**Ngene Ugochukwu**

**Applied Project and Thesis Progress**

High Level Design

The overall system is being run in asp.net core and will make use of the service, controller, repository architecture.

* **The User Interface (UI) Module** serves as the front-facing component of the system, allowing farmers to interact with the platform through a web-based dashboard built with Angular. Farmers can upload crop images, enter crop details, and receive AI-driven insights on their crops.
* **The Authentication Module** ensures secure access to the platform by managing user logins, roles (farmer, admin), and session security using JWT-based authentication.
* **Crop Image Processing Module** handles storage by saving the image to AWS S3 and extracting the image URL. This URL is then passed to the AI Crop Analysis Module, where an AI model processes the image to detect diseases, deficiencies, and overall crop health. The AI model assigns a health score (1-5) and identifies any specific issues affecting the crop.
* **Recommendation Module** takes in information from user, real time climate data and retrieves crop-specific fertilizer guidelines from a database sourced from the University of Georgia’s Crop Fertilization Document. This module matches the AI analysis results with predefined NPK (Nitrogen, Phosphorus, Potassium) values and application instructions, ensuring that farmers receive tailored advice based on both AI diagnostics and scientifically proven recommendations.
* **Database Module** allows for the storage of all user information, AI analytics, crop data in a single platform for easy retrieval for serving to UI.

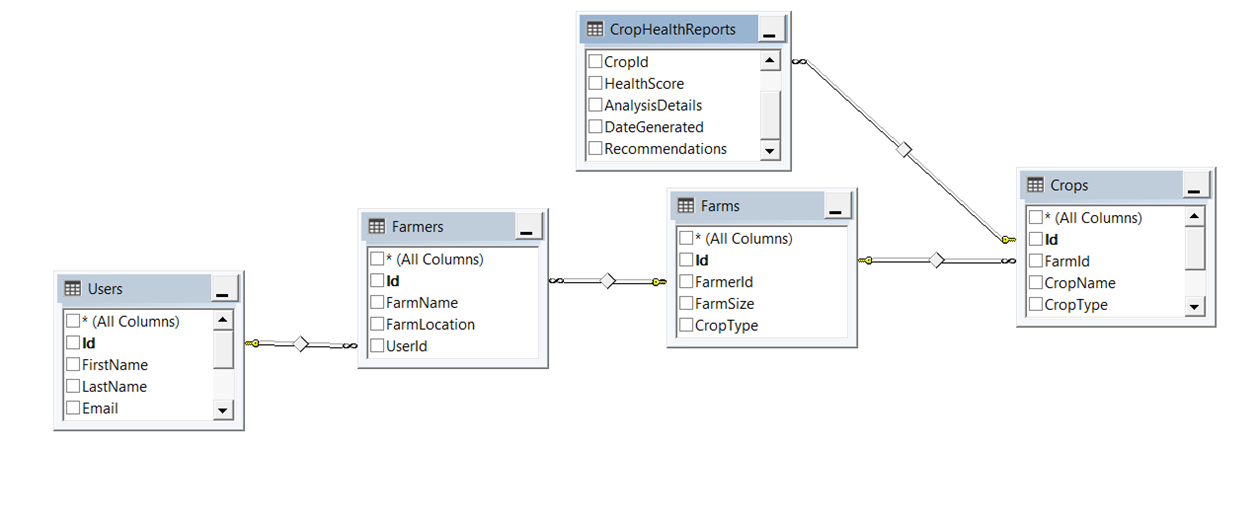
By structuring the platform around these core modules, **CropWise** ensures a seamless workflow—from image upload to AI-driven analysis, fertilizer recommendations, and final report delivery to farmers.

Low Level Design

* **Authentication Module** makes use of jwt tokens that are created in the backend and sent to the frontend to be injected in headers during every api call. Uses generatejwtToken() function.
* **Crop Image Processing Module h**andles image uploads, stores them in AWS S3, and retrieves the image URL. Makes use of amazon’s aws s3 sdk for integration to external api. The url is then passed on to gemini 1.5 flash api for visual reasoning.

This module uses the following functions: Task<IActionResult> UploadImage(IFormFile imageFile), Task<string> ConvertImageUrlToBase64(string url), Task<string> CallGeminiAIAsync(string imageUrl, string googleApiKey). When information is finished processing analysis is stored in our database in the crophealthreports table under column, AI analysis.

Database module design:



1Database Schema

* **Fertilizer Recommendation Module**

Retrieves crop-specific fertilizer recommendations from the database based on the AI analysis results.

Class Structure: (FertilizerService.cs) Retrieves NPK values and application guidelines based on crop type and health score. No recommendation if health score is 4 or 5.

Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Result** |
| 1 | Upload a valid crop image | 1. Navigate to the Crop Analysis Tab  2. Upload Crop image. | Image is successfully uploaded to AWS S3, and the image URL is returned. | Pass |
| 2 | Upload an invalid file format | 1. Select a non-image file (PDF/DOCX). | System should reject the upload and display "Invalid file format" error. | Pass |
| 3 | Trigger AI analysis with a valid image | 1. Upload a valid image. 2. Click on Analyze image button | AI returns health score (1-5) and diagnosis. | Pass |
| 4 | AI detects a healthy crop (4-5 score) | Upload a healthy crop image. | AI should do an analysis of the crop and return a crop health score between 4-5 | Pass |
| 5 | AI detects an unhealthy crop (1-2 score) | Upload an unhealthy crop image | AI should do an analysis of the crop and return a crop health score between 1-2 as well as recommendations. | Pass |
| 6 | Handle missing crop type in fertilizer lookup | 1. AI assigns a health score. 2. System queries the database for an unknown crop. | System should return "No recommendation found" message. | Fail |
| 7 | System logs analysis results | 1. Perform an AI analysis. 2. Check the Database for stored results. | System correctly logs image, health score, and diagnosis in the CropHealthReports table. | Pass |