**Phase 1: Problem Understanding & Industry Analysis**

**Project Title: Smart Farm-to-Market Management System**

**Industry: Agriculture**

**Prepared By: [Hariom Balang]**

**1. Introduction**

The agricultural sector, particularly in rural areas, faces significant challenges in crop management, market access, subsidy utilization, pest control, transportation, and financial tracking. These processes are largely dependent on manual methods that result in inefficiencies, data inaccuracies, and communication gaps among stakeholders. The objective of this phase is to understand the underlying problems and define requirements for building a Salesforce-based solution that automates workflows, improves visibility, and enhances productivity — all without relying on external paid services.

This document outlines the activities performed in Phase 1, including requirement gathering, stakeholder analysis, business process mapping, industry-specific use case analysis, and exploration of Salesforce’s AppExchange offerings.

**2. Requirement Gathering**

The first step involved identifying the functional and non-functional requirements necessary to address the core problems in farm-to-market operations.

**Functional Requirements**

* **Crop Data Management**: Farmers should be able to enter crop details such as type, location, expected yield, and planting schedule.
* **Market Price Tracking**: Administrators should update market prices across regions, and farmers should receive notifications on price fluctuations.
* **Pest & Disease Alerts**: The system must allow for quick reporting of pest outbreaks and notify affected farmers instantly.
* **Subsidy Application Management**: Farmers should apply for subsidies via structured forms, with automated eligibility checks and approval processes.
* **Transport Scheduling**: Farmers should request transport services, and managers should assign agents based on availability and location.
* **Invoicing & Payment Tracking**: The system should generate invoices and track payments in real-time.
* **Reporting & Dashboards**: Users should access reports on crop production, pricing trends, subsidy utilization, and transport logistics.

**Non-Functional Requirements**

* **Usability**: The platform must be user-friendly, especially for farmers with limited technical experience.
* **Security**: Role-based permissions, data access controls, and privacy settings must be implemented.
* **Scalability**: The solution should support increased data volumes as more users and regions are added.
* **Performance**: The system should operate seamlessly, even with large datasets and concurrent users.
* **Accessibility**: Interfaces should be compatible with both desktop and mobile devices to accommodate users in remote areas.

**3. Stakeholder Analysis**

* **Farmers**
  + Primary users of the system
  + Responsibilities: Enter crop data, apply for subsidies, track crop status
  + Expectations: Simple interface, timely alerts, financial tracking
* **Co-operative Managers**
  + Supervisory users
  + Responsibilities: Manage market pricing, approve subsidies, assign transport
  + Expectations: Automated workflows, reporting dashboards, real-time monitoring
* **Transport Agents**
  + Service providers for logistics
  + Responsibilities: Handle scheduling and delivery of crops
  + Expectations: Efficient task allocation, clear tracking of deliveries
* **Government Bodies**
  + Policy administrators
  + Responsibilities: Approve subsidies, monitor compliance
  + Expectations: Structured data, accurate eligibility reporting, audit-ready information
* **Salesforce Admin**
  + System operator
  + Responsibilities: Configure platform, manage users, roles, and access
  + Expectations: Reliable, secure, and scalable environment for all users

**4. Business Process Mapping**

A detailed workflow was mapped to understand how data and tasks flow across stakeholders.

**Crop Planning Workflow**

* Farmers enter crop details → Co-operative manager reviews → Data stored for reporting.

**Market Price Update**

* Admin updates price trends → Notifications sent → Dashboard updated.

**Pest Alert Workflow**

* Pest outbreaks are reported → System identifies affected regions → Farmers are notified → Preventive actions advised.

**Subsidy Application Workflow**

* Farmer submits subsidy request → System validates eligibility → Manager reviews and approves → Farmer is notified.

**Transport Scheduling Workflow**

* Farmer requests transport → Manager assigns available agent → Delivery scheduled and tracked.

**Billing & Payment Workflow**

* Post-harvest, invoices are generated → Payments tracked → Reports updated for transparency.

These workflows help in identifying automation points that reduce manual effort and errors.

**5. Industry-specific Use Case Analysis**

A study of agricultural challenges and workflows highlighted the following:

**Challenges**

* Lack of structured data entry and storage.
* Inconsistent access to market information.
* Delayed communication between stakeholders.
* Manual subsidy applications prone to errors.
* Inefficient transport coordination leading to crop wastage.

**Best Practices**

* Centralized data systems that streamline crop tracking.
* Automated workflows to reduce manual intervention.
* Real-time alerts for pest management.
* Interactive dashboards to guide decision-making.

**Salesforce Alignment**

Salesforce’s standard functionalities like custom objects, flows, approval processes, and reporting tools provide a cost-effective solution to these challenges without relying on external paid APIs.

**6. AppExchange Exploration**

To ensure that the system leverages Salesforce’s ecosystem efficiently, an exploration of AppExchange was conducted.

**Findings**

* Several agriculture-related apps exist but mostly require paid subscriptions or external integrations.
* Free apps offered limited functionality or were not directly applicable to farm workflows.
* Salesforce’s native tools — custom objects, flows, validation rules, and dashboards — are sufficient to meet the project requirements.

**Phase 2: Org Setup & Configuration**

**1. Introduction**

In Phase 2, the Salesforce environment is configured to support the application’s business processes and workflows. This phase ensures that organizational settings, user permissions, roles, and data sharing rules are properly defined before implementation begins. The configuration focuses on scalability, data security, and user accessibility while using Salesforce’s free tools wherever possible.

This phase lays the foundation for building a robust and secure application tailored to the agricultural industry’s specific needs.

**2. Salesforce Editions**

For this project, the **Developer Edition Org** has been selected, as it provides access to all Salesforce features without subscription costs.

* **Key Features Available:**
  + Custom Objects, Fields, and Relationships.
  + Flows, Workflows, and Approval Processes.
  + Apex programming and Lightning Web Components (LWC).
  + Reports, Dashboards, and Analytics tools.
* **Limitations:**
  + Sandbox environment is not available in Developer Edition, so testing is performed in a separate environment or within the development org.

This edition ensures cost-efficiency while offering a fully functional Salesforce environment for development and testing.

**3. Company Profile Setup**

The organization’s profile was configured to align with regional requirements and agricultural workflows.

* **Steps Taken:**
  1. Accessed Setup → Company Information.
  2. Updated the organization’s name to “Smart Farm-to-Market Management System.”
  3. Set the default currency to **INR** to reflect local transactions.
  4. Set Locale to **India** for date formats and regional settings.
  5. Configured Timezone to **IST** (Indian Standard Time).
* **Purpose:**  
  These settings ensure that reports, dates, and financial fields align with real-world requirements, improving user experience and reducing confusion.

**4. Business Hours & Holidays**

Defining business hours and holidays is essential for scheduling transport services, subsidy approvals, and task assignments.

* **Business Hours Configured:**
  + Monday to Saturday, 9:00 AM to 6:00 PM.
* **Holidays Added:**
  + Republic Day
  + Independence Day
  + Diwali
  + Local harvest festival days
* **Use Case:**  
  Business hours ensure that automated processes such as task scheduling and approval routing occur within working hours. Holidays are considered to avoid assigning tasks when offices are closed.

**5. Fiscal Year Settings**

The fiscal year was set to align with agricultural cycles and reporting requirements.

* **Settings Applied:**
  + Fiscal year starts from **April 1st**, ending on **March 31st**, to correspond with the seasonal crop calendar.
* **Purpose:**  
  Aligning the fiscal year helps produce accurate seasonal reports and financial tracking for subsidies and crop payments.

**6. User Setup & Licenses**

Users were created to represent stakeholders in the system.

**Test Users:**

1. **Admin User** – Full access for system configuration and management.
2. **Co-operative Manager** – Oversees pricing, subsidy approvals, and transport coordination.
3. **Farmer User** – Inputs crop data, applies for subsidies, and requests transport.
4. **Transport Agent** – Manages transport logistics and delivery schedules.

* **Licenses Assigned:**
  + Standard Salesforce user licenses were used for all roles.
* **Purpose:**  
  Ensures role-based data access while allowing each user group to perform their respective tasks effectively.

**7. Profiles**

Profiles control baseline permissions for users based on their role.

**Profile Definitions:**

* **Admin Profile:**
  + Full access to all objects, fields, and reports.
  + Manage users and system settings.
* **Co-operative Manager Profile:**
  + Create, view, edit crop records, market prices, and subsidy applications.
  + Access approval workflows and reports.
* **Farmer Profile:**
  + Create and view crop entries.
  + Submit subsidy applications and transport requests.
  + View assigned invoices and notifications.
* **Transport Agent Profile:**
  + View assigned transport requests.
  + Update delivery status.
* **Use Case:**  
  Profiles ensure data privacy and structured access, enabling users to see only what’s relevant to their responsibilities.

**8. Roles**

A role hierarchy was designed to streamline data access based on organizational structure.

**Role Hierarchy:**

* **Admin → Co-operative Manager → Farmer → Transport Agent**
* **Benefits:**
  + Managers can view records submitted by their subordinates.
  + Farmers see only their own data.
  + Transport agents access requests assigned to them.

This hierarchy mirrors real-world communication flows and simplifies reporting.

**9. Permission Sets**

Permission sets were created to extend access for specific tasks without modifying existing profiles.

* Example Use Cases:
  + A farmer accessing new reports.
  + A manager using advanced dashboards during seasonal planning.

Permission sets ensure flexibility while maintaining security controls.

**10. Organization-Wide Defaults (OWD)**

The OWD was set to control data visibility at a global level.

**Configuration:**

* Crop Records → **Private**
* Market Prices → **Read Only**
* Subsidy Applications → **Private**
* Transport Requests → **Controlled by Parent**
* Invoices → **Private**

**Purpose:**  
These settings ensure that sensitive data such as crop details and subsidy requests are visible only to authorized users, enhancing privacy and compliance.

**11. Sharing Rules**

Sharing rules were implemented to allow data access based on user roles and business requirements.

**Example Rules:**

* Crop records shared with co-operative managers based on territory.
* Subsidy requests shared with approval managers.
* Transport requests visible to assigned transport agents.

**Purpose:**  
Facilitates data sharing where collaboration is required while ensuring restricted access elsewhere.

**12. Login Access Policies**

Default login policies were applied to ensure security while allowing authorized access.

* Two-factor authentication can be enforced.
* IP restrictions may be configured later based on regional needs.
* Login hours were defined to prevent access outside business hours if required.

**13. Dev Org Setup**

The project was built in Salesforce’s Developer Edition environment.

* All configurations and development work were performed in this org.
* Version control through local tools like VS Code and SFDX ensured backup and rollback capabilities.

**14. Sandbox Usage**

Since Developer Edition does not provide sandboxes, testing workflows, triggers, and reports were conducted in a dedicated testing environment.

* All changes were validated before deployment.
* Realistic sample data was created to simulate production scenarios.

**15. Deployment Basics**

For moving changes between environments and ensuring data integrity, the following tools were used:

* **Change Sets** – For transferring configuration settings.
* **Data Loader** – For importing bulk data such as farmers, crop records, and pricing information.
* **VS Code & Salesforce CLI (SFDX)** – For version-controlled deployment and automated script execution.

**Best Practices Followed:**

* Validate in sandbox/test environments before deployment.
* Create backup snapshots prior to applying updates.
* Document configurations for audit purposes.

**Phase 3: Data Modeling & Relationships**

Phase 3 of the **Smart Farm-to-Market Management System** focuses on designing the **data structure** and relationships in Salesforce. Effective data modeling ensures accurate storage, integrity, and accessibility of information for farmers, managers, and administrators.

**1. Standard & Custom Objects**

The system leverages **standard Salesforce objects** like User, Account, and Contact to manage users, farmer groups, and individual farmers. Standard objects also support reporting and dashboards for analytics.

In addition, **custom objects** were created to capture project-specific data:

* **Crop**: Records details for each crop planted, including crop type, season, expected yield, and farm location.
* **Market Price**: Stores the prices of crops across different markets and updates over time.
* **Pest Alert**: Captures notifications about pest or disease outbreaks affecting crops.
* **Subsidy Application**: Manages farmer applications for government subsidies and tracks approval workflows.
* **Transport Request**: Handles requests for crop delivery and assignment of transport agents.
* **Invoice**: Tracks crop sales, payments, and billing status.
* **Farmer Feedback**: Collects suggestions, complaints, or reports from farmers to improve system operations.

**2. Fields**

Each object contains fields designed for accuracy and usability. Field types include:

* **Text fields** for crop names, market names, and other short text entries.
* **Long Text Area** fields to store detailed descriptions, such as pest alert notes.
* **Number fields** for expected yields, quantities, and pricing calculations.
* **Date and DateTime fields** to track planting, harvest, or transport dates.
* **Picklist fields** for predefined options like crop type or subsidy status, ensuring data consistency.
* **Formula fields** to calculate derived values automatically, such as total invoice amounts.
* **Checkbox fields** to track binary statuses, such as approved or pending.
* **Lookup and Master-Detail fields** to establish relationships between objects, linking related records for workflows and reporting.

**3. Record Types**

Record Types allow different business processes and layouts within the same object. For example:

* The Crop object has record types for **Seasonal Crops** and **Perennial Crops**, providing tailored page layouts and validation rules.
* Subsidy Applications use record types for **State Subsidy** and **Central Subsidy**, enabling distinct approval processes and notifications for each type.

**4. Page Layouts**

Page layouts define which fields, sections, and actions appear to users. Examples include:

* The Crop layout displays crop details, farmer information, and expected yield.
* Subsidy Application layout shows subsidy type, requested amount, attached documents, and approval history.
* Invoice layout presents sold crops, quantities, payment status, and total amounts.

Page layouts ensure users see relevant information efficiently while minimizing clutter.

**5. Compact Layouts**

Compact layouts provide **quick views** for record highlights, especially useful on mobile devices. For instance, the Crop object’s compact layout displays the crop name, season, farmer, and status. This allows users to access critical information without opening the full record.

**6. Schema Builder**

The Schema Builder is a visual tool for designing and viewing the Salesforce data structure. It shows objects, fields, and relationships in a drag-and-drop interface. This tool helps admins and developers maintain a clear overview of the data model, simplifying design, troubleshooting, and communication.

**7. Lookup vs Master-Detail vs Hierarchical Relationships**

Different relationships define how records connect:

* **Lookup Relationship**: A loose link where child records can exist independently. For example, a Crop can be linked to Market Price, but deleting the Market Price does not delete the Crop.
* **Master-Detail Relationship**: A strong dependency where the child inherits the parent’s ownership and sharing settings. For example, a Subsidy Application depends on a Crop record; if the Crop is deleted, the Subsidy Application is also deleted.
* **Hierarchical Relationship**: Used exclusively on the User object to define managerial hierarchies, such as a Manager overseeing a group of farmers.

**8. Junction Objects**

Junction objects support **many-to-many relationships**. For example, crops can be assigned to multiple transport agents, and each transport agent can handle multiple crops. The junction object stores assignment details, such as pickup date and status, enabling flexible logistics management without duplicating records.

**9. External Objects**

External objects allow Salesforce to access data stored outside of the platform, such as government market databases. This ensures users can view and interact with external information in real-time without storing it in Salesforce, maintaining a seamless interface for farmers and managers.