SOFTWARE ENGINEERING & OBJECT ORIENTED MODELING DESIGN

FARM TO TABLE

STAGE 1

Problem Statement

Millions of farmers with small scattered individual production do not have enough network or platform to negotiate for better prices, nor are they able to retain their own production to increase time utility. They are generally dominated by the local traders which leads to unnecessary intermediaries between the producer and the consumer. Also, consumers are becoming increasingly conscious about the food they consume, from where it comes to how it is grown and processed.

Our solution i.e the farm-to-table approach allows the farmers to deliver food directly to customers so that the customers are assured of quality and farmers can earn more which will help boost the local economy. This ensures the local food producer a lot of value when it comes to being able to get paid directly.

Analysis

Actors: User, Farmer, Delivery Agent, Admin

Current situation: Farmers are forced to sell their products to wholesalers and distributors even in local areas at higher rates.

Benefits: Using this model allows the farmer to sell to local members for a price that is higher than the wholesale price but lower than the retail price for the customer. Customers are also assured of quality.

Requirements Elicitation

1. **Functional Requirements**

- **Customer Account -** The customer has an account with which he can contact and order the required items as per his needs.
- **Farmer Account -** The farmer has an account with which he can receive orders from customers and provide them with the items
- **Admin** - Controls and manages the users, items, various categories in the system, performs updation, deletion tasks.
- **Database -** The database to store all information on users, farmers, items, account info etc.
- **Delivery Time -** An estimated date and time should be provided for the delivery of the ordered items so that the customer knows when his order is arriving.
- **Payment Method -** Payment Method is the mode or way through which the customer and farmer agree upon for the payment to take place.
- **Refund -** A trusted method for a refund should be available in case of any cancellation or non-delivery of the order.

2. Non-Functional Requirements

- **Farmer Distance from Customer -** The distance of the farmer from the customer location so that localization is ensured.
- **Usability -** The user should be able to learn to operate, prepare inputs for, and interpret the outputs of a system easily.
- **Performance -** The system must be interactive and the delays involved must be less. So in every action response of the system, there are no immediate delays. The server must be robust with respect to any possible occurring failures.
- **Availability -** The system must be available at all times, except when upgrading or restarting the backend database. In case of no network connection, the system shall enable the admin to add, update or delete item entities to the server once the network connection becomes available again.
- **Safety -** Information transmission should be securely transmitted to the server without any changes in the information, also money transactions should be safe.
- **Reliability -** The system should be able to perform its required functions under stated conditions for a specified period of time. Also, it should have the ability to detect specified faults or to withstand specified security attacks.
- **Database Security -** Sensitive data (e.g. password) will be stored in an encrypted manner. A database backup will be taken in a regular interval and stored in a secured place. The database will be secured with a strong password.
- **Localization -** The system must be available in local languages understood by both the farmers and consumers.
- **Membership discounts -** Including the membership plan on a monthly basis allows us to offer certain discounts to the user as we are assured a minimum income from them and it encourages customers to reorder from the farmer as well.