
SOFTWARE REQUIREMENTS SPECIFICATION

for

HEALTHKARD

Group ID: 3

Umang Thadani (1914061)

Anurag Singh (1914058)

Dhruv Solanki (1914059)

Aayush Kapoor (1914066)

Guide: Prof. Era Johri

Contents

1	Introduction	3
1.1	Product Overview	3
1.2	Intended Audience and Requirement Specification	3
1.3	Problem Definition	3
2	External Interface Requirements	4
2.1	Hardware Interfaces	4
2.2	Software Interfaces	4
2.3	Communications Interfaces	4
3	Functional Requirements	5
3.1	MetaMask Authentication for authenticating the user	5
3.2	Unique Health ID Generation	5
3.3	Health and Consultation History	5
4	Nonfunctional Requirements	6
4.1	Usability	6
4.2	Security	6
4.3	Correctness	6
4.4	Maintainability	6
4.5	Legal	7

1 Introduction

1.1 Product Overview

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.

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

1.2 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".

1.3 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)

2 External Interface Requirements

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.

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)

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

3 Functional Requirements

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.

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.

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 Nonfunctional Requirements

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?

4.2 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.

4.3 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 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.

4.5 Legal

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