**Seed Certification using Blockchain Technology**

**By Team Number 34 Untitled Folder**

**Requirements Analysis**

**1. Introduction**

**1.1. Purpose of the System**

There's a need for an effective traceability solution in the seed supply chain. We need to ensure the maximum quality of seeds taking into account many important criterias such as genetic purity, germination rate, etc. For this, and to improve the seed supply chain system we need a traceable, transparent and immutable seed certification system. Some of the numerous benefits of this system are:

* The best quality of seeds will be produced.
* Transparency that builds trust. In the supply chain. And among the consumers.
* Honesty in the supply chain system due to immutability of ledgers.
* People are more likely to buy if there's seed certification done for the seeds they're buying. Seed certified seeds will increasingly take higher trust in the consumer market.

Seed Certification system using blockchain technology will be helping us in enabling a trust free environment where each and every norms or standards will be fulfilled with any involvement of the central authority. With enabling blockchain in seed certification, traceability will become one of the most known applications since implementing blockchain will be helping us in tracing each and every activity from the initial stage to the final stage and that too, tracing will be done in no time. Immutability since there will be no discrepancies or any unwanted modification to the data as once data is recorded, it cannot be changed by any means therefore maintains data integrity. Transparency as each and every activity will be recorded and can be viewed i.e. from whom or from where the data has been passed onto. Security, a plus point and an inbuilt feature of blockchain in which the blocks are cryptographically linked with each other due to which major attacks are not possible to be conducted over the blockchain platform.

**1.2. Scope of the system:**

**1.2.1. Description:**

The team "Untitled Folder" will design, develop and implement a solution to bring transparency, traceability, and immutability of data to the existing seed supply chain and certification system.

**1.2.2. Assumptions:**

* Team will develop a system that will be autonomous in nature i.e. it will be self-governing and no central authority.
* Traceability will be done efficiently and within no time because of the blockchain technology.
* A trusted reliable system where unauthorized modifications will be strictly prohibited.
* Better role management since we can assign designated roles to designated entities so that they don’t interfere in another’s designated job.
* A certification that’ll be valued more than the current system certification since the certificate that’ll be generated with our system will be verified with proper standardized procedure.

**1.2.3. Project Deliverables:**

* A WebApp
* A Portal/Website as the front end to the app (with Multilingual Support)
* Authentication/Authorisation mechanism for Access Control
* A Chaincode (a.k.a. Smart Contract) that meet the requirements of the proposed system

**1.2.4. Project Constraints:**

* Time: Since the project will be developed during the Hackathon, features that can be developed during the time frame is very limited
* Resources
* Personal Limitation

**1.3. Objectives and success criteria of the project**

The objectives that will lead to success criteria of the project are:

* **Traceability**: The key feature of the system is traceability. It will help is tracking the process from the initial stage till the final stage. It will also help in tracking down the origin of the seed within a few seconds which the current system takes some days to track the route.
* **Autonomous**: The system will be autonomous i.e. it will not be controlled by any centralized or third party authority. Since there will be no central authority which will help in making the system unbiased.
* **Immutability**: Since blockchain is embedded with the feature of immutability in nature, the application will have the capability to eliminate unwanted modification or discrepancies of the data that may lead to unauthorized certification of the seed.
* **Better Role Management**: The system will have the feature where the designated authority will be doing their designated job and not interfering in another job allocated to someone else. Hence, they'll be doing the job that they have to do in the certification process.
* **Verified Certification**: The certificate that will be generated after the approval of all the requirements will be a verified certificate with a tag number such that seed certification is verified and there are no chances of it being unhealthy seed for the consumers end.

**1.4. Definitions, Abbreviations and Acronyms**

* **Blockchain:** Blockchain is a paradigm in the field of computer science that is based on the mechanism of distributed, decentralized and cryptography.
* **Immutability**: Immutability refers to the ability of not changing/modifying data for whatsoever reason.
* **Autonomous**: Autonomous refers to an organization or a body which has no authority i.e. power of self governance. There is no central authority to govern the body or an organization.
* **SPA**: SPA stands for Seed Producing Agency, and is the owner of a dataset of seed lots.
* **STL**: STL stands for Seed Testing Lab that is responsible to test and update the results of the seed.
* **SCA**: SCA stands for Seed Certified Agency that updates the seed details and is also responsible to issue the certificates of seed.
* **Chaincode**: Chaincode also known as Smart Contracts are the contracts written in computer programming language that defines the business logic for a system.

**1.5. Overview**

Summarizing, the reason to build a system over blockchain technology was to bring transparency, traceability, immutability to the current system. The shortcomings of the current system will be solved in the proposed system such as the traceability feature where the process can be traced from the origin to final phase in a span of few seconds, transparency as we can check the status of the process and immutability since there will not be a chance to make any modifications to the datasets. Also, autonomous in nature is a plus point for our system since there will be no central authority and it will act as a self-governance system. Our ultimate goal is to make the system transparent and reliable.

**2. Current System**

**2.1. Existing System**

There isn't a system put in place for quality control of seeds at a large scale that can be overlooked by a competent authority to ensure better seeds and better crops in a way that is traceable, transparent and immutable. In effect creating a gap for a great seed certification system.

**2.2. Problems**

* There's a lack of inspection and verification of the seed quality, which lead to crops grown from poor seeds.
* There's no field inspection done. Seeds are sown but become genetically inferior due to field conditions. And they create poor crops.
* Overall, a lack of inspection and verification to ensure maximum seed quality. At various phases. Due to the complexity of carrying it out.

**3. Proposed System**

**3.1. Overview**

**3.1.1. Blockchain**

The proposed system is seed certification based on blockchain technology. Blockchain is a technology that is based on the concept of decentralization in which each and every block is connected cryptographically therefore maintaining security and reliability. The reasons that made us choose blockchain to create certification are:

* **Decentralized**: Blockchain is based on decentralization where each block is connected to another block which makes a chain. Also there is no third party involved in this network and no central server is there.
* **Secure**: All the data in the network are cryptographically secured using SHA-256 (Secure Hash Algorithm) and makes data reliable. It maintains data confidentiality.
* **Immutability**: Blockchain Network keeps a record of the activities in a network known as ledger. That ledger is validated by each node in a blockchain network and every block contains a ledger. That ledger makes sure that it maintains non-repudiation and accountability.
* **Transparency**: Blockchain is also known to be transparent in nature as we can see how and where data is sent to.

**3.1.2 IPFS (InterPlanetary File System)**

The InterPlanetary File System (IPFS) is a [protocol](https://en.wikipedia.org/wiki/Communications_protocol) and [peer-to-peer](https://en.wikipedia.org/wiki/Peer-to-peer) network for storing and sharing data in a [distributed file system](https://en.wikipedia.org/wiki/Distributed_file_system). IPFS uses [content-addressing](https://en.wikipedia.org/wiki/Content-addressable_storage) to uniquely identify each file in a [global namespace](https://en.wikipedia.org/wiki/Global_Namespace) connecting all computing devices. Some of the components of IPFS are:

* **Distributed Hash Table**: A hash table is a data structure that stores information as key/value pairs. In distributed hash tables (DHT) the data is spread across a network of computers, and efficiently coordinated to enable efficient access and lookup between nodes.
* **Block Exchanges**: The popular file sharing system Bittorrent is able to successfully coordinate the transfer of data between millions of nodes by relying on an innovative data exchange protocol, however it is limited to the torrent ecosystem.
* **Merkel Directed Acyclic Graph**: A merkle DAG is a blend of a Merkle Tree and a Directed Acyclic Graph (DAG). Merkle trees ensure that data blocks exchanged on p2p networks are correct, undamaged and unaltered.
* **Self-certifying File System (SFS)**: It is a distributed file system that doesn’t require special permissions for data exchange. It is “self-certifying” because data served to a client is authenticated by the file name (which is signed by the server).

**4. Tools used to build the system:**

Some of the tools that we have used to build our system are as follows:

* Smart Contracts: Smart Contracts are a piece of code written in computer language to help solving business logic.
* Ethereum Testnet: Ethereum Testnet is another framework to develop our system for prototype model of seed certification.
* Truffle: Truffle is an Integrated Development Environment that acts as an intermediary that helps in communication between smart contracts and frontend application.
* EJS: EJS stands for Embedded Javascript, a framework that we have used to develop the front end of the application.
* Postgresql: Postgresql is an open source database that we have used to facilitate login facility in our system.
* Node JS: The programming language we used to design the backend of the web interface.
* Solidity: The programming language we used to design the smart contracts.
* HTML/CSS: The Programming language we used to design the frontend.

**5. Requirements**

Some of the requirements that are necessary for our system is:

**5.1. Hardware Requirements:**

Monitor: LCD, LED, CRT, TFT

Input Devices: Mouse, Keyboard

Output Devices: Monitor

CPU: Minimum Intel Core 2 Duo Processor with clock speed of 1.3 ghz

Storage: Hard Disk minimum 60 GB

**5.2. Software Requirements:**

Operating System: Windows 7/8/10, Linux (Any Build), MacOS

Browser: Google Chrome, Mozilla Firefox, Opera Mini, Microsoft Edge, Safari.

**5.3 Functional Requirements**

* SG and Farmer should be able to **only view** the details of a Seed Lot/Tag
* SPA should be able to **create**/**register** the seeds
* STL should be able to **update** the test results against the secrete code of the sample
* SCA should be able to **update** the processing details, sample details with secrete code, certificate & tag details against a particular Seed Lot

**5.4 Non-Functional requirements**

**5.4.1 Usability**

User-friendly pages with clear instructions requiring only the ability to read the language. Indian languages and English therefore have the capability of multilingual support.

**5.4.2 Legal**

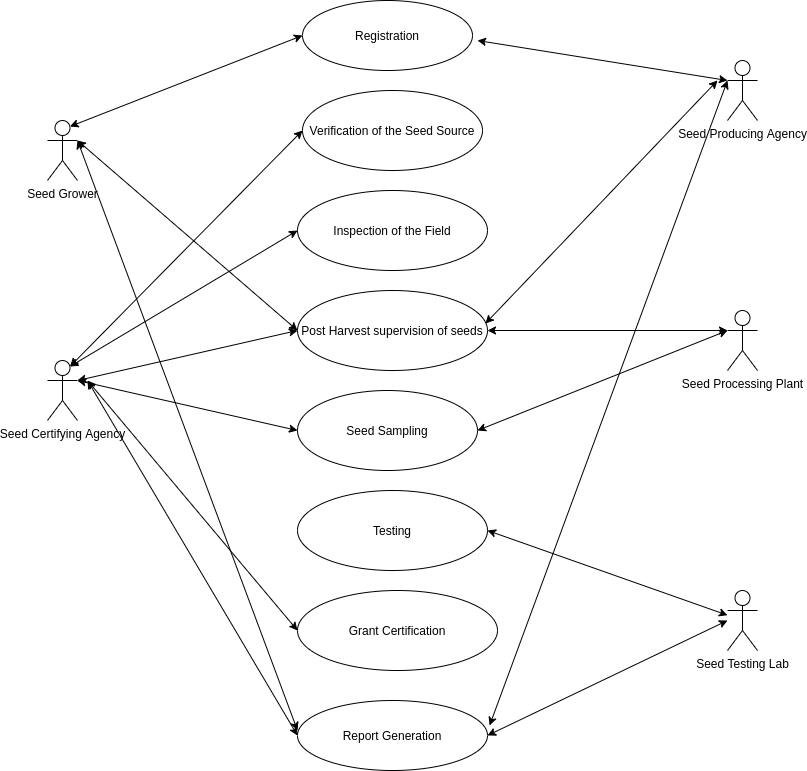
Necessary legal documentation required by users to participate in the seed certification system.

**5.4.3 Documentation required**

* Technical Documents
* Installation Procedure
* Development Platform
* Deployment Architecture
* Technology Stacks
* User Manual
* Test Cases with Test Data

**5.5. System Models**

**5.5.1. Use Case Model**

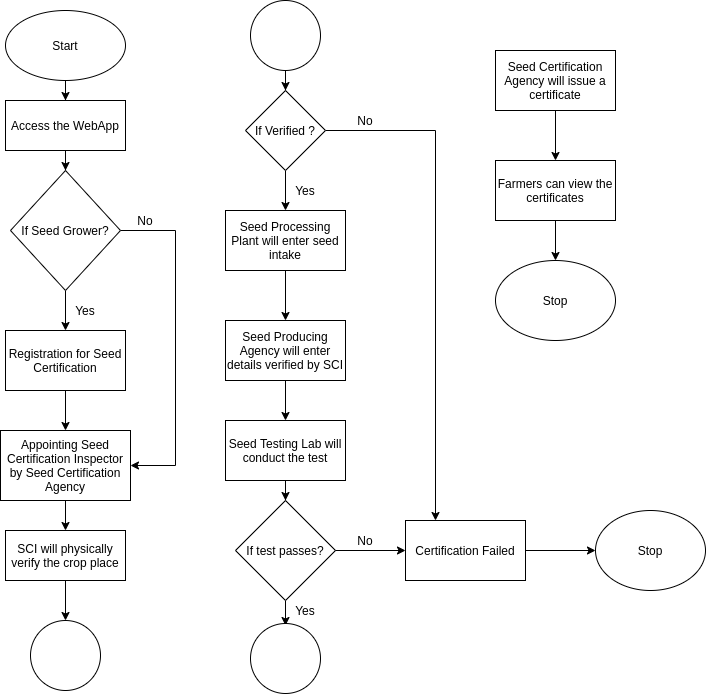


**Fig: Use Case Diagram for Seed Certification**

The above diagram represents the Use Case Diagram for the seed certification process of our proposed system. There are in total 5 actors involved in the whole process, are as follows:

* Seed Grower: Seed Grower is an entity that registers himself for the certification. The activities that he performs are:
* Registration: Seed Grower registers himself to be a certified seed grower by standard norms and procedures.
* Post Harvest Supervision of Seeds: Seed grower looks after the post harvest of the seeds in order to maintain the quality of seeds.
* Report Generation: Seed Grower will be provided with the report of the quality of seed.
* Seed Certifying Agency (SCA): SCA is a certifying agency that verifies and certifies the seed. The activities performed by the SCA are:
* Verification of the Seed Source: SCA is responsible for the verification of the seed source i.e. the origin of the seed.
* Inspection of the field: SCA inspects the field of seed origin. SCA appoints a field inspector to inspect the field.
* Post Harvest supervision of seeds: SCA takes care of seeds even after post harvest in order to maintain quality of seed
* Seed Sampling: SCA does the process of seed sampling by analysing the high quality sample with procured seed.
* Grant Certification: SCA is responsible to grant certification of the seed to ensure quality of the seeds.
* Report Generation: SCA generates report of the seed that contain the characteristics of the seed.
* Seed Processing Plant (SPP): SPP organization is responsible for the processing of the seed. The activities responsible for SPP are:
* Post Harvest Supervision of Seed: SPP takes care of the seed after post harvest in order to maintain the quality of seed.
* Seed Sampling: SPP does the sampling of the seeds in order to know the degree of the seeds.
* Seed Producing Agency (SPA): SPA is a creator of data set lots. The activities responsible for SPA are:
* Registration: SPA is responsible for registration of the seed growers.
* Post Harvest Supervision of Seeds: SPA is responsible for Post Harvest Supervision of seeds.
* Report Generation: SPA is a part of Report Generation as it adds some features of seeds.
* Seed Testing Lab (STL): STL is a body for testing the seeds. They test the seeds in the lab to define the compositions. The activities of an STL are:
* Testing: STL is responsible for testing the seeds in the lab for composition checks.
* Report Generation: STL also takes part in report generation as they have to ensure that the quality of seed is high grade.

**4. Flowchart:**

****

**Fig:** Flowchart of Seed Certification System

**6.1. Description**

The above diagram is the representation of the flow of the seed certification system. The process of seed certification will be as per follows:

* At first, the user will have to have a web browser to access our web application in order to avail the seed certification system features.
* After going to the homepage, the seed grower can register himself so that he can be certified to grow seed.
* After registration, SCA (Seed Certification Agency) will be assigning a SCI (Seed Certification Officer).
* The SCI will be responsible to inspect and verify the location of the seed physically grower so that the seed grower can be certified.
* The SPP (Seed Processing Plant) will therefore enter the seed intake details on the system.
* The SPA (Seed Producing Agency) will be entering the details to the system that are being verified by the SCI.
* Once SPA enters the details, the STL (Seed Testing Lab) will perform testing and sampling of the seed in order to check the grade quality of the seed.
* After STL passes the test, SCA generates a certificate with a tag in order to make sure that seed is being certified by following all the norms and procedures.
* At the user end, farmers and seed growers can check and view the certificate and can even suggest some opinions over the certified seeds.

**7. Algorithmic Approach:**

* The Seed Certification Inspector's name, id is logged. And a string init is set to Init, signalling initialization.
* The duration of the string is set.
* The struct Seed Grower stores the name of the seed grower, the seed grower ID, the location of the string, the adhaar number, the crop seasons, the type of crop and the type of seed grower.
* The struct Seed stores the seed id, the seed lot and the status.
* The struct Test stores the code of the test, the test itself and the date of the test.
* The struct Cert stores the certification number, who the certification is by and the validity of the certification.
* The address of the seed testing lab is stored.
* The address of the seed certification authority is stored.
* The address of the seed processing plant is stored.
* A registered event is set using the status.
* A showDate event is set using the date.
* A few public classe arrays are declared namely Seed\_Grower, Test, Seed and Cert.
* A public constructor for the Seed Certification Agency is created.
* A function to register seedGrower is created with many parameters such as memory name, memory location, aadhaar, memory season, memory cropType, memory sgType is set public. With seed grower pushing out using attributes such as name, id, location, aadhaar, season, cropType and sgType.
* A function for seed entry is created with parameters of seed id and seed lot set as public. There are two requirements. The message sender should not be the seed certification agency and seed testing lab.
* The seed processing plant is the message sender.
* Then seed is pushed with its attributes seed id, seed lot and init.
* A function seed test that is public is created with attributes id, secret code and result. It follows the requirement that the message sender isn't a seed certification agency and seed processing plant.
* The seed testing lab is the message sender.
* A timestamp is taken and stored in date.
* The date is shown. We encounter an if-else condition which test the parameters, secret code, result, date and results in either test successful or test failed according to the test case.
* A final public function for certification is made with parameters as certification number, certification by and id. This requires that the message sender isn't a seed production plant and seed testing lab. A final test condition is passed. If seed id matches the id, a certificate is pushed with certificate no., certificated by and duration.

**8. Pseudocode**

create contract seed\_certifi

set string sci\_name, sci\_id, init values.

set string duration.

create struct Seed\_Grower{sgName, sgID, location, aadhaar\_no, crop\_season, crop\_type, sg\_type}

create struct Seed {seed\_id, seed\_lot, status}

create struct test {code, test, date}

create struct Cert{cert\_no, cert\_by, validity}

initialize address st1, sca, spp;

event registered(strin status);, showdate(date);

public Seed\_Grower[] seedGrower, Test[] test, Seed[] seed, Cert cert;

public constructor{sca = msg.sender;}

function reg\_seedGrower(name, id, location, aadhaar, season, cropType, sgType) public {

seedGrower.push(Seed\_Grower(attributes));

}

function seed\_entry(seed\_id, seed\_lot) public{

require(conditions where msg.sender != sca && != stl);

spp is the msg.sender;

seed.push(Seed(seed\_id, seed\_lot, init));

}

function seed\_test(id, secret\_code, result) public {

require(conditions where msg.sender != sca && != spp);

stl is the msg.sender;

seed.push(Seed(seed\_id, seed\_lot, init));

}

function seed\_test(id, secret\_code, result) public {

require (conditions where msg.sender != sca && != spp);

stl is the msg.sender;

record date = block.timestamp;

showDate(date)

if(result){

test.push(Test(code, result, date));

for(i = 0; i<seed.length;i++)

if seed[i].seed\_id== id

then seed[i] status will result in Test Successful;

}

else{

test.pus(Test(code, result, date));

for(i = 0; i<seed.length;i++)

if seed[i].seed\_id== id

then seed[i] status will result in Test Successful;

}

final certification(cert\_no, cert\_by, id) public{

require(conditions where msg.sender != spp && != stl);

sca is the msg.sender;

if condition

i=0; i<seed.length;i++

if condition

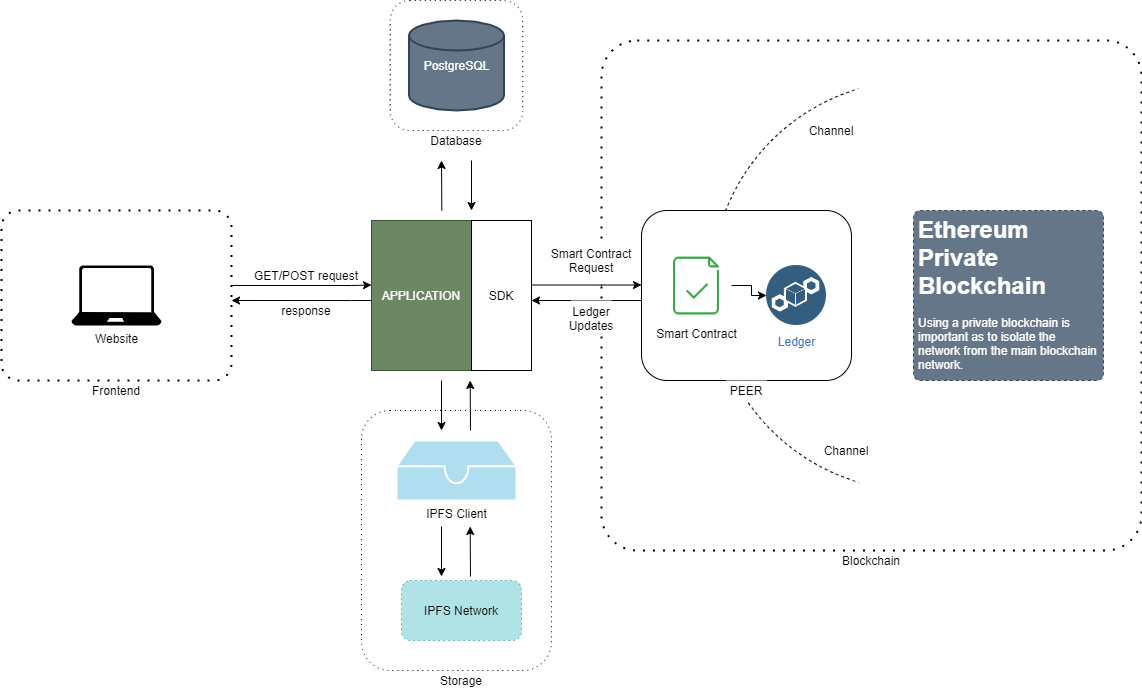
seed[i].seed\_id == id

cert.push(Cert(cert\_no, cert\_by, duration));

Certificate pushed.

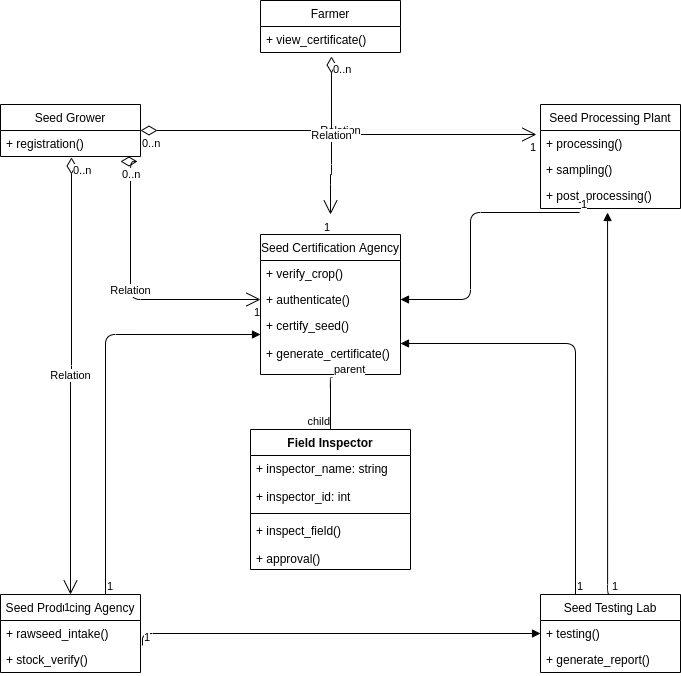
**Architecture Design**

**Architecture Design of the System:**

****

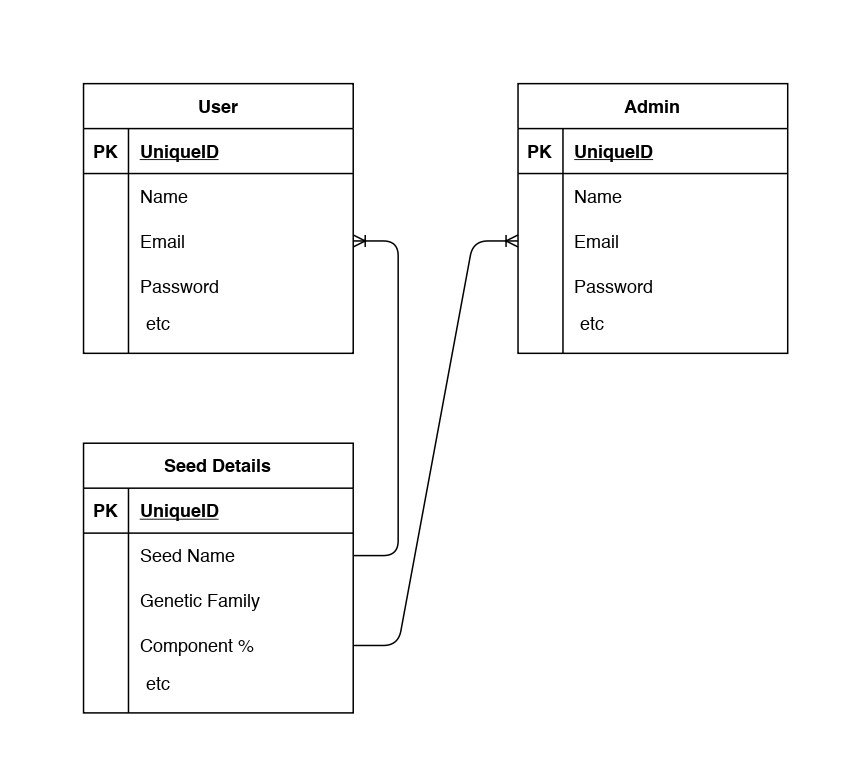
**Fig: Architecture Design of the System**

**Class Diagram:**

****

**Fig: Class Diagram for Seed Certification**

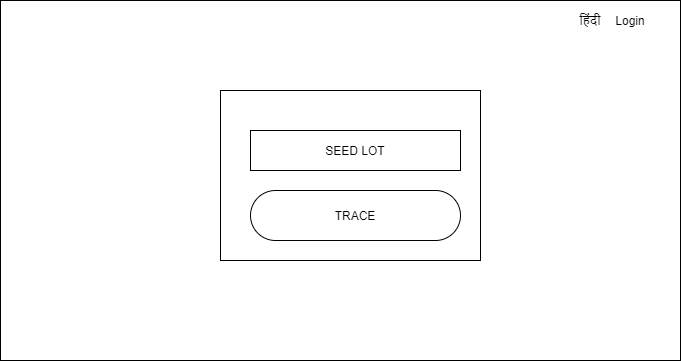
**Database Diagram**

****

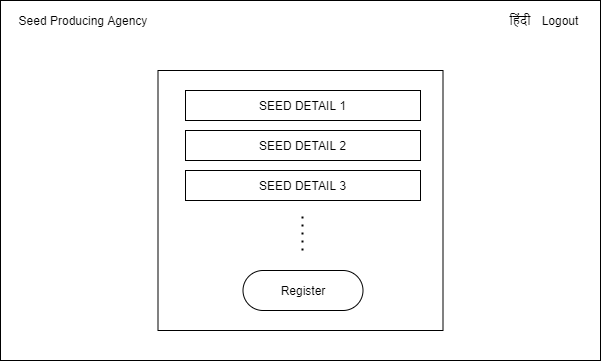
**Fig: Database Diagram for Seed Certification**

**UI - Wireframe**

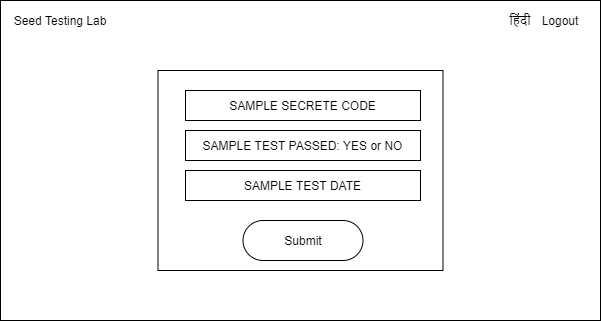
**Home Page**

****

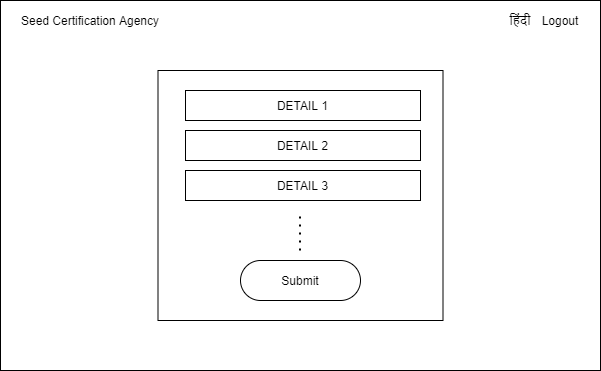
**Seed Producing Agency Dashboard**

****

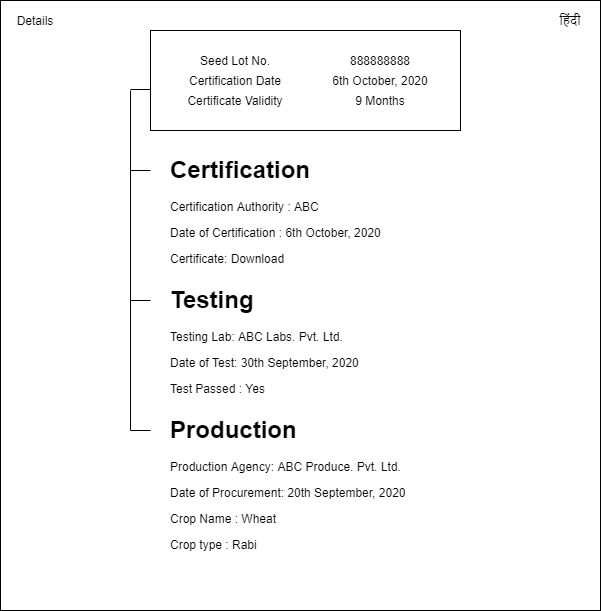
**Seed Testing Lab Dashboard**

****

**Seed Certification Authority**

****

**Seed Details Page**

****