

PHASE 3: PROJECT DESIGN PHASE

3.1 System Architecture

- The *FoodConnect* system has been designed following a modular and scalable architecture, ensuring flexibility, efficiency, and ease of maintenance. The architecture consists of four key layers: frontend, backend, database, and visualization.
- The **frontend layer** represents the user interface built using Salesforce Lightning components. It provides an intuitive and interactive experience for users such as administrators, NGOs, and volunteers. This interface enables users to manage data records, execute tasks, and visualize system outputs easily through a browser or Salesforce mobile app.
- The **backend layer** includes the custom logic and automation processes built using Salesforce's development tools such as Apex Triggers and Flows. This layer handles all core operations, including record manipulation, workflow execution, and business logic implementation. It ensures that system processes are automated and operate seamlessly behind the scenes.
- The **database layer** manages the data structure of the application. It includes various custom objects and their interrelationships, ensuring smooth data flow between modules. Data integrity is maintained through master-detail and lookup relationships that link different objects logically.
- The **visualization layer** provides analytical insights through Reports and Dashboards. This layer transforms raw data into meaningful metrics and visual representations, enabling administrators and NGOs to monitor performance, track donations, and evaluate volunteer efficiency in real-time.

3.2 Object Model

- The design of the object model forms the core of the FoodConnect application. Each object represents a key entity within the food collection and distribution process.
- The **Venue object** represents food donors such as restaurants, hotels, event halls, and organizations that contribute leftover food. It stores critical information including venue name, contact details, location, and food availability.
- The **Drop-Off Point object** defines the locations where food is delivered or distributed. These may include shelters, community kitchens, or NGO-operated centers. The object helps track where each donation is received.
- The **Volunteer object** stores the details of individuals who participate in the food distribution process. It includes attributes like name, contact information, gender, and assigned tasks.

- The **Task object** represents the operational activities involved in delivering food from a venue to a drop-off point. It contains information such as delivery date, distance, food type, and number of people served.
- Finally, the **Execution Details object** is used to link volunteers and tasks. It keeps track of who completed which task, along with timing and performance details. This object ensures transparency and accountability in volunteer operations.

3.3 Relationships

- Establishing accurate relationships between objects is crucial for maintaining data integrity and logical flow within the system.
- The **Venue** and **Drop-Off Point** objects are connected through a **Lookup relationship**, allowing each drop-off point to be associated with a specific food donor.

3.4 Flow Design

- Automation is one of the key features of Salesforce, and in this project, a **Screen Flow** titled “*Venue Form*” has been designed to automate record creation. The flow captures inputs such as Venue Name, Email, Phone Number, Location, Latitude, and Longitude. Once the details are entered by the user, the flow automatically creates a new record in the Venue object.
- This automation eliminates the need for manual data entry and reduces errors. It also improves user experience by providing a guided interface for data submission

3.5 Trigger Logic

An **Apex Trigger** named *DropOffTrigger* has been implemented to automate the calculation of distance between a venue and its corresponding drop-off point. The trigger executes before the insertion of a new record in the Drop-Off Point object. The logic is written as follows:

```
trigger DropOffTrigger on Drop_Off_point__c (before insert) {
    for (Drop_Off_point__c Drop : Trigger.new) {
        Drop.Distance__c = Drop.distance_calculation__c;
    }
}
```

- This trigger ensures that whenever a new drop-off point record is created, the system automatically calculates and assigns the distance value.
- This automation not only improves efficiency but also ensures data accuracy for reporting and logistical analysis.

3.6 Interface Design

- The application includes several key tabs such as Venue, Drop-Off Point, Task, Volunteer, Execution Details, Reports, and Dashboards. Each tab serves a specific purpose and helps users quickly access and update relevant data.
- The **Home Page** of the app includes two major components: the **Venue Form Flow** and the **Food Distribution Dashboard**. The Venue Form allows users to create new venue records directly from the homepage, while the dashboard provides an at-a-glance overview of key statistics such as the number of tasks completed, volunteers involved, and the total number of people served.