**PROJECT DESIGN PHASE**

|  |  |
| --- | --- |
| Date | 29-05-2025 |
| Team ID | LTVIP2025TMID28821 |
| Project Name | TO SUPPLY LEFTOVER FOOD TO POOR |
| Maximum Marks | 4 Marks |

**SOLUTION ARCHITECTURE**

**1. Introduction**

The success of any technology-driven platform lies in its architectural design. The project *“To Supply Leftover Food to the Poor”* introduces a smart, cloud-based system—**NourishBridge**—that seamlessly connects food donors with NGOs, ensuring the timely redistribution of surplus food. This section outlines the layered architecture of the solution, detailing each component, integration, and data flow that empowers the system to function effectively, securely, and at scale.

**2. Architectural Goals**

The core goals that shaped the system architecture include:

* **Scalability**: Support a growing number of users and donations across regions.
* **Availability**: Ensure real-time access and uptime for all stakeholders.
* **Security**: Protect sensitive user and location data.
* **Modularity**: Allow easy maintenance, upgrades, and feature expansions.
* **Automation**: Minimize manual steps through intelligent process flows.

**3. Overview of the Architecture**

The architecture consists of the following layers:

1. **Presentation Layer (User Interfaces)**
2. **Business Logic Layer (Automation & Processing)**
3. **Data Layer (Storage & Structure)**
4. **Integration Layer (External APIs)**
5. **Analytics & Monitoring**

Each layer is described in the sections below, illustrating how they work together to deliver seamless user experiences and backend coordination.

**4. Presentation Layer**

This is the user-facing part of the system, where different roles interact with the platform through responsive portals.

**4.1 Donor Portal**

* Developed using Salesforce Experience Cloud and Lightning Web Components (LWC).
* Allows users to post donations, track status, and view impact dashboards.
* Responsive UI ensures accessibility from desktops, tablets, and mobile phones.

**4.2 NGO Portal**

* Designed with more advanced features for accepting donations, assigning volunteers, and managing distribution workflows.
* Visual maps and dashboards help NGOs monitor their outreach.

**4.3 Admin Interface**

* Accessible via Salesforce Lightning Experience.
* Enables administrators to verify users, manage configurations, and access system-wide analytics.

**5. Business Logic Layer**

At the heart of NourishBridge is a robust set of automations and rules that govern the platform's behavior.

**5.1 Salesforce Flow**

* Automates key processes like:
* Matching donations with NGOs.
* Sending notifications (email/SMS).
* Updating donation statuses through each lifecycle stage.

**5.2 Apex Code**

* Applied for business-critical logic not achievable through declarative tools.
* Examples include:
* Preventing multiple NGOs from claiming the same donation.
* Custom validation rules for donation windows and pickup timelines.

**5.3 Scheduled Jobs**

* Used to clean up expired donations and trigger reminder alerts for pending pickups.

**6. Data Layer**

This layer handles all data storage, validation, and relationships across platform entities.

**6.1 Data Model**

* **Standard Objects**:
  + Account: Represents organizations (Donors and NGOs).
  + Contact: Represents individual users (managers, coordinators).
* **Custom Objects**:
* Donation\_\_c: Core object capturing food type, quantity, pickup info, and status.
* NGO\_Profile\_\_c: Stores NGO service radius, capacity, and verification.
* Volunteer\_Task\_\_c: Logs assignments and delivery updates.
* Feedback\_\_c: Tracks donor and recipient satisfaction for quality assurance.

**6.2 Data Validation**

* Field-level validations ensure all entries meet business logic.
* Lookup relationships enforce data integrity between connected objects.

**7. Integration Layer**

This layer facilitates communication with external services to enhance real-time functionality.

**7.1 Twilio Integration**

* Sends SMS alerts to NGOs and volunteers.
* Allows for instant delivery notifications and pickup confirmations.

**7.2 Salesforce Maps (Planned Feature)**

* Will offer visual tracking of available donations, NGO coverage, and volunteer routes.
* Helps optimize logistics and reduce food delivery times.

**7.3 APIs (Future Scope)**

* Open APIs to allow integration with:
* Inventory management tools used by NGOs.
* Government dashboards for public health and food safety compliance.
* Corporate CSR platforms for donation tracking and sponsorships.

**8. Analytics and Reporting**

**8.1 Real-Time Dashboards**

* Each user role has customized dashboards:
* Donors: Meals donated, food saved, location heatmaps.
* NGOs: Beneficiaries reached, types of food received.
* Admins: City-wise donations, system activity, user trends.

**8.2 Monthly Reporting**

* Auto-generated reports for audits, funding proposals, and community updates.
* Downloadable in PDF and Excel formats.

**9. Security Architecture**

* Role-based access control to restrict data visibility.
* Session timeout and login monitoring to prevent misuse.
* Field-level encryption and record-level sharing for data protection.
* Apex triggers prevent unauthorized edits of sensitive records.

**10. Scalability & Maintainability**

* Cloud-native deployment allows the platform to expand across new cities or countries.
* Metadata-driven design using Salesforce ensures that new workflows, fields, or pages can be added without code.
* Modular object structure allows for easy plug-ins and service expansion (e.g., donations of clothing or medicine in future).

**11. Conclusion**

The solution architecture for *“To Supply Leftover Food to the Poor”* is designed with both **heart** and **engineering precision**. It addresses real-world needs by connecting technology with compassion—leveraging cloud scalability, automation, and transparent data handling. Every component, from the donor form to the impact dashboard, is built to ensure that no act of generosity is wasted and every surplus meal finds a meaningful destination.