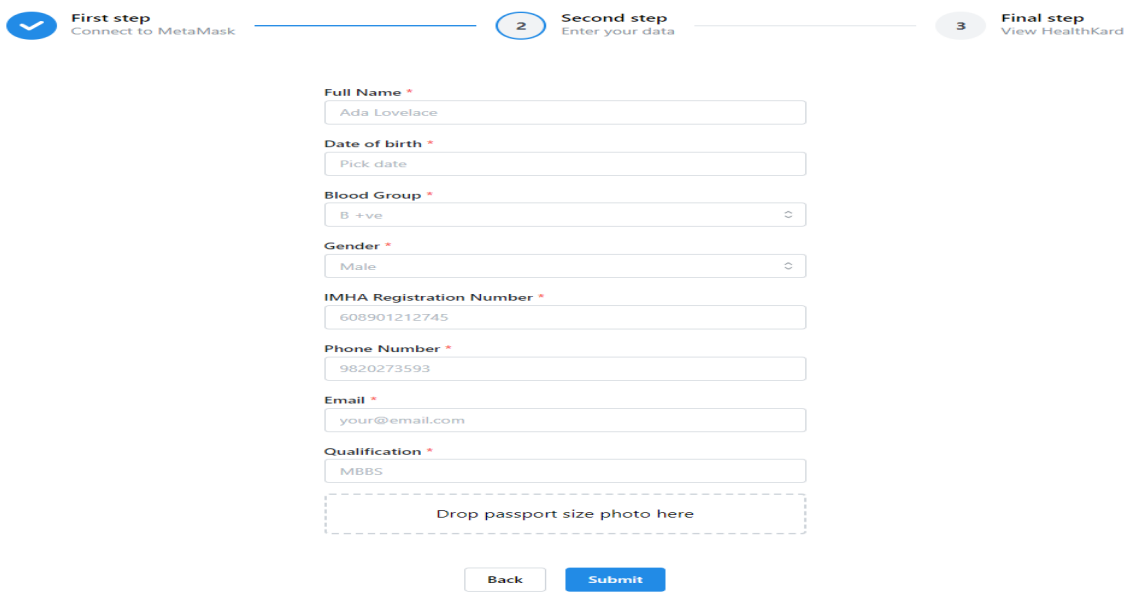
It comprises of the current connection status and asks the user to connect if not connected.




## 4.6 Enter Health Expert Details

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

### 4.6.1 Screen Images



### 4.6.2 Objects and Actions

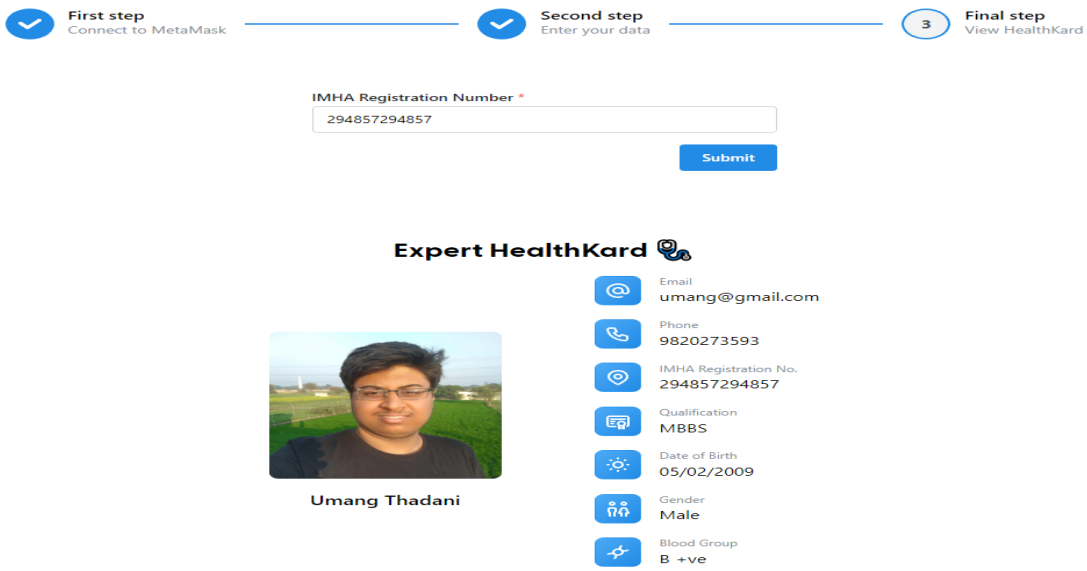
#### Enter Patient Data

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

## 4.7 View Health Expert Healthcard

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

### 4.7.1 Screen Images




### 4.7.2 Objects and Actions

#### Enter Patient Data

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

# 5 System Architecture

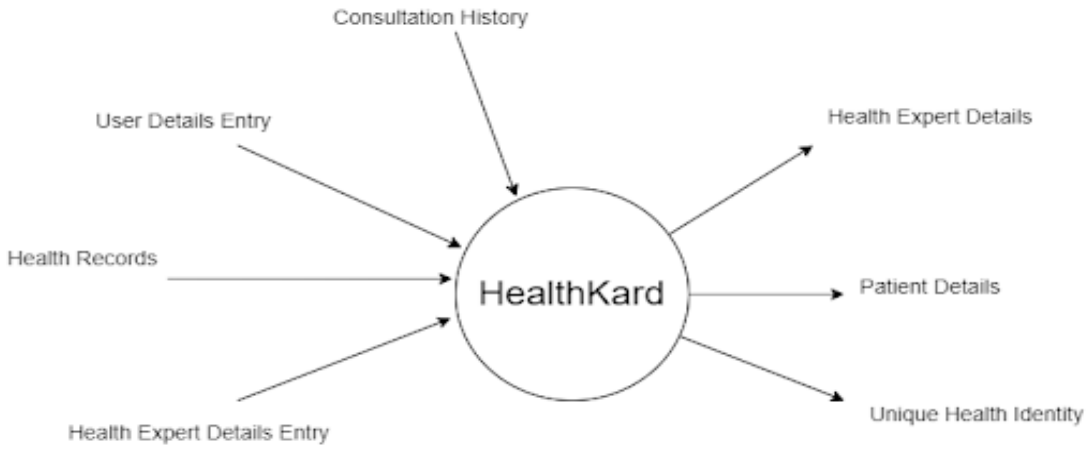
| Use Case ID: | 1 | | |
|---|---|---|---|
| Use Case Name: | Unique Health ID Generation | | |
| Created By: | Anurag | Last Updated By: | Umang |
| Date Created: | 29/10/2022 | Date Last Updated: | 03/11/2022 |

| 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 |

# 6 Data Flow Specifications

DFD is created from the SRS document provided.

## 6.1 Level 0 DFD with description




## 6.2 Level 1 DFD with description