**Innovature Software Labs P Ltd**

**Software Design Document of**

**Kensha**

| Disclaimer | This document contains confidential information and it is intended for Innovature Software Labs Pvt Ltd use only. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited. |
| --- | --- |
| Format ID : | ISL-ISMS-49-Software Requirement Specification |

# Contents

[**Contents**](#_heading=h.1y810tw) **2**

[**Document Control**](#_heading=h.qsh70q) **4**

[**Change Log**](#_heading=h.23ckvvd) **5**

[**1.Introduction**](#_heading=h.i8e0bfjaqb0k) **5**

[1.1 Purpose](#_heading=h.j5q9q21yqhsy) 5

[1.2 Scope](#_heading=h.2grqrue) 5

[1.3 Definitions, Acronyms & Abbreviations](#_heading=h.a4yqdr9k74hi) 5

[**2.Document Overview**](#_heading=h.b9t6t4axniit) **6**

[**3.System Overview**](#_heading=h.ezi0gtg8vc8d) **6**

[**4. Design Overview**](#_heading=h.pbkof3vqoy5h) **6**

[4.1 Application Platform](#_heading=h.1pxezwc) 6

[4.2 Network Aspects & Data Storage](#_heading=h.y9yg0c4qxzi3) 7

[4.2.1 Distributed Ledger Technology (DLT)](#_heading=h.uhjeeuh79o5g) 7

[4.2.2 Blockchain](#_heading=h.edinlxl4cv80) 7

[4.2.3 Ethereum Blockchain](#_heading=h.7ybhh7a7kghs) 7

[4.2.4 Smart Contract](#_heading=h.5qjsuqkgvu9x) 7

[4.2.5 NFTs](#_heading=h.7mgca04iph3r) 8

[**5. Detailed Design**](#_heading=h.qpaik0obd0zs) **8**

[5.1 Patent Holder/Applicant](#_heading=h.d2nh3rywgxfb) 8

[5.1.1 Search for patents](#_heading=h.5xo0b266wnlf) 8

[5.1.2 Submit Patent Application](#_heading=h.ht2agurkhhvl) 9

[5.1.3 View Patent Details](#_heading=h.au9w125bp0r9) 9

[5.1.4 Communicate with Authority](#_heading=h.ixdif7hdxzp6) 9

[5.1.4 Pay Patent Fees](#_heading=h.sc0ycagu5jbn) 9

[5.1.6 Buy Patents](#_heading=h.2wom238rzzdu) 9

[5.1.7 Sell Patents](#_heading=h.we0xgt3y80oq) 9

[5.2 Patent Certifying Authority](#_heading=h.8p8n3e6iwhgb) 10

[5.2.1 Authority login](#_heading=h.np9rvnymdzu3) 10

[5.2.1 List All Patent Applications](#_heading=h.qd4ec2oa4dr3) 10

[5.2.2 Review Patent](#_heading=h.bjvv0bnos2p) 10

[5.2.3 Approve or Reject Patent](#_heading=h.xdm76oh8t7q8) 11

[5.3 Public](#_heading=h.f5dy4m5zg5k) 11

[5.3.1 Search And View Patent Details](#_heading=h.rgi2sm56cq70) 11

[**6. Database Design**](#_heading=h.rq4l62y3mim1) **11**

[**7. User Interface Design**](#_heading=h.mmiu2r9y0bkk) **14**

[7.1 All Patents (Public User)](#_heading=h.y2phlximoat5) 14

[7.1 Registration](#_heading=h.g0c0zd6ixuz3) 14

[7.2 User Profile](#_heading=h.golvtswhyk9d) 15

[7.3 Patent Details](#_heading=h.1rf1h4jxni77) 15

[7.4 My Patents (User)](#_heading=h.nx3zi4qf058t) 16

[7.5 Apply Patent (User)](#_heading=h.20c27hq9zrx) 16

[7.6 My Bids (User)](#_heading=h.xrh53yfxbciw) 17

[7.7 Make Payment (User)](#_heading=h.9fku4xf97s4s) 17

[7.8 Approve Bid (User)](#_heading=h.wfoqluu5kbr7) 18

[7.9 All Patents (Authority)](#_heading=h.pt9smfdu27tu) 18

# Document Control

| Document Name | Software Design Document |
| --- | --- |
| Abstract | This document defines the Software Design Documentation at Innovature Software Labs Pvt Ltd. |
| ID Reference | ISL-ISMS-83-Software Design Document |
| Date | 2022-07-26 |

Authorization:

| Prepared by | Reviewed by | Authorized by |
| --- | --- | --- |
| PBC | ISMS Core Team | CISO |
| Signature: Nirmal N | Signature: | Signature: |
| 2022-07-26 |  |  |

Security Classification: Internal Use

# Change Log

| Version No | Details of Change | Change Date | Prepared By | Approved By |
| --- | --- | --- | --- | --- |
| 0.1 | Initial Version | 2022-07-26 | Jerald Jacob | Nirmal N |
| 0.2 |  | 2022-09-15 | Prannoy Mathew |  |

# 1.Introduction

## 1.1 Purpose

The purpose of the document is to present a description of the design aspects of the Kensha blockchain project. This document will be the main input for the development phase of this project.

## 1.2 Scope

This document gives a detailed description of the architecture and implementation of the Kensha blockchain project.

## 1.3 Definitions, Acronyms & Abbreviations

| Definition/ Acronyms | Details |
| --- | --- |
| S3 | Amazon Simple Storage Service |
| DLT | Distributed Ledger Technology |
| NFT | Non-Fungible Token |
| SHA-256 | Secure Hash Algorithm 256-bit |

ewD

# 2.Document Overview

Section 3 – 5 contains discussions on different design aspects of the project. Section 6 contains database design and section 7 contains screenshots of the system.

# 3.System Overview

This is a blockchain based application that helps to keep a patent document in a safe and secure format. The main concept of the project is to implement blockchain technology. This portal contains mainly three types of users.

* Patent Holder
* Patenting Authority
* Public

Patent Holders can upload the application to the portal and send it to the Patent authority for approval. The details of the patents will be stored in our MongoDB database and the patent files will be uploaded on an S3 bucket.

The patent authority can review the application and approve/reject it. Once the patent is approved, an NFT will be created on the blockchain embedded with the details of the patent including a json string containing SHA-256 hash of the files uploaded in the S3 bucket.

Public users can view all the patents approved by the authority.

This application currently supports only Metamask digital wallet. Hence, users are required to install the Metamask browser extension in order to use the application correctly.

# 4. Design Overview

## 4.1 Application Platform

| Kensha Blockchain Application | |
| --- | --- |
| Application Type | Ethereum Blockchain Application |
| Development language | React, NodeJs, Solidity |
| Database | MongoDB, Ethereum Blockchain |

## 4.2 Network Aspects & Data Storage

### **4.2.1 Distributed Ledger Technology (DLT)**

Distributed ledger technology (DLT) is a digital system for recording the transaction of assets in which the transactions and their details are recorded in multiple places at the same time. Unlike traditional databases, distributed ledgers have no central data store or administration functionality.

### **4.2.2 Blockchain**

A blockchain is a distributed database or ledger that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Large files cannot be efficiently stored on blockchains because the blockchain becomes bloated with data that has to be propagated within the blockchain network. On the other hand, since the blockchain is replicated on many nodes, a lot of storage space is required without serving an immediate purpose, especially if the node operator does not need to view every file stored on the blockchain. It furthermore leads to an increase in the price of operating blockchain nodes because more data needs to be processed, transferred, and stored.

### **4.2.3** **Ethereum Blockchain**

Ethereum is a decentralized, open-source blockchain with smart contract functionality. Ether (ETH or Ξ) is the native cryptocurrency of the platform. Ethereum's smart contracts are written in high-level programming languages such as Solidity, and then compiled down to EVM bytecode and deployed to the Ethereum blockchain. There are two types of accounts on Ethereum: user accounts (also known as externally-owned accounts) and contracts. Both types have an ETH balance, may send ETH to any account, may call any public function of a contract or create a new contract, and are identified on the blockchain and in the state by an account address.

### **4.2.4** **Smart Contract**

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately sure of the outcome, without an intermediary’s involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met. A smart contract is just a digital contract with the security coding of the blockchain. A smart contract has details and permissions written in code that require an exact sequence of events to take place to trigger the agreement of the terms mentioned in the smart contract. It can also include the time constraints that can introduce deadlines in the contract. This contract is deployed on the blockchain, making it transparent, immutable, inexpensive, and decentralized. Every smart contract has its address in the blockchain. The contract can be interacted with by using its address, presuming the contract has been deployed on the network.

### **4.2.5** **NFTs**

An NFT or non-fungible token, is a unique digital identifier that cannot be copied, substituted, or subdivided, that is recorded in a blockchain, and is used to certify authenticity and ownership. The ownership of an NFT is recorded in the blockchain and can be transferred by the owner, allowing NFTs to be sold and traded. NFTs typically contain references to digital files such as photos, videos, and audio. Because NFTs are uniquely identifiable assets, they differ from cryptocurrencies, which are fungible. NFTs can be created and traded using smart contracts deployed on the blockchain.

# 5. Detailed Design

## 5.1 Patent Holder/Applicant

Patent holders can apply for new patents and view the statuses of patents applied by them. When a new application for a patent is applied, it is reviewed by the Authority and then approved. Once the authority approves a patent application, a transaction is sent to the blockchain invoking a function on the smart contract to store a json containing the SHA-256 hash of the patent files, along with other information of the patent as a non-fungible token (NFT). Once this transaction is mined successfully, the user will be notified and an NFT will be created under the address of the smart contract. The user will then be provided with a link to see the details of the NFT on a public ethereum blockchain explorer (etherscan).

### **5.1.1** **Search for patents**

Applicants can view and search among the list of all approved patents and all patents applied by the applicant itself.

### **5.1.2** **Submit Patent Application**

Applicants can apply for patents using the apply patent button which opens a popup and lets the applicant enter the patent name and description and also upload necessary documents in PDF format. Upon successfully applying for a patent, the application is sent to the patent authority for review. Until the patent authority starts the review process of the patent, the applicant is able to edit the contents of their application.

### **5.1.3** **View Patent Details**

Applicants can click on any patent on his screen upon which he/she will be redirected to a detail page of that particular patent where more detailed information about the patent is displayed. The user can also view/download the patent documents that the patent owner has uploaded with the patent. For approved patents, a button called “View on Etherscan” is provided which will redirect the user to Etherscan, an ethereum public blockchain explorer, where they can see the NFT details of that patent.

### **5.1.4** **Communicate with Authority**

Applicants can communicate with the patent authority by adding comments under the patents they have created. These comments can be viewed only by the patent owner and patent authority. In case a patent has been rejected, the patent owner can ask for a justification in the comment section and later re-apply their patent.

### **5.1.4** **Pay Patent Fees**

Once a patent is ready to be approved, the user will receive a notification in the application after which he/she has to make a fee payment in order to complete the approval process and initiate the minting of NFT. On successful completion of fee payment, an NFT of the patent will be minted for the patent owner.

### **5.1.6** **Buy Patents**

Applicants can buy patents from other patent owners who have listed their patents for sale. If a patent is up for sale, applicants are given the option to place their bids above a base amount set by the patent owner. Once their bid is approved by the patent owner, the applicant receives a notification and is given two days to complete his/her payment. On successful payment, the patent is transferred to the applicant and he/she will be the new owner of the patent. In case the applicant fails to make payment by the due date, his/her bid will be cancelled and the patent will be once again up for bidding. The applicants can see their active bid statuses in the “My Bids” page.

### **5.1.7** **Sell Patents**

Once a patent gets approved by authority, applicants who wish to sell their patents to other users have the option to list their patents for sale for a price of their choice. On doing so, other users can place bids on patents they wish to own. The seller of the patent can then view all bid amounts and approve the highest bidder to sell their patent to. The selected bidder gets notified and can complete the payment within 2 days upon which the patent ownership will be transferred to his account. The seller will receive the bidded amount to his wallet after deducting a small amount as commission fee. In case, the bidder fails to make payment within the due date, his/her bid will be cancelled and the patent will be listed back on sale.

## 5.2 Patent Certifying Authority

### 

### **5.2.1** **Authority login**

Patent certifying authority can login to the portal using their digital wallet for accessing and approving the patent applications submitted by applicants. After the wallet address of the authority is authenticated, they will be permitted to access the remaining functionalities of the application as patent authority.

### **5.2.1** **List All Patent Applications**

The certifying authority can view a list of all patent applications from all users. He/she can also filter the list to show only approved, reviewed and rejected applications.

The table below demonstrates the fields for listing all applications

| # | Name | Type | Description |
| --- | --- | --- | --- |
| 1 | Patent Name | Label | Title for the patent |
| 2 | Patent Description | Label | Short description about the patent |
| 3 | Patent owner | Label | Name of the patent owner |
| 4 | Patent status | Label | Status of the patent |
| 5 | Patent Hash | Label | SHA-256 Hash of files |

### **5.2.2** **Review Patent**

The patent certifying authority can select a patent application on which he/she will be redirected to a detail page containing all the details and files pertaining to the selected patent. In this step, the authority downloads and reviews the patent files to check if all necessary information is present and the patent doesn’t create a conflict with other already approved patents.

### **5.2.3** **Approve or Reject Patent**

Once the authority has reviewed the application, he/she can decide to approve or reject the same. On approval, a transaction will be sent to ethereum blockchain invoking the smart contract to mint an NFT based on the selected patent application. The SHA-256 hashes of the files uploaded along with the patent will be converted to a JSON object which is then stringified and stored along with the NFT. Once this transaction is mined successfully, the patent status gets updated to “approved” and the NFT can be viewed on public ethereum blockchain explorers like etherscan. On Rejection, the application status will be updated to “rejected” and the authority will be asked to enter a comment stating the reason for rejection. The applicant can then view the reason and communicate with the authority and later re-apply the patent.

During the creation of NFT, the following details will be stored on the blockchain.

| # | Name | Type | Description |
| --- | --- | --- | --- |
| 1 | Token Id | Label | Id of the Token |
| 2 | User Id | Label | Id of the user |
| 3 | Initial Token Owner | Label | Wallet address of initial token owner |
| 4 | Current Token Owner | Label | Wallet address of current token owner |
| 5 | Token URI | Label | Hash of the patent files |

## 5.3 Public

### 

### **5.3.1** **Search And View Patent Details**

All approved patents which have been successfully converted to NFTs are displayed to public users. Users can view this list, click on any patent to view more details about it and also search through this list. A button called “View on Etherscan” is also provided which will redirect the user to Etherscan, an ethereum public blockchain explorer, where they can see the NFT details of that patent.

# 6. Database Design

**Signature Table**

{

publicAddress: String, nonce: Number, expire\_at: { type: Date, default: Date.now, expires: 60 }

}

**User Table**

{

\_id: String,

name:String,

address:String,

district:String,

pincode:String,

phoneNo:String,

email:String,

usertype:Number

}

**Patents Table**

{

\_id: String,

patentName: String,

publicAddress: String,

description: 'Sample patent file',

s3Address: Array,

blockchainTransactionNo: String,

statusLog: [

{

status: Number,

message: String,

\_id: String,

statusDate: Date

}

],

status: Number,

message: [

{

message: String

user: String

\_id: String

statusDate: Date

}

],

createdAt: Date,

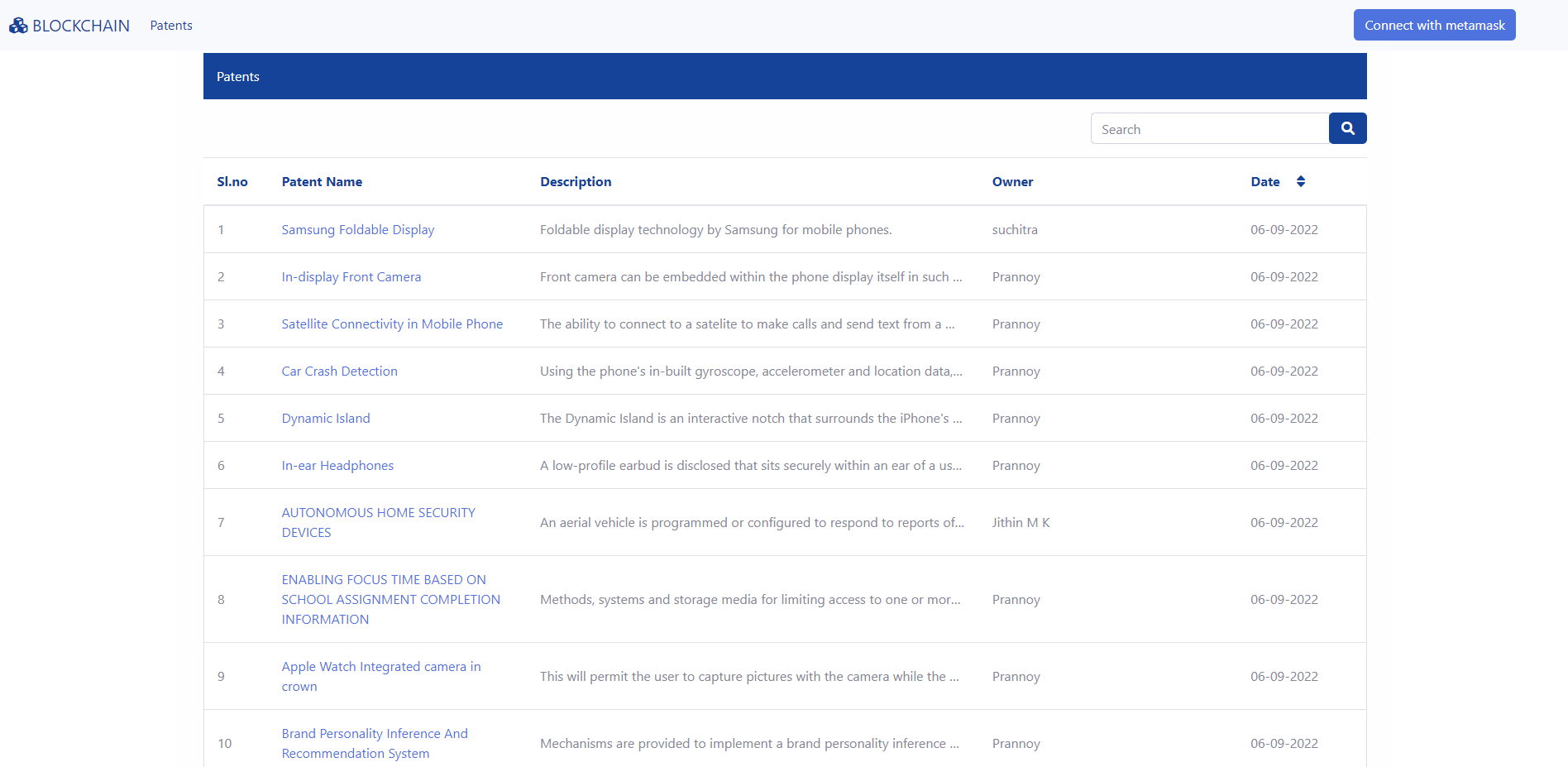
updatedAt: Date,

\_\_v: Number

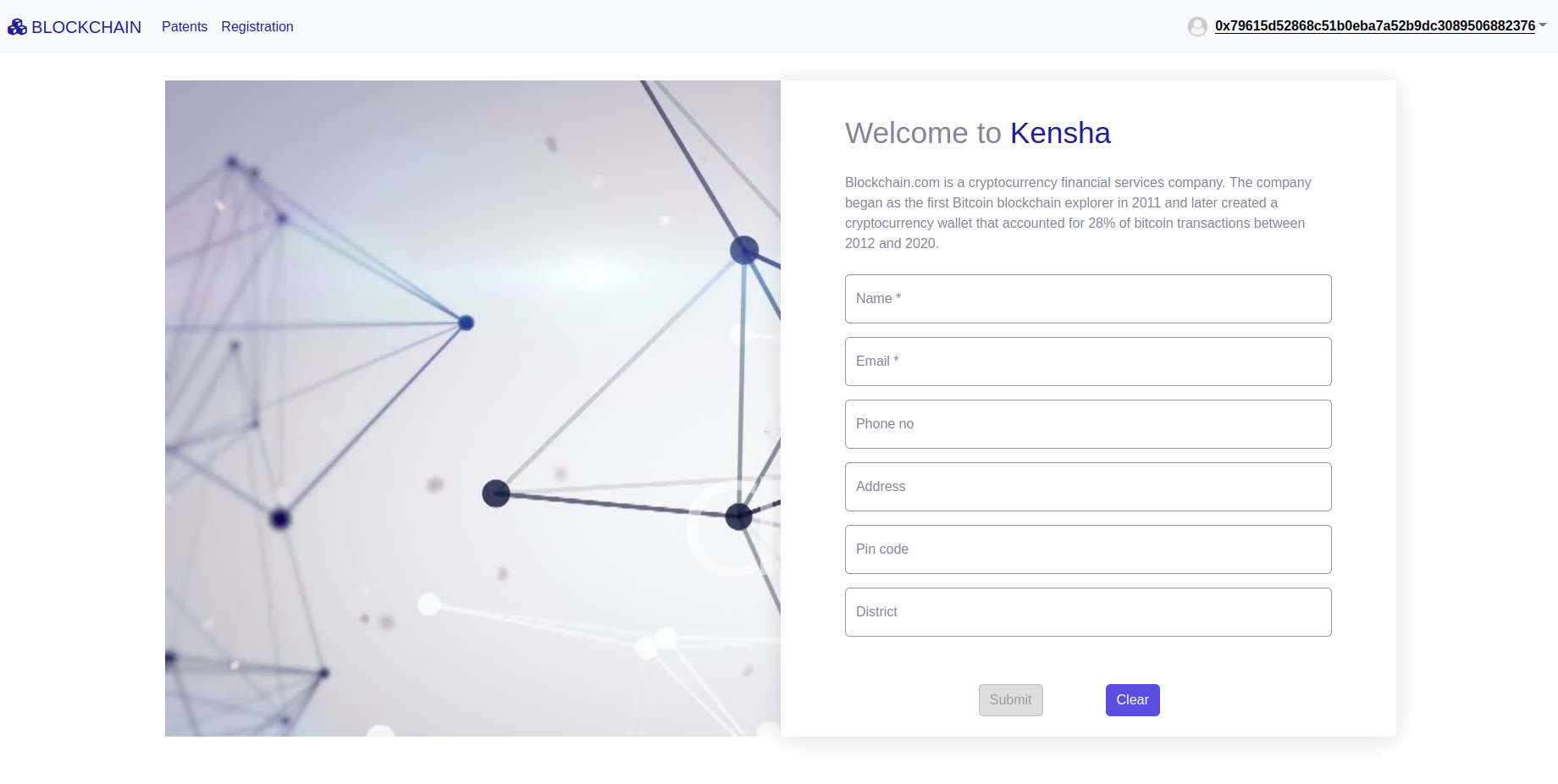
}

# 7. User Interface Design

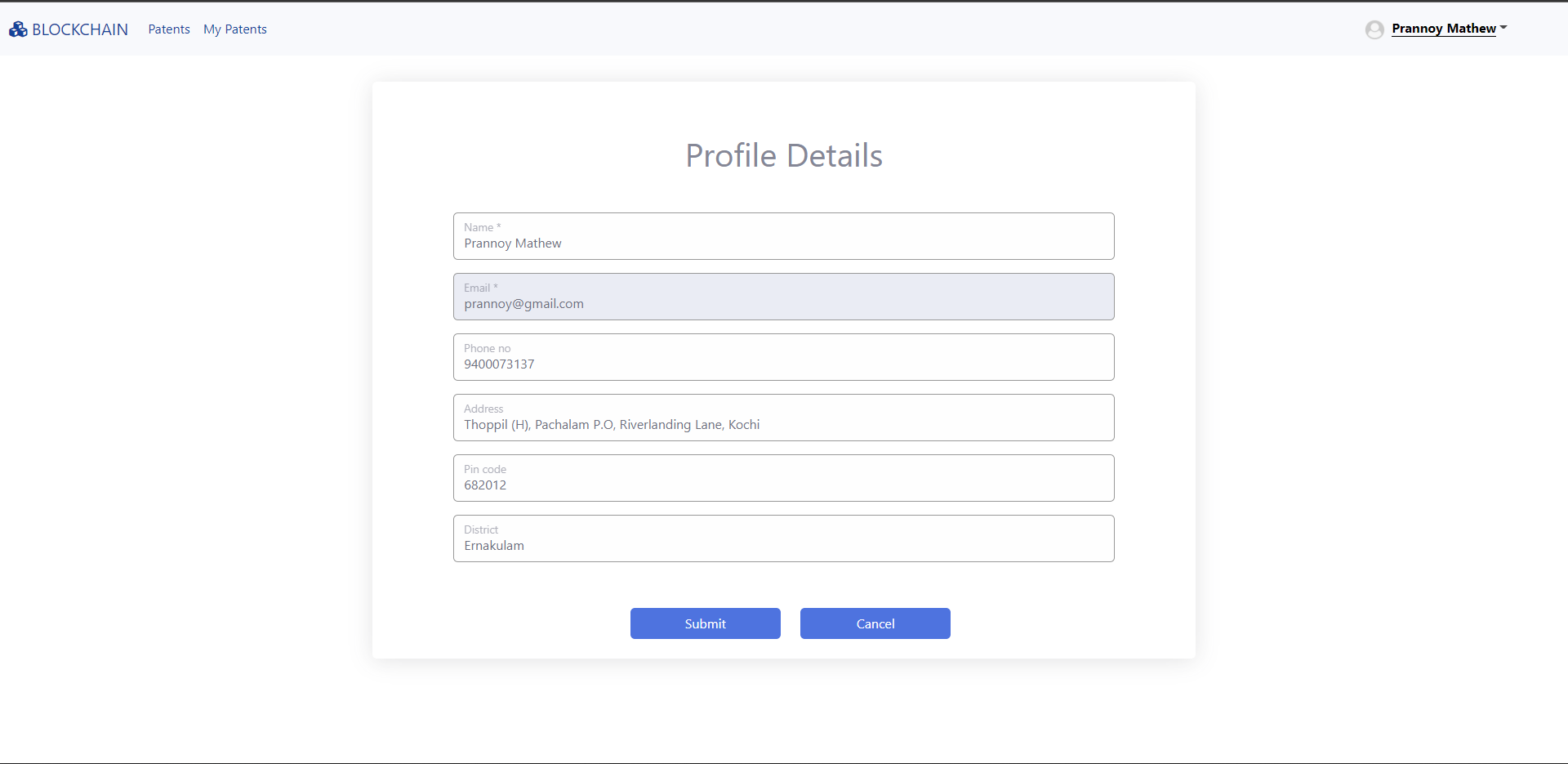
## 7.1 All Patents (Public User)



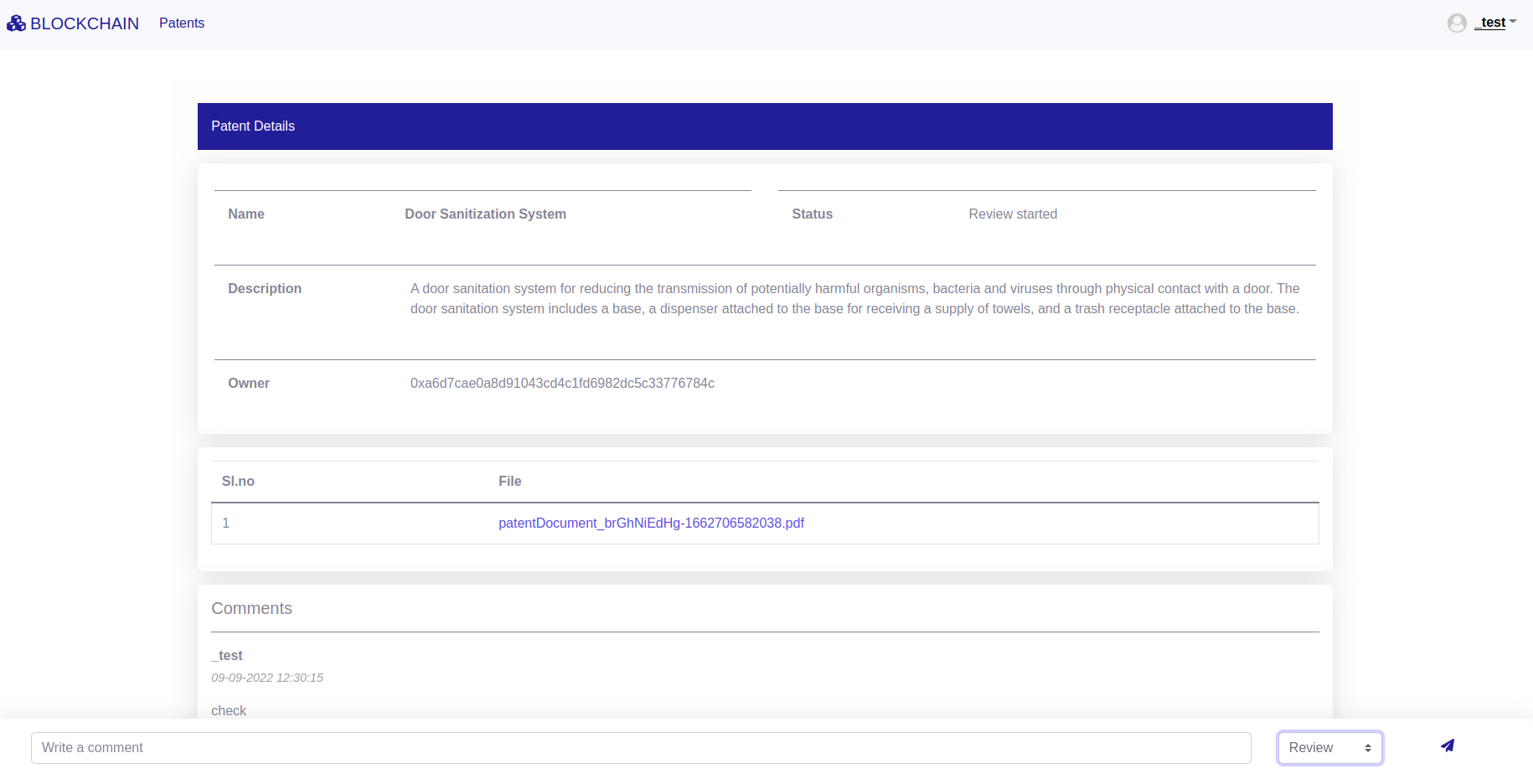
## 7.1 Registration



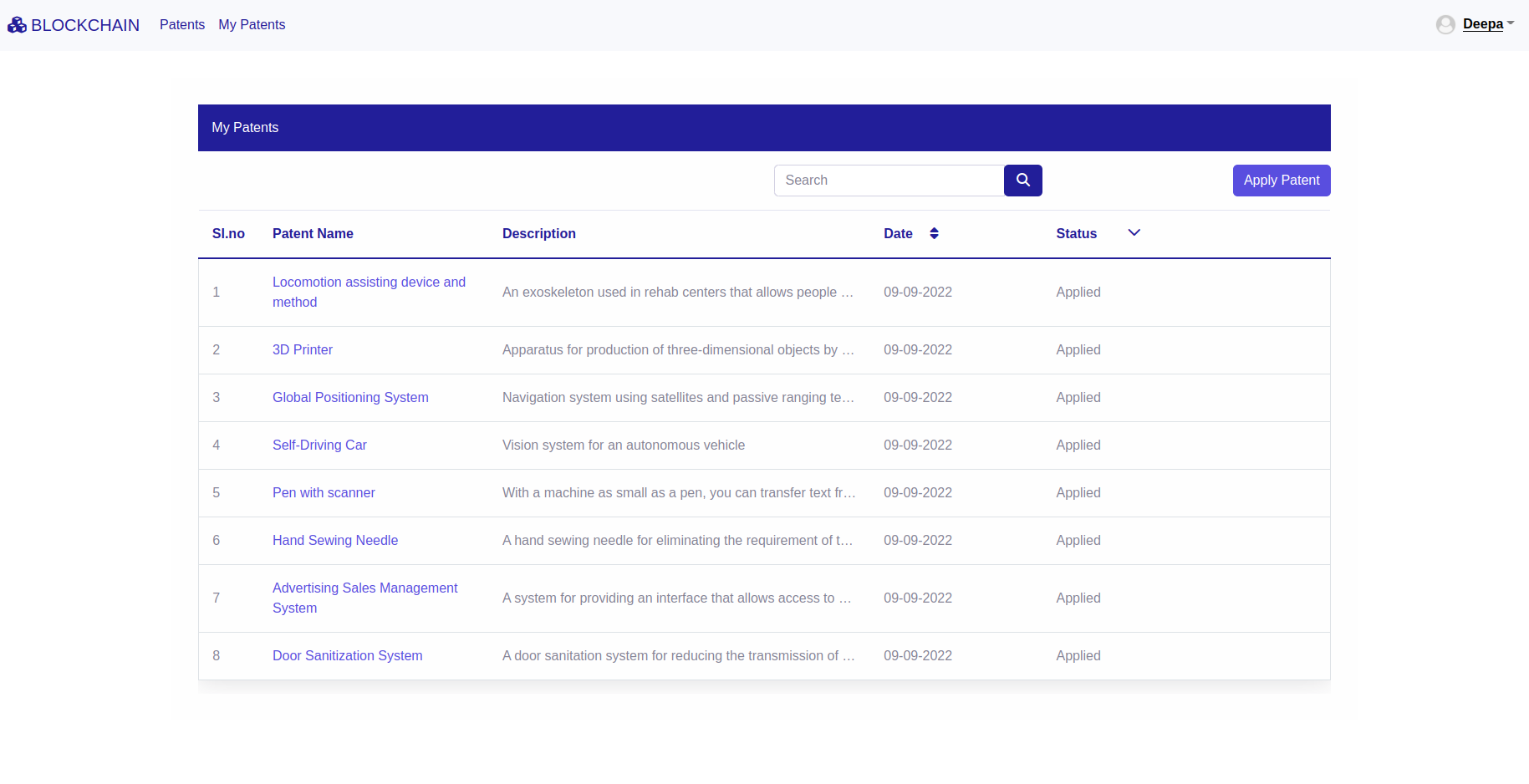
## 7.2 User Profile



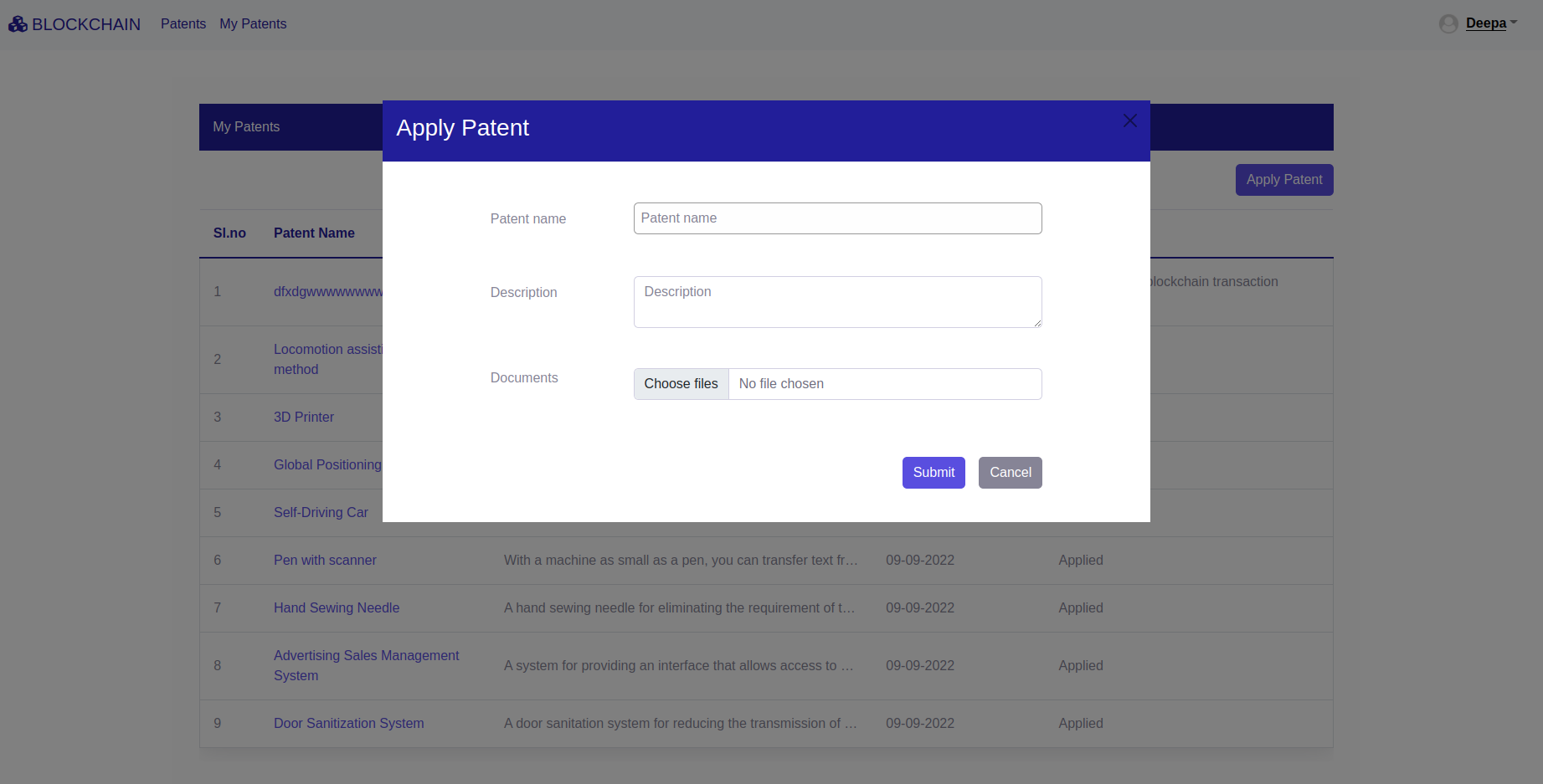
## 7.3 Patent Details



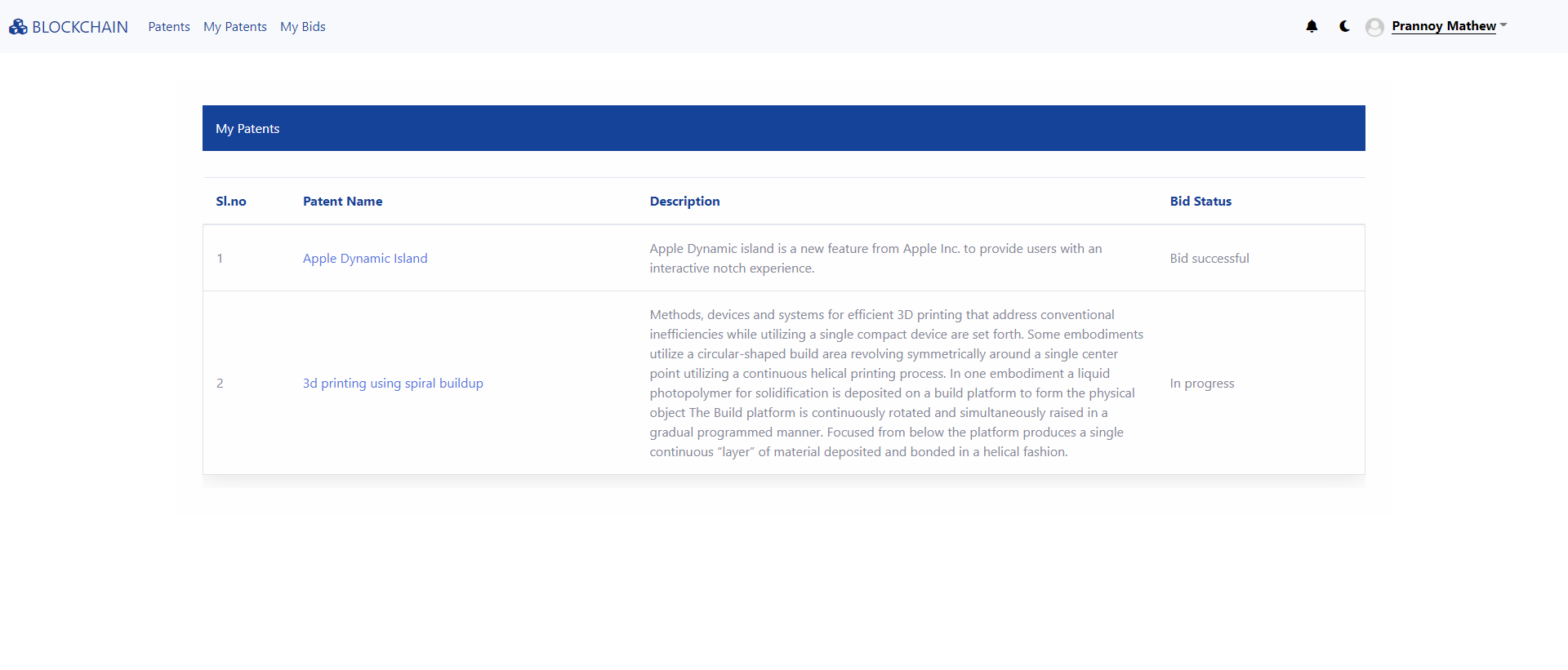
## 7.4 My Patents (User)



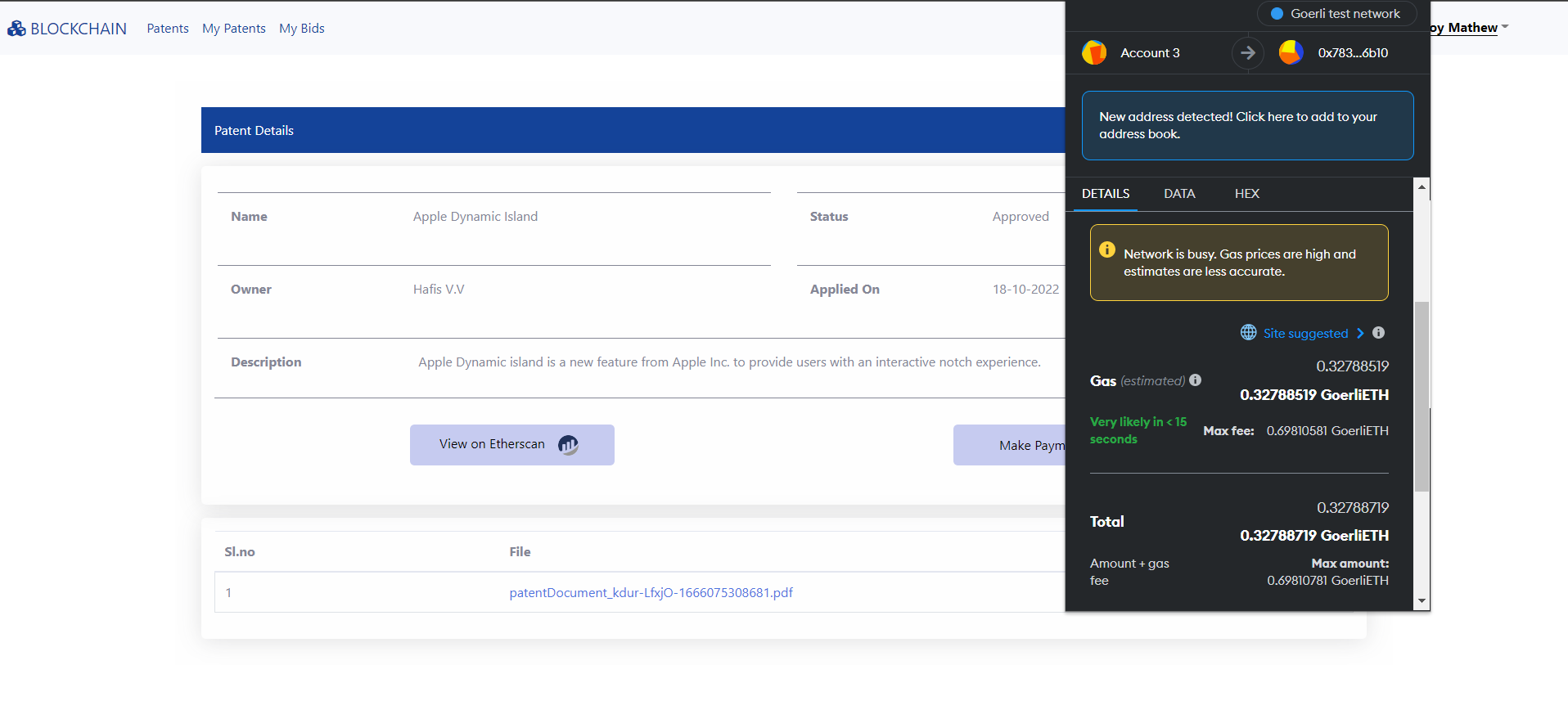
## 7.5 Apply Patent (User)



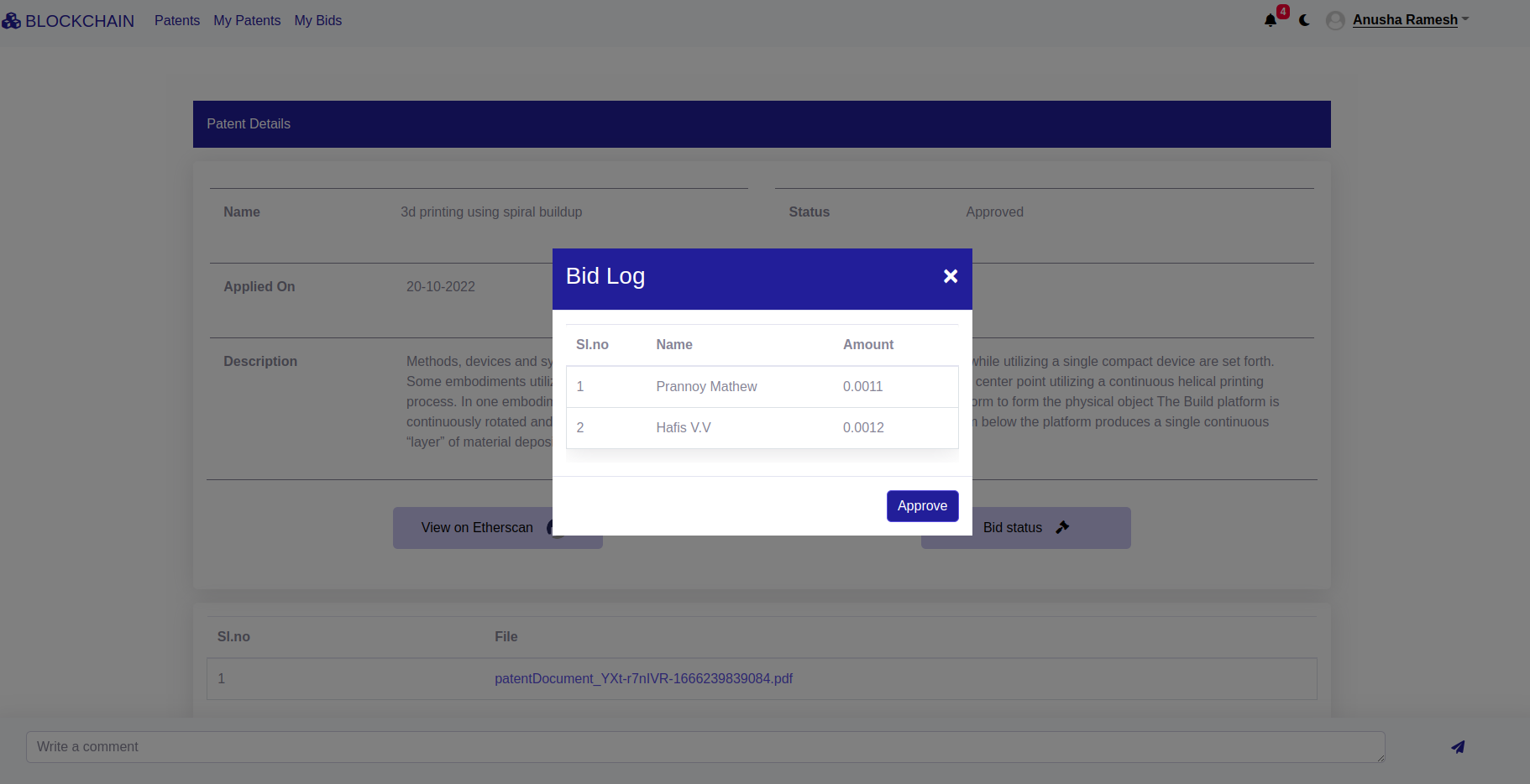
## 7.6 My Bids (User)



## 7.7 Make Payment (User)



## 7.8 Approve Bid (User)



## 7.9 All Patents (Authority)

