CROP DEAL



**Low Level Design (LLD)**



Kovvuri Tejaswi Sai Keerthi Reddy

Batch : - **3**

Date of Started : - 13/02/2023

**Document Approval :**

**Approvers of this Document :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Department | Role | Signature | Date |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Document Change History : -**

|  |  |  |  |
| --- | --- | --- | --- |
| Document version # | Author | Date | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

**1.0 Objective……..……………………………………………………05**

**2.0 Why this Project…………………………………………….…..05**

**3.0 About this Project………………………………..……...…05-06**

**4.0 Design Pattern…………………..…………………..……....06-07**

**5.0 Architecture of the Project……..……………………..…..…07**

**6.0 Flow Diagram……………………………………………..…..…08**

**7.0 E-R Diagram(Entity-Relationship Model)….….…………09**

**8.0 Use Case Diagram…………………………………….……...…10**

**9.0 Class Diagram….………………………………………....…..…11**

**10.0 Requirements…………………………………………........12-13**

**10.1 User Requirements…………………………………..…..…12**

**10.1.1 Hardware………………………………………….…….12**

**10.1.2 Software………………………………………….……..12**

**10.2 Developer Requirements………………………….……..12**

**10.2.1 Hardware………………………………………….…….13**

**10.2.2 Software……………………………………….………..13**

**10.3 Technologies…………………………………….………….13**

**11.0 Solution Steps…………………………………………........13-15**

**11.1 User Registration……………………………………..…...13**

**11.2 Crop Listing…………………………………………..…..…13**

**11.3 Crops on Sale……………………………………….……...14**

**11.4 Payment……………………………………………….….....14**

**11.5 Admin Module……………………………………………..15**

**12.0 Classes/Functions…………………………………………..16-17**

**13.0 Data Model………………………………………………….…...18**

**14.0 API Canvas………………………………………………………..19**

1. **HTTP Status Code……………………………………………...19**

**16.0 Unit Testing……………………….......................................20**

**1.0 Objective :-**

Crop Deal Management Application or Website is very useful for the farmers to sell their crop at a perfect rate and they will be paid for their struggle also. It also helps dealers to find their needed crop location and buy them in a easy way using this application.

1. **Why this project ??**

Now a days we are seeing how much struggle the farmers are facing to sell their crop in the market for proper price. Farmer must bear the cost of transportation, wait time, negotiations for proper price, etc. Even though farmer sells the product in the market, he/she must pay lot of intermittent charges as commissions to make his way for coming out of market.

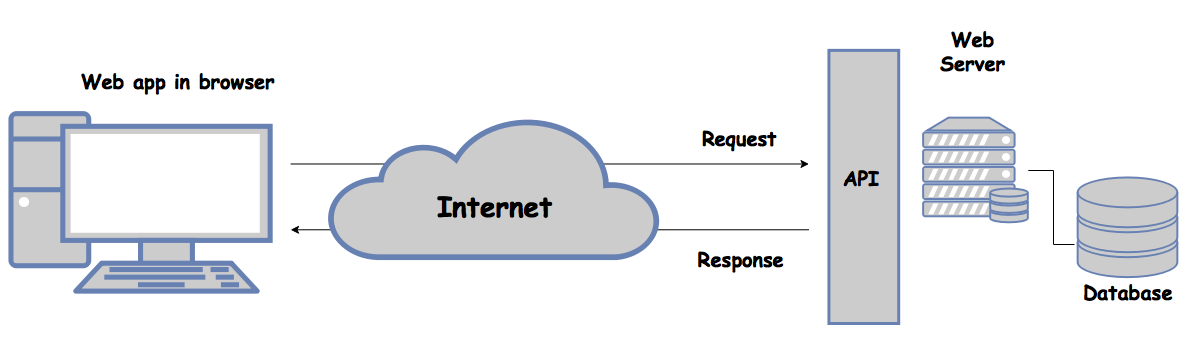
1. **About this project ??**

To Develop a platform (Android/iOS app) which helps the farmers to sell their product directly to dealers from Farm. The App should act as a bridge between the Farmer and the Dealer (Crop purchaser). All you need to do is selecting whether you are a Farmer or the Dealer, signing up with details and the payment information. There is no need for the farmer to carry the crop till the market, paying unexpected commissions and wait for proper price from the Dealers. Whenever farmer want to sell the vegetables/fruits at the Farm itself, he just wants to select the type of Crop, quantity available, input the location/address and publish the information to Dealers.   
Whoever(Dealer) is interested in purchasing will connect with the Farmer, will reach the location, checks the quality of the crop, negotiates the price, weighs the crop and payment will be done to the farmer from the App itself which will be more   
transparent rather than paying tolls/commissions.

1. **Design Pattern:-**

|  |  |  |
| --- | --- | --- |
| S.No | Name | Description |
| 1 | ASP.NET Core Web API | Using HTTP requests, we will use the respective action to trigger various operations. |
| 2 | Angular | To create and design the frontend |
| 3 | SQL | To store and retrieve the data |

1. **Solution Diagrams :-**



Http Protocols

GET,

POST,

PUT,

DELETE

**CLIENT**

**5.0 . Architecture of the Project : -**

**Diagram

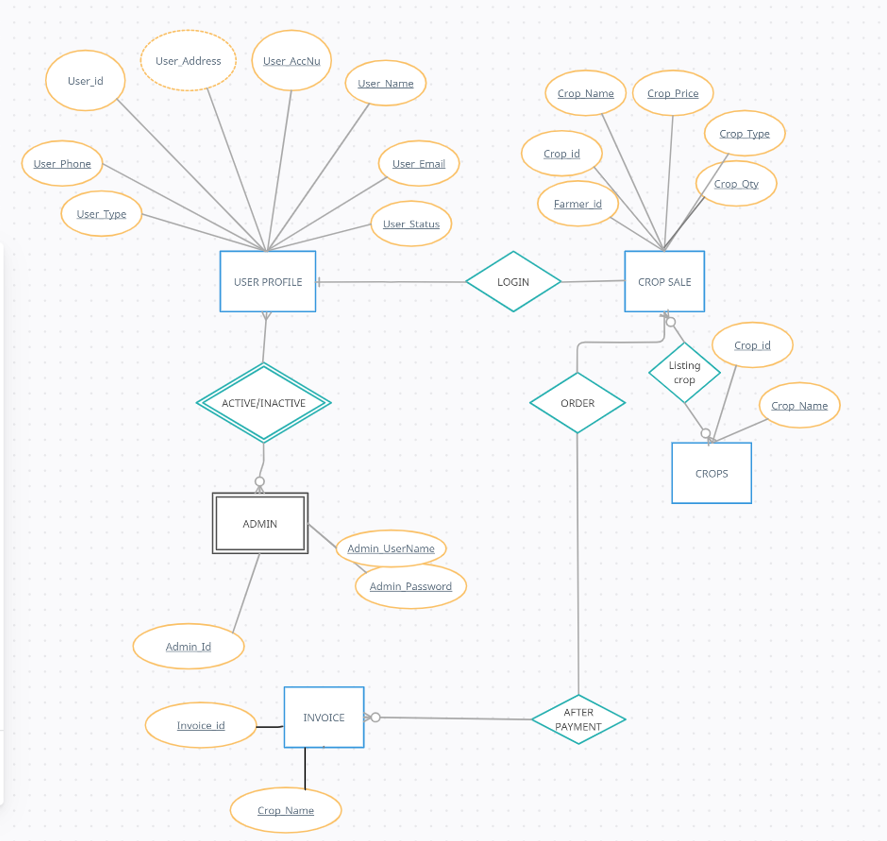
Description automatically generated**

1. **Flow Chart :-**

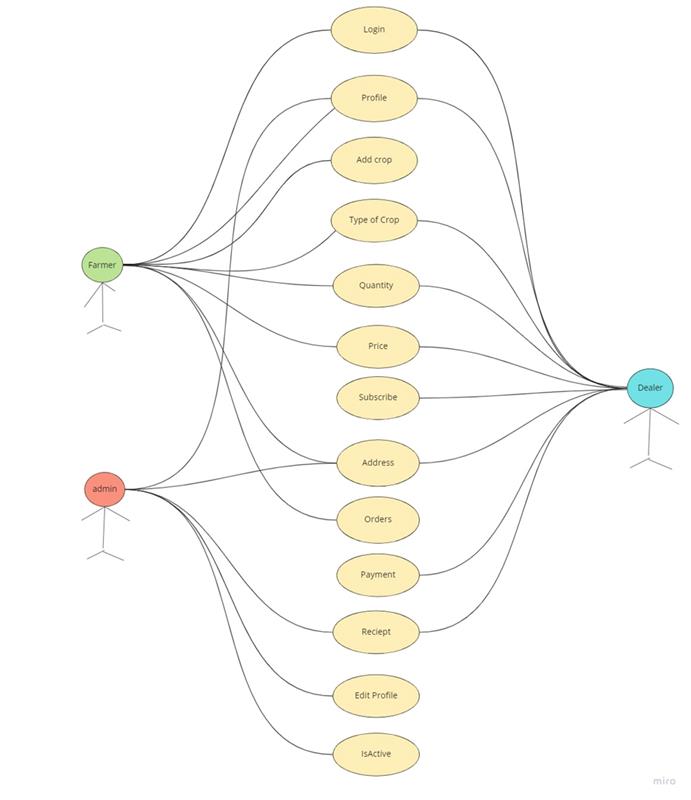
**Diagram

Description automatically generated**

1. **ER-Diagram(Entity-Relationship Model):-**

****

1. **USE-Case Diagram:-**



1. **Class Model Diagram:-**

**Diagram

Description automatically generated**

1. **Requirements:-**

**10.1 User Requirements :-**

User is nothing but who uses our app/Website that means they are Farmer and Dealer .The basic requirements for users to use our app was

**10.1.1 Hardware :-**

* Processor : Minimum 1.8Ghz . Recommended 2Ghz or more.
* Ethernet Connection (LAN) OR a Wireless Adapter(Wi-Fi).
* Hard Drive : Minimum 100GB,Recommended 500GB or more.
* Memory (RAM) : Minimum 4GB,Recommended 8GB or above
* Operating System : Windows.

**10.1.2 Software :-**

* Any Latest Browsers.

**10.2 Developer Requirements :-**

**10.2.1 Hardware :-**

* Processor : Minimum 1.8Ghz, Recommended 2Ghz or more.
* Ethernet Connection(LAN) OR a wireless Adapter (Wi-Fi).
* Hard Drive : Minimum 500GB, Recommended 500GB or more.
* Memory(RAM): Minimum 4GB, Recommended 8GB or above.
* Operating System: Windows.

**10.2.2 Software :-**

* Visual Studio 2019.
* Sql server management studio (ssms).
* ASP.NET core Web API.
* Visual Studio Code.
* Angular.

**10.3 Technologies :-**

* ASP.NET Core Web API for backend
* SQL for the Database Operations
* Angular for Frontend.

**11.0 Solution Steps :-**

**11.1 User Registrations : -**

1. The user is nothing but a Farmer or a Dealer.
2. First, User will select their role.
3. Then, they will enter their Name, Phone Number, Bank Details, Address during Registration Process.
4. If they are not a new user then they should login with their Credentials like User ID and Password.
5. The input validation will be done
6. If validation fails, then it will return the error code and error description with status code.
7. If validation is Successful, then the User details is stored in the database and success code is sent.

**11.2 Crop Listing : -**

1. The Farmer will Login using his User name and password.
2. The Farmer can post all the available crops with their name and their availability, Quantity, Price for Sale.
3. After posting all the details regarding the crops the farmer will be waiting for the dealers.

**11.3 Crops on Sale : -**

1. The Dealer login with their username and password.
2. The sees the all the crops that are listed by the various farmers.
3. The dealer search the crops that they are interested to buy that are posted by farmers.
4. The dealer will select and click on proceed button.

**11.3 Payment : -**

1. After the dealer proceeded, the dealer can see the farmer location and he can go and check the quality and weighs the crop.
2. If the Dealer is Satisfied with the quality and quantity of the crop then he will login using his credentials and proceed the payment.
3. After successful payment an invoice is generated and it will be sent to both the user like Farmers and Dealers.

**11.4 Admin Module : -**

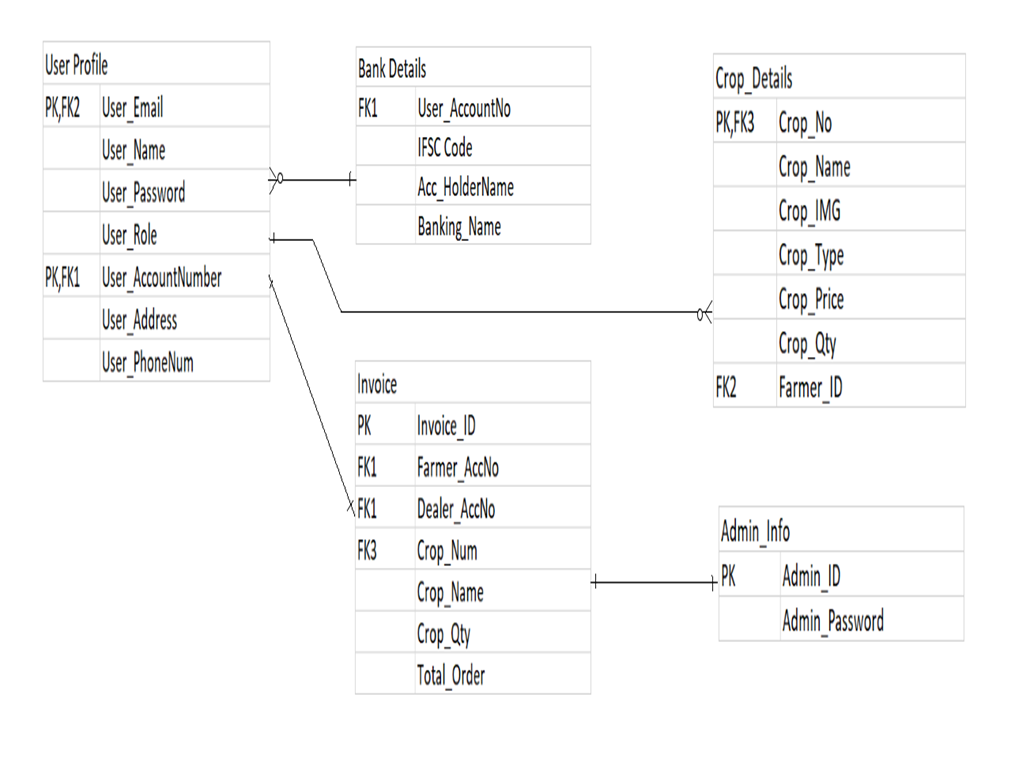
1. Admin will login using with his Username and Password.
2. Admin can able to see the Active /Inactive users.
3. The Admin can get a Dealer’s Report when he clicks to generate a particular dealer report.

**12.0 Classes/Functions :-**

|  |  |  |
| --- | --- | --- |
| S.NO | Classes | Description |
| **1** | User\_Profile.cs | Model holds the user details |
| **2** | RegisterUserController.cs | It Contains the core business logic for the registration of Farmers. Which calls the registerUserDA class to create the farmer in Database. |
| **3** | RegisterUserDA.cs | This class deals with the data accessibility for Farmer registration. |
| **4** | CropListDA.cs | This class deals with data accessibility for crop list. |
| **5** | CropList.cs | It contains the core business logic for adding the crops of farmers which calls the CropListDA to add to the Database. |
| **6** | ActiveUser.cs | This class is for the admin which makes a user active. |
| **7** | InactiveUser.cs | This class is for the admin which blocks a user. |
| **8** | UpdateUser.cs | This class is for the admin to update the user details. |
| **9** | DealerReportGeneration.cs | This is class for the admin to generate a particular dealer’s report. |
| **10** | CropOnSale.cs | This class contains the data and the logics used when the crops on sale. |

NOTE : There are many other classes but the most important classes are mentioned here.

**13.0 Data Model :-**



**14.0 API Canvas :-**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Verb** | **Description** |
| **1** | **post** | **Farmer Registartion** |
| **2** | **get** | **To get crop list** |
| **3** | **post** | **To list crops for sale** |
| **4** | **put** | **To update farmer details** |
| **5** | **get** | **To get list of crops put on sale** |
| **6** | **delete** | **to remove the farmer** |
| **7** | **post** | **Dealer registration** |
| **8** | **delete** | **to remove the dealer** |
| **9** | **post** | **to make the payment for crops** |
| **10** | **get** | **to generate invoice** |

**15.0 HTTPS Codes : -**

|  |
| --- |
| 201: Farmer registered and Dealer registered. |
| 200: request succeeded. |
| 400: invalid inputs |
| 404: Data not found |
| 502: Bad gateway |