

Smart Agriculture System

MAZR3TY

220200180	عبدالرحمن عيد هنداوي	اسماء اعضاء الفريق
220200084	يوسف محمد الشحات محمد	
220200177	جورج سمير اسحاق	
220200461	علي محمد علي احمد	
220200339	محمد نبيه محفوظ	
220200181	عبد الرحمن خالد عبدالحفيظ	

Testing Levels Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified. Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels

- **Unit Testing** While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free. **Integration Testing** Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updating etc. **76 System Testing** The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

- **Functionality testing** - Tests all functionalities of the software against the requirement.
- **Performance testing** - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.

- Security & Portability - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

Acceptance Testing When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

- Alpha testing - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.

- Beta testing - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

Regression Testing Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code. This is known as regression testing.

Stack holders:

- Farmers: Primary users who directly benefit from the system. They are interested in improved crop yields, reduced resource wastage, and efficient farm management.
- Agricultural Experts: Professionals with expertise in agriculture and technology, providing guidance on system design and best practices.
- Software Developers: Those responsible for creating and maintaining the software components of the system.
- Hardware Engineers: Experts in designing and maintaining the physical devices and sensors used in the system.
- Data Analysts: Professionals who analyze agricultural data to derive insights and optimize farming strategies.
- Government Agencies: Interested in promoting sustainable agriculture and ensuring that the project aligns with regulations and standards.
- Environmental Organizations: Concerned with the project's impact on the environment and sustainability.
- Investors: Individuals or organizations providing financial support for the project in exchange for potential returns on investment.
- Technology Suppliers: Companies supplying the hardware and software components used in the system.
- End Users (Consumers): Individuals or organizations consuming agricultural products produced with the help of the smart agriculture system.
- Local Communities: Communities surrounding the farms affected by the project's outcomes.
- Research Institutions: Organizations interested in studying the project's impact on agriculture and the environment.
- Non-Governmental Organizations (NGOs): Organizations dedicated to sustainable agriculture and rural development.
- Regulatory Bodies: Entities responsible for overseeing and regulating agricultural practices and technology usage.
- Weather Forecasting Services: Stakeholders interested in the project's integration with weather data to improve farming strategies.

Use Case Diagram

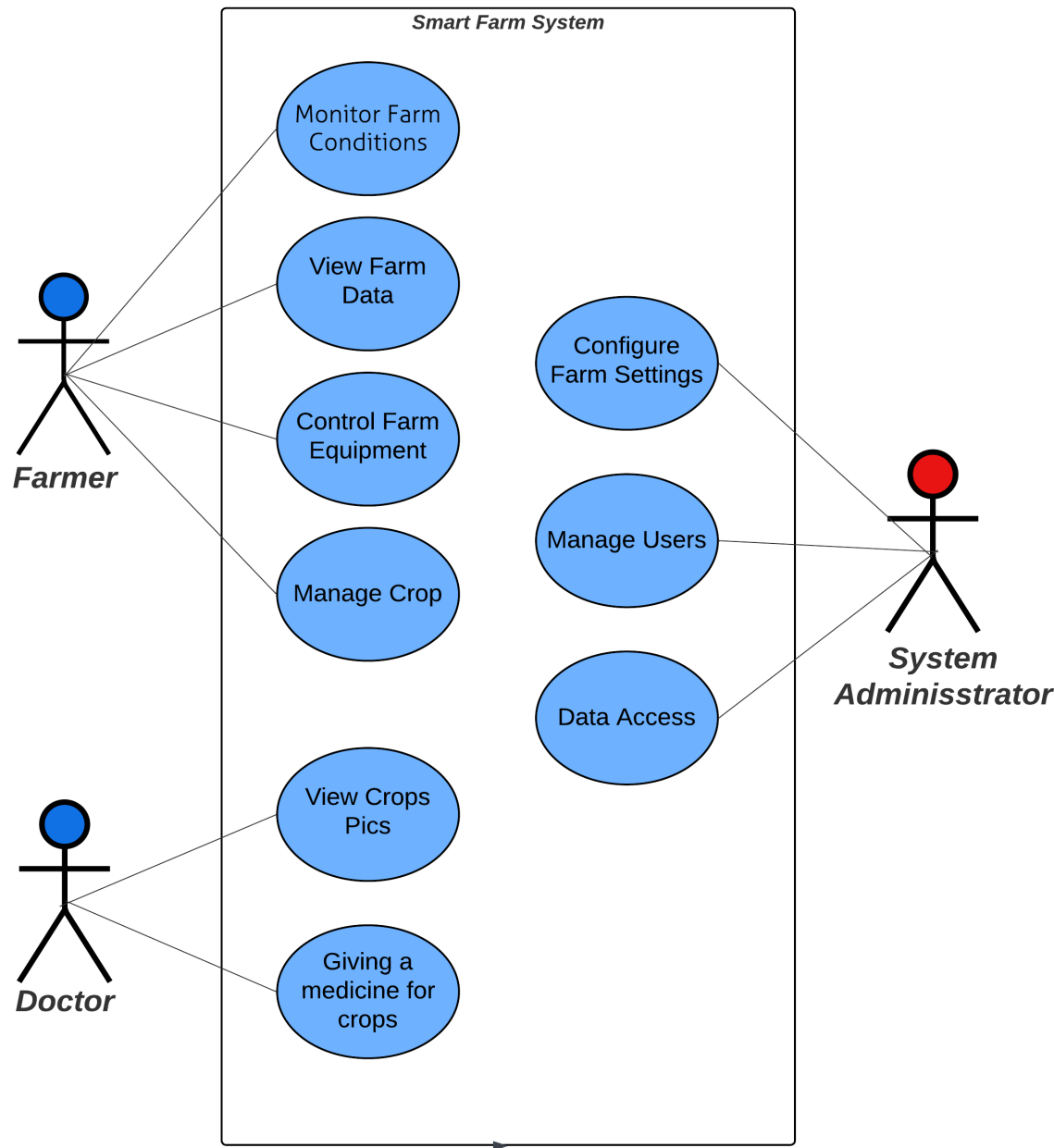


Figure 1: Use Case Diagram

Class Diagram

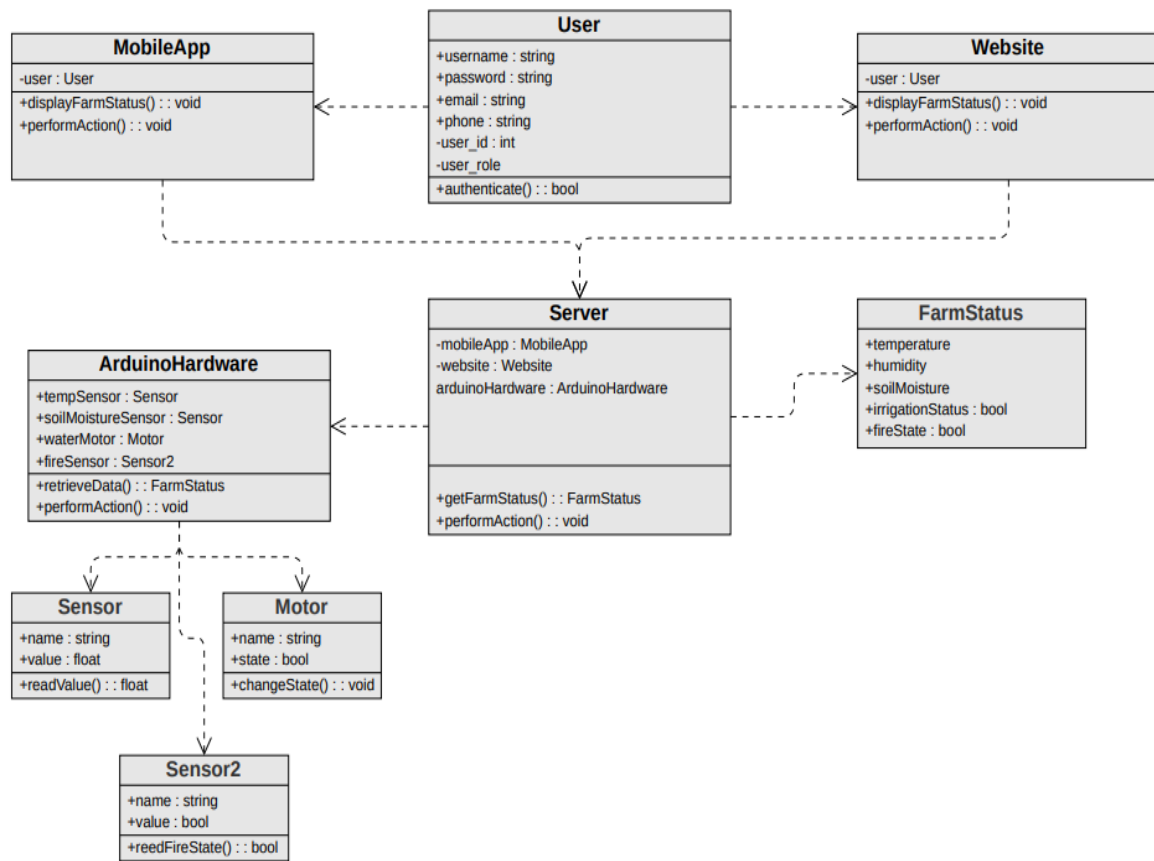


Figure 2: Class Diagram

Sequence Diagram

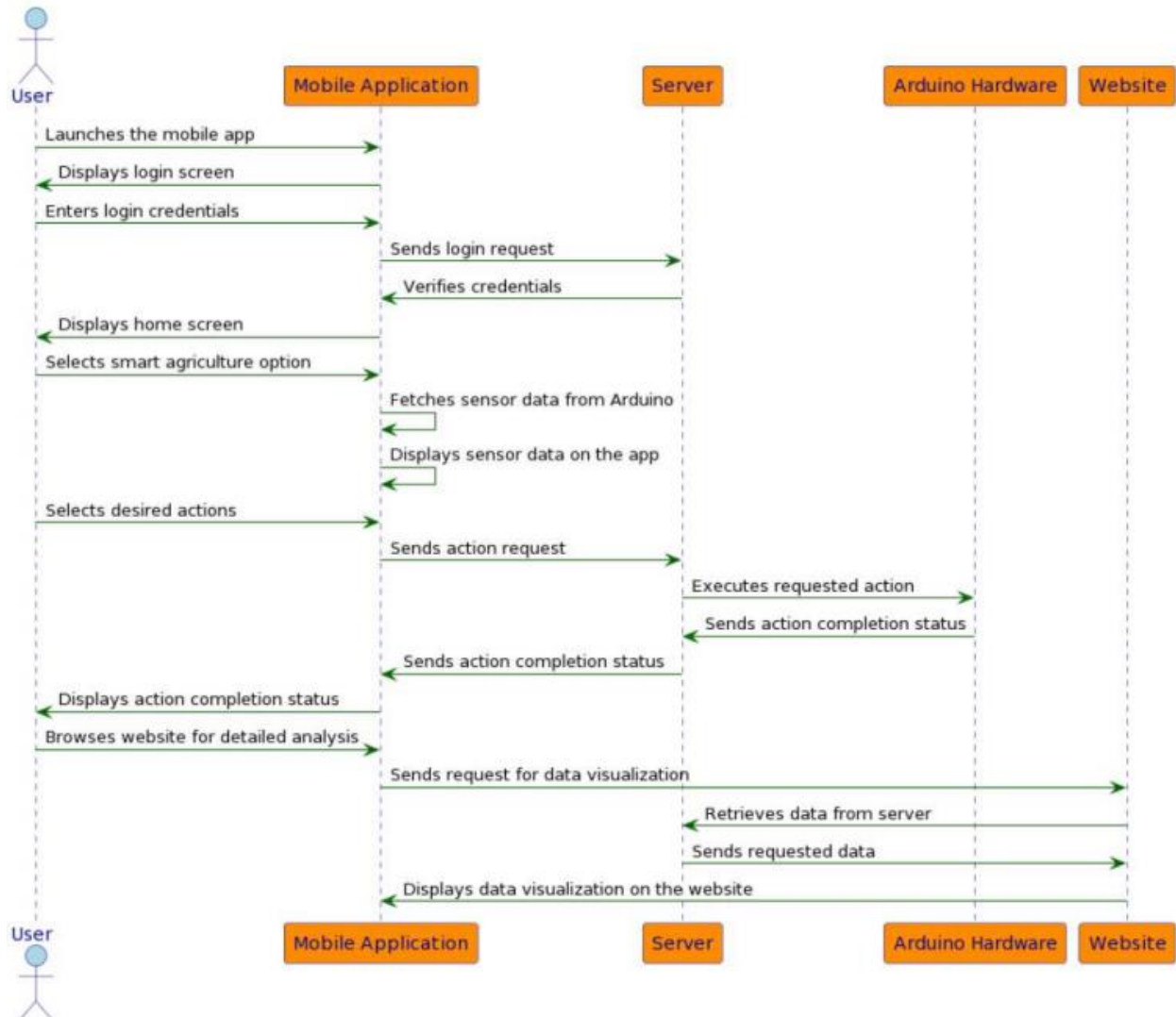
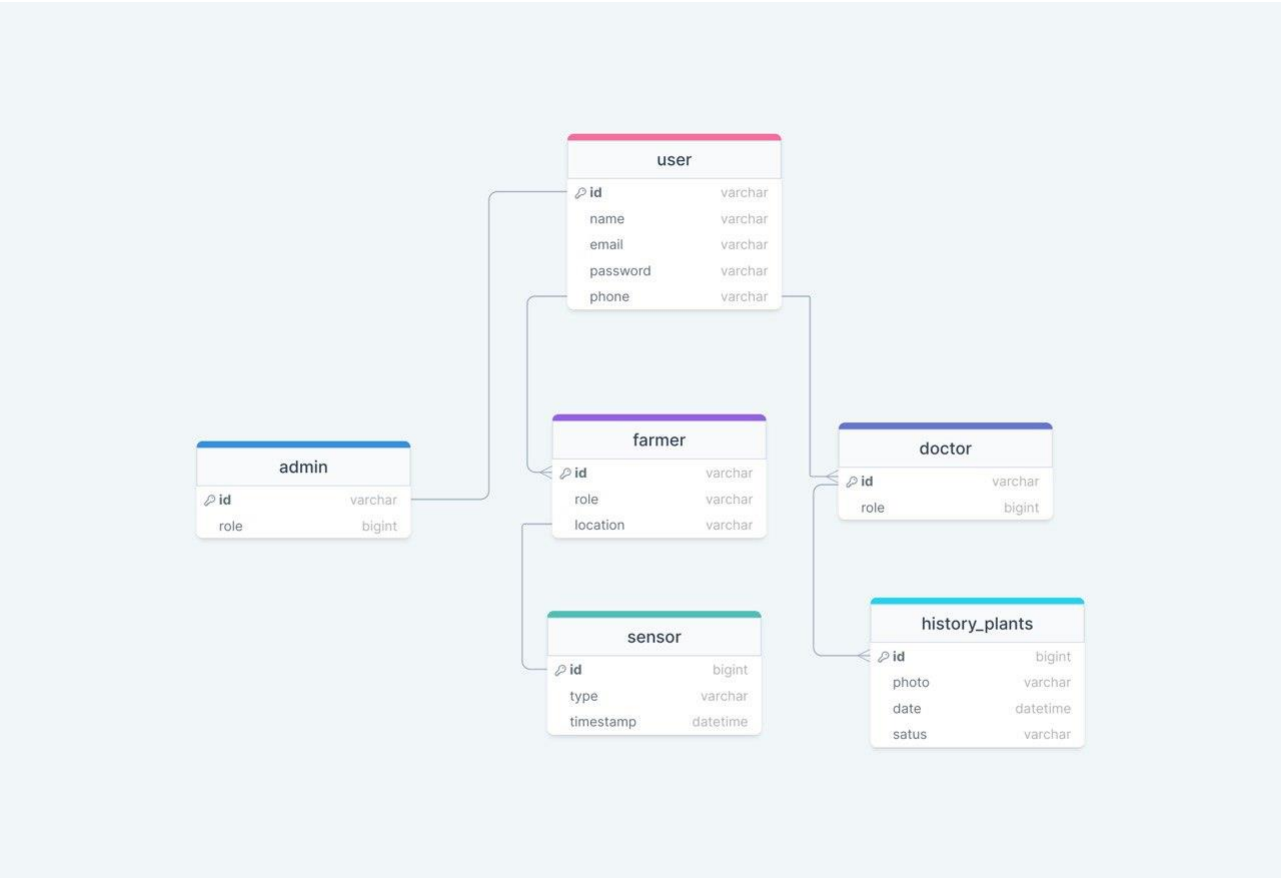


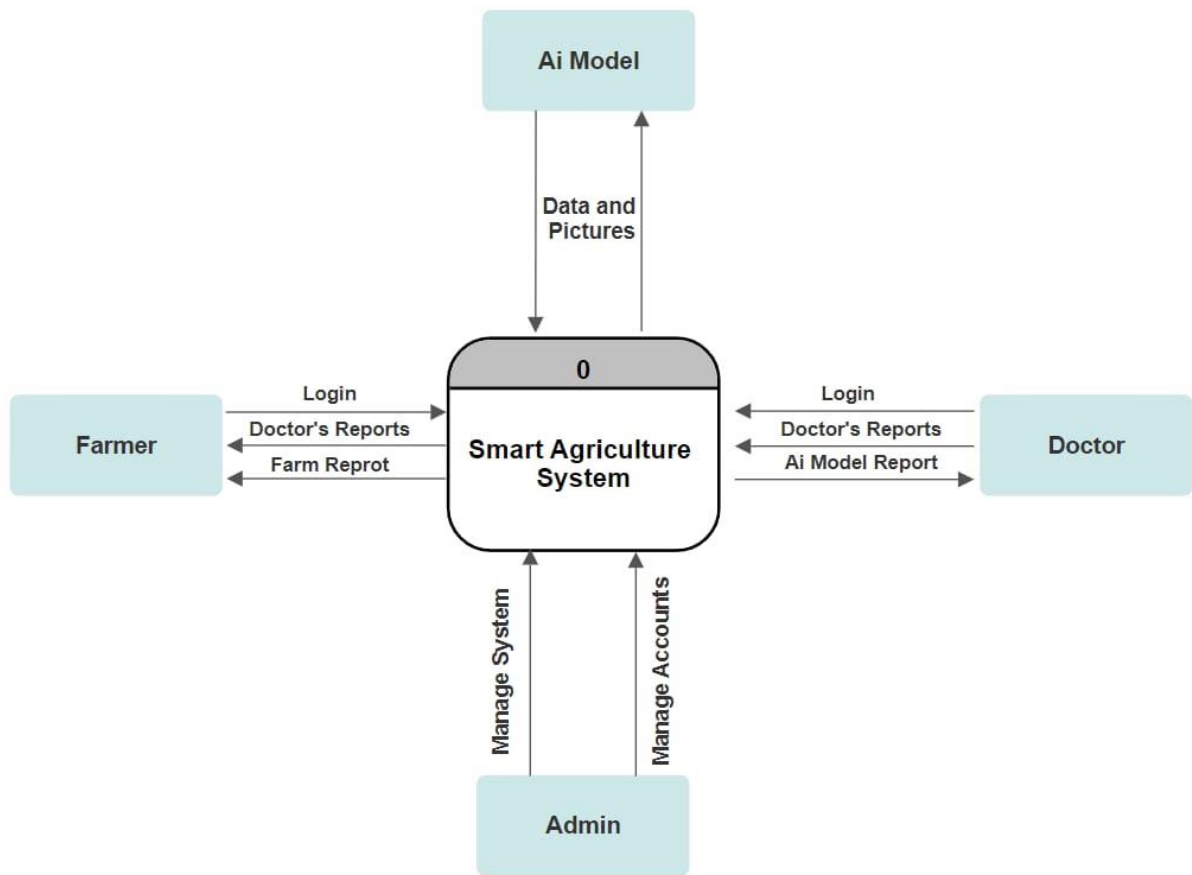
Figure 3: Sequence Diagram

Database Diagram

Figure 4: Database Diagram



Context Diagram



ERD Diagram

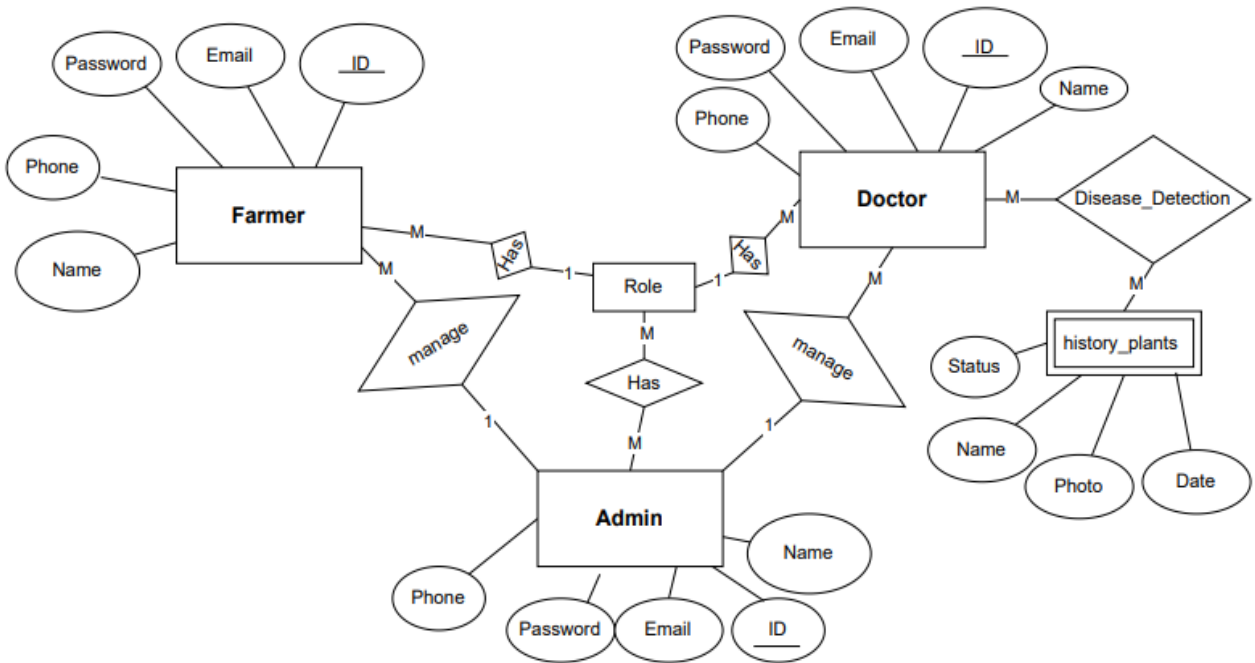


Figure 5: ERD Diagram

Used technologies and tools:

Software:

Front-end:

- Html-CSS
- JavaScript
- Bootstrap
- jQuery
- Typescript
- Angular 16

Flutter App:

- Dart
- Bloc state Management
- Firebase
- http

Back-end:

- Nodejs
- Postman
- Express JS
- Nodejs

Database:

- Mongoose

Artificial Intelligence:

- Deep Learning
- Computer Vision
- Image Processing

IDE:

- VS-Code
- Android Studio

- PyCharm
- Arduino IDE
- Jupyter
- Google Colab

Hardware:

- Node-MCU (ESP-8266)
- Arduino UNO
- Arduino UNO Mini
- ESP-32
- ESP-Cam
- Soil Moisture
- Water Level
- Water Pump
- DHT22
- NPK
- Fan
- Rain Drop
- Flame
- Resistors
- Transistors
- Capacitors
- Jumper Cable
- Relay Module