**Crop Deal**

**POC**  
**Low Level Design (LLD)**



Date: 09/07/2022

Current Document Version: [*1.0*]

DOCUMENT APPROVAL

**Approvers of this document**

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

**Document Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version #** | **Author** | **Date** | **Description** |
| 1.0 | DHARHSAN KANNA | 09/07/2022 | Document Creation |
| 1.1 | DHARHSAN KANNA | 18/07/2022 | User and Developer Requirements Added |
|  |  |  |  |
|  |  |  |  |

Contents

[1.0 Document Purpose 5](#_Toc109286148)

[2.0 Intended Audience 5](#_Toc109286149)

[3.0 Project Background, Objective(s) 5](#_Toc109286150)

[3.1 Project Background 5](#_Toc109286151)

[3.2 Project Objective 5](#_Toc109286152)

[4.0 Design Pattern 5](#_Toc109286153)

[5.0 Solution Diagram 6](#_Toc109286154)

[6.0 Architecture Diagram 6](#_Toc109286156)

[7.0 Flow Diagram 7](#_Toc109286157)

[8.0 Use case Diagram 8](#_Toc109286158)

[9.0 Class Diagram 9](#_Toc109286159)

[10.0 E-R Diagram 10](#_Toc109286160)

[11.0 User Requirements 11](#_Toc109286161)

[11.1 Hardware 11](#_Toc109286162)

[11.2 Software 11](#_Toc109286168)

[12.0 Developer Requirements 11](#_Toc109286170)

[12.1 Hardware 11](#_Toc109286171)

[12.3 Technology 12](#_Toc109286181)

[13.0 Solution Steps 12](#_Toc109286185)

[14.0 Classes/function 14](#_Toc109286186)

[15.0 Data Model 15](#_Toc109286187)

[16.0 API Canvas 16](#_Toc109286188)

[17.0 HTTP Status Code 16](#_Toc109286189)

# Document Purpose

This document describes the solution architecture for Crop deal

# Intended Audience

This document is intended as a reference for the following roles and stakeholders who are interested in the Crop Deal technical architecture.

|  |  |
| --- | --- |
| Role | Nature of Engagement in WB Classics Portal Technical Architecture |
| Product Owners/SME | Key stakeholder to ensure that the architecture is aligned with business goals. |
| Business Analysts | Business analysts are one of the stakeholders who are informed with the key architectural decisions. |
| Enterprise Architects | Platform Architecture is aligned to business goals and architecture, architectural guidelines. |
| Solution Architects | To ensure solution design and architecture is aligned to business requirements, architectural guidelines. |
| Developers | Use Technical Architecture Document as the guiding document for detail design and implantation approach to align with Crop Deal Micro service |

# Project Background, Objective(s)

## Project Background

Crop Deal micro service is implemented to make farmer’s life easier by advertising their crops and sell them directly to the dealer, as it neglects all

The commissions’ farmers pays to sell their crops.

## Project Objective

The main objective of Crop deal is to act as bridge between farmer and the dealer by registering famer and dealer and farmer will list their crops and the dealer will able to make payment directly to farmer’s account.

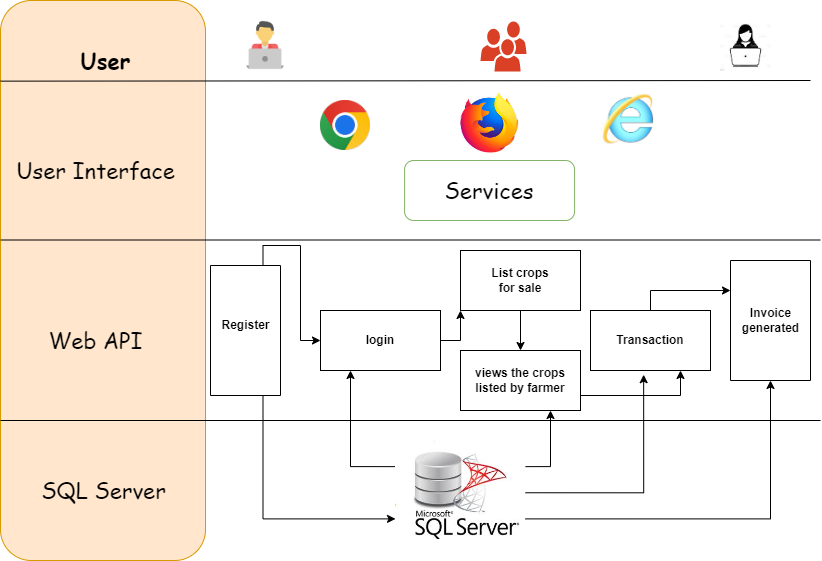
# Design Pattern

|  |  |  |
| --- | --- | --- |
| # | Name | Description |
| 1 | ANGULAR | Angular is used for the frontend and designing of the application |
| 2 | API | Api is used to process all the http  Requests. |
| 3 | SQL SERVER | SQL server is used to store data to the database |

# Solution Diagram

# 

# Architecture Diagram



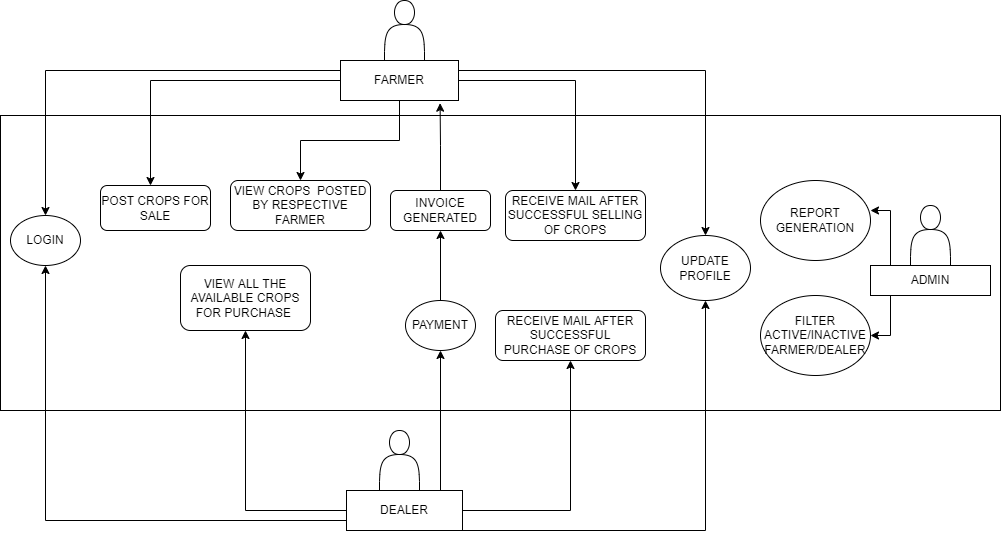
# Flow Diagram

# 

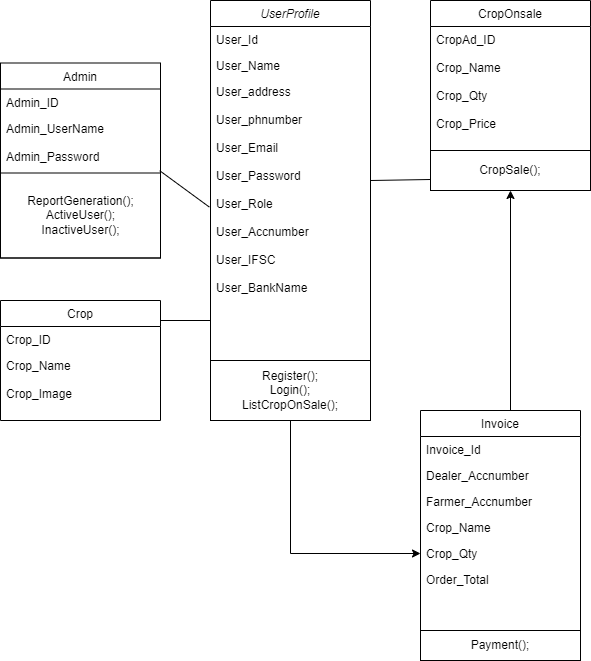
REPORT

GENERATION

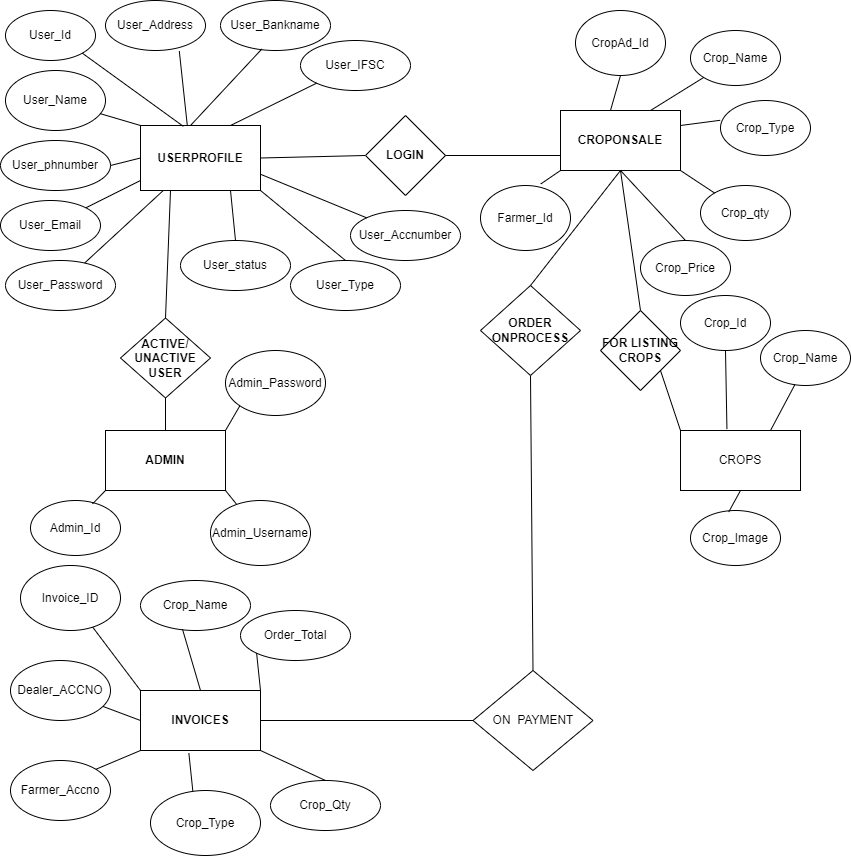
# 8.0 Use case Diagram



# 9.0 Class Diagram



# 10.0 E-R Diagram



# 11.0 User Requirements

# 11.1 Hardware

# • Processor: Minimum 1.8 GHz. Recommended 2GHz or more.

# • Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

# • Hard Drive: Minimum 100 GB; Recommended 500GB or more.

# • Memory (RAM): Minimum 4 GB; Recommended 8 GB or above.

# • OS: Windows.

# 11.2 Software

# • Any Latest Browsers.

# 12.0 Developer Requirements

# 12.1 Hardware

# • Processor: Minimum 1.8 GHz. Recommended 2GHz or more.

# • Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

# • Hard Drive: Minimum 100 GB; Recommended 500GB or more.

# • Memory (RAM): Minimum 4 GB; Recommended 8 GB or above.

# • OS: Windows.

# 12.2 Software

# • Visual studio 2022.

# •Sql server management studio (ssms).

# • Node, Angular.

# 12.3 Technology

# • ASP.NET for backend

# •SQL for the database operations

# • Angular for frontend.

# 13.0 Solution Steps

**13.1 User Registration**

1. The user will select their role as a farmer or dealer.
2. The user will enter the name, phone Number, Address and their bank account details during registration.
3. The input validation will be done
4. If validation fails, then it will return the error code and error description. with status code
5. If validation is successful, then the User details is stored in the database and success code is sent.

**13.2 Crop Listing**

1. The Farmer will login using his username and password.
2. The Farmer will be posting all the available crops with the Name, quantity, price for sale.
3. After posting all the details regarding the crops the farmer will be waiting for the dealers.

**13.3 Crops on Sale**

1. The dealer logins with their username and password.
2. The dealer sees the all-listed crops that was listed by the various farmers.
3. When the dealer interested in purchasing the crops which was listed by the farmer.
4. The Dealer will click proceed button.

**13.4 Payment**

1. Then Dealer will reach the farmer location and checks the quality and weighs the crops.
2. If the dealer is satisfied with the crop quality and the quantity.
3. He will again login using his login credentials and proceed with the payment.
4. After successful payment an Invoice will be generated and it will be sent to both the farmer and Dealer.

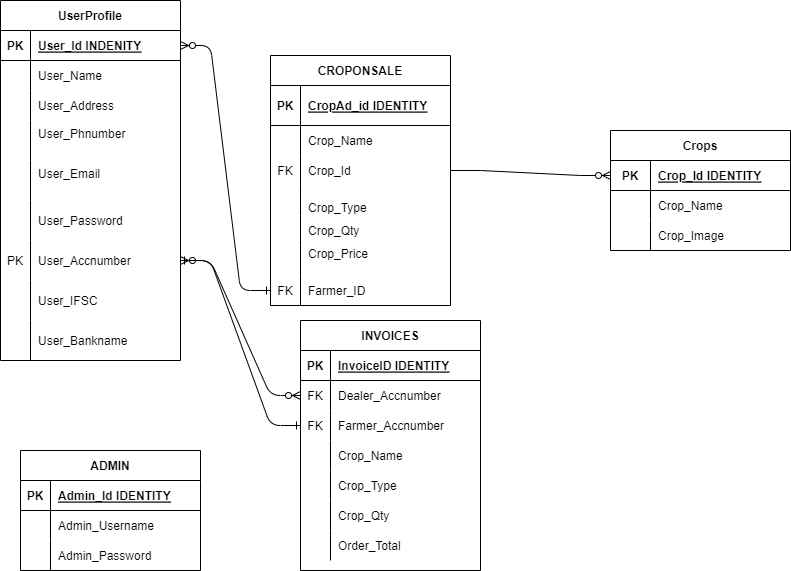
**13.5 Admin Module**

1. Admin will login with his Username and Password.
2. Admin can able to filter active/inactive farmers and dealers.
3. The admin can get a Dealer’s report when he clicks to generate a particular dealer report.
   1. **Regional Language Support**
      * 1. As all the famer are not aware of English We are planning to make this application with regional language support in the nearest future.

# 14.0 Classes/function

|  |  |  |
| --- | --- | --- |
| **#** | **Class** | **Description** |
| 1 | Userprofile.cs | Model holds the user schema 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 the 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 | DealerReportGenration.cs | This is class is 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 |
| 11 | Invoice.cs | This class is for the invoice generation when the payment  Is completed successfully |

# 15.0 Data Model

****

# 16.0 API Canvas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| POC | Path | Verb | API Description | Role | Auth |
| Crop-Deal | /farmers | POST | Farmer registration | No | True |
| Crop-Deal | /crops | GET | To get Crop list | Farmer | True |
| Crop-Deal | /sellcrops | POST | To list crops for sale | Farmer | True |
| Crop-Deal | /farmer/id | PUT | To update Farmer details | Farmer | True |
| Crop-Deal | /farmercrops/id | GET | To get list of crops put on sale | Dealer | True |
| Crop-Deal | /farmer/id | DELETE | To remove the Farmer | Admin | True |
| Crop-Deal | /dealer | POST | Dealer registration | NO | True |
| Crop-Deal | /dealer/id | DELETE | To remove the Dealer | Admin | True |
| Crop-Deal | /buycrops | POST | To make payment for the crops | Dealer | True |
| Crop-Deal | /Invoice | GET | To generate invoice | Dealer | True |

# 17.0 HTTP Status Code

201 – Farmer registered, dealer registered.

200 - Request succeeded

400 – Inputs are invalid

404 – Data not found

502 – Bad gateway

**15.0 Unit Testing**

Project Name Crop Deal Micro service

Created by DHARSHAN KANNA

Date of Creation

Date of review

**For Registration OF Farmers**

**16.0 Request**

Register Farmer

/farmer

{

“Farmer\_name”: “Dhanush”,

“Farmer\_Address”: “Chennai TamilNadu”,

“Farmer\_phnumber” :” 0123456789”,

“Farmer\_Password”:”dhanush1”,

“Farmer\_Accnumber”:”75231458964563456”,

“Farmer\_IFSC”:”SNINB021”,

“Farmer\_bankname”:”SBI”,

}

Update Farmer

/farmer/id

{

“Farmer\_name”: “Dhanush”,

“Farmer\_Address”: “Chennai TamilNadu”,

“Farmer\_phnumber” :” 0123456789”,

“Farmer\_Password”:”dhanush1”,

“Farmer\_Accnumber”:”75231458964563456”,

“Farmer\_IFSC”:”SNINB021”,

“Farmer\_bankname”:”SBI”,

}

Register Dealer

/dealer

{

“Dealer\_name”: “Kiran”,

“Dealer\_Address”: “Chennai TamilNadu”,

“Dealer\_phnumber” :” 0123456789”,

“Dealer\_Password”:”kiran1”,

“Dealer\_Accnumber”:”55423964563456”,

“Dealer\_IFSC”:”SNINB021”,

“Dealer\_bankname”:”SBI”,

}

Update Dealer

/dealer/id

{

“Dealer\_name”: “Kiran”,

“Dealer\_Address”: “Chennai TamilNadu”,

“Dealer\_phnumber” :” 0123456789”,

“Dealer\_Password”:”kiran1”,

“Dealer\_Accnumber”:”5542638964563456”,

“Dealer\_IFSC”:”SNINB021”,

“Dealer\_bankname”:”SBI”,

}

**17.0 Response:**

Register Farmer

If valid details

{

"Message": “Farmer registered successfully"

}

Status code: 201

If invalid details

{

“Message”: “Inputs are not valid”

}, status code: 400

If server encounters unexpected error

{

“message”:” Internal server error contact admin”

}, status code: 500

Update Farmer

If valid ID

{

"Message": “Farmer updated successfully"

}

Status code: 200

If invalid ID

{

“Message”: “Farmer not found”

}, status code: 404

If server encounters unexpected error

{

“message”:” Internal server error contact admin”

}, status code: 500

Register Dealer

If valid details

{

"Message": “Dealer registered successfully"

}

Status code: 201

If invalid details

{

“Message”: “Inputs are not valid”

}, status code: 400

If server encounters unexpected error

{

“message”:” Internal server error contact admin”

}, status code: 500

Update Dealer

If valid ID

{

"Message": “Dealer updated successfully"

}

Status code: 200

If invalid ID

{

“Message”: “Dealer not found”

}, status code: 404

If server encounters unexpected error

{

“Message”:” Internal server error contact admin”

}, status code: 500