

Salesforce Project Implementation Phases with Concepts (Admin + Developer)

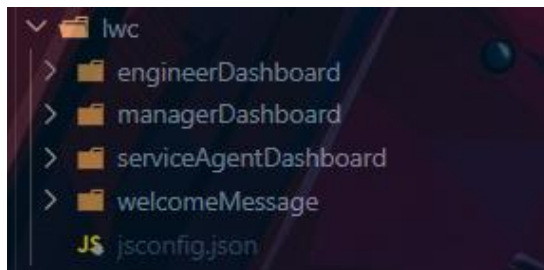
Project Title: Manufacturing After-Sales & Service CRM

Phase 6: User Interface Development

This phase focuses on creating a modern, responsive, and user-friendly interface for the Manufacturing After-Sales & Service CRM. The primary technologies used are Lightning Web Components (LWCs) for building custom user interfaces and the Lightning App Builder for creating and customizing pages.

Lightning Web Components (LWCs)

LWCs are used to build custom, reusable components that provide a rich and interactive user experience.



- **serviceAgentDashboard**
 - **Use Case:** This component serves as the primary interface for service agents. It provides a comprehensive overview of key metrics, recent cases, and customer feedback, enabling agents to efficiently manage their workload and stay informed.
 - **Implementation Details:**
 - Displays key statistics such as total cases, open cases, SLA-breached cases, and average customer rating.
 - Includes tabs to view recent service cases, recent customer feedback, and the status of registered products.
 - Provides quick-action buttons to create a new case, view all cases, and look up products.

```
JS serviceAgentDashboard.js X
force-app > main > default > lwc > serviceAgentDashboard > JS serviceAgentDashboard.js > ServiceAgentDashboard > openCases
1 import { LightningElement, track, wire } from 'lwc';
2 import { NavigationMixin } from 'lightning/navigation';
3 import { ShowToastEvent } from 'lightning/platformShowToastEvent';
4 import getServiceAgentStats from '@salesforce/apex/ServiceAgentController.getServiceAgentStats';
5 import getRecentCases from '@salesforce/apex/ServiceAgentController.getRecentCases';
6 import getRecentFeedback from '@salesforce/apex/ServiceAgentController.getRecentFeedback';
7 import getProductStatus from '@salesforce/apex/ServiceAgentController.getProductStatus';
8
9 export default class ServiceAgentDashboard extends NavigationMixin(LightningElement) {
```

- **managerDashboard**

- **Use Case:** This dashboard is designed for managers to monitor the overall performance of the service team and identify trends. It provides high-level metrics, engineer performance data, and SLA compliance information.
- **Implementation Details:**
 - Displays executive summary cards for total cases, open cases, SLA-breached cases, average customer rating, and active engineers.
 - Features tabs for an overview of case status distribution and top products by service requests, engineer performance, SLA analysis, and recent customer feedback.

```
JS serviceAgentDashboard.js JS managerDashboard.js X
force-app > main > default > lwc > managerDashboard > JS managerDashboard.js > ManagerDashboard > recentFeedback
1 import { LightningElement, track, wire } from 'lwc';
2 import getManagerStats from '@salesforce/apex/ManagerController.getManagerStats';
3 import getCaseStatusDistribution from '@salesforce/apex/ManagerController.getCaseStatusDistribution';
4 import getTopProducts from '@salesforce/apex/ManagerController.getTopProducts';
5 import getEngineerPerformance from '@salesforce/apex/ManagerController.getEngineerPerformance';
6 import getSlaAnalysis from '@salesforce/apex/ManagerController.getSlaAnalysis';
7 import getRecentFeedback from '@salesforce/apex/ManagerController.getRecentFeedback';
8
9 export default class ManagerDashboard extends LightningElement {
```

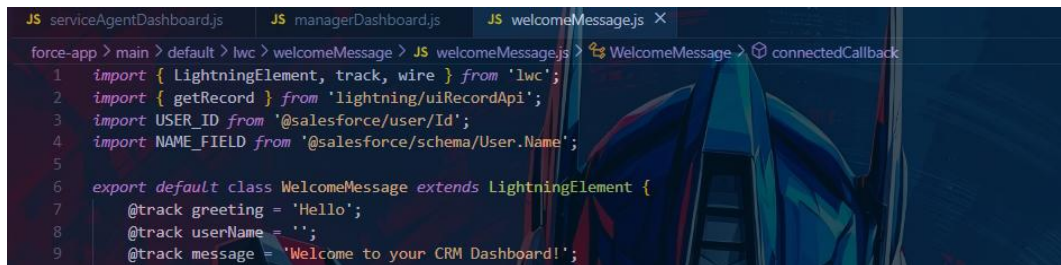
- **engineerDashboard**

- **Use Case:** This dashboard provides field engineers with a clear and concise view of their assigned jobs and dispatch details. It allows them to quickly access the information they need and update the status of their work.
- **Implementation Details:**
 - Displays the engineer's profile information, including their name, email, availability, and skills.
 - Lists all assigned jobs with details such as the case subject, priority, status, and SLA deadline.
 - Includes a section for dispatch jobs with the service date and related case information.
 - Allows engineers to update the status of their assigned jobs.

```
JS engineerDashboard.js X
force-app > main > default > lwc > engineerDashboard > JS engineerDashboard.js > EngineerDashboard > wiredEngineerInfo
1 import { LightningElement, track, wire } from 'lwc';
2 import { ShowToastEvent } from 'lightning/platformShowToastEvent';
3 import getEngineerInfo from '@salesforce/apex/EngineerDashboardController.getEngineerInfo';
4 import getAssignedJobs from '@salesforce/apex/EngineerDashboardController.getAssignedJobs';
5 import getDispatchJobs from '@salesforce/apex/EngineerDashboardController.getDispatchJobs';
6 import updateJobStatus from '@salesforce/apex/EngineerDashboardController.updateJobStatus';
7
8 export default class EngineerDashboard extends LightningElement {
```

- **welcomeMessage**

- **Use Case:** This is a simple, reusable component that displays a personalized welcome message to the logged-in user. It enhances the user experience by providing a friendly greeting based on the time of day.
- **Implementation Details:**
 - Displays a greeting ("Good Morning," "Good Afternoon," or "Good Evening") based on the current time.
 - Shows the name of the logged-in user.



```

JS serviceAgentDashboard.js JS managerDashboard.js JS welcomeMessage.js X
force-app > main > default > lwc > welcomeMessage > JS welcomeMessage.js > WelcomeMessage > connectedCallback
1 import { LightningElement, track, wire } from 'lwc';
2 import { getRecord } from 'lightning/uiRecordApi';
3 import USER_ID from '@salesforce/user/Id';
4 import NAME_FIELD from '@salesforce/schema/User.Name';
5
6 export default class WelcomeMessage extends LightningElement {
7   @track greeting = 'Hello';
8   @track userName = '';
9   @track message = 'Welcome to your CRM Dashboard!';

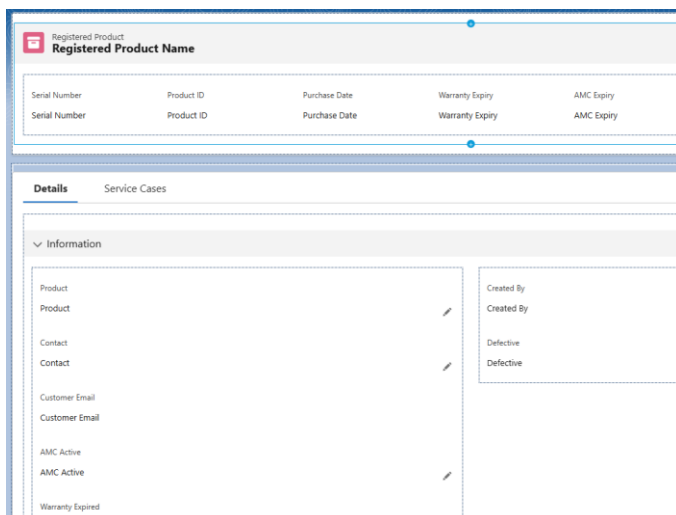
```

Lightning App Builder

The Lightning App Builder is used to create and customize pages with standard and custom components, providing a flexible and tailored user experience.

- **Record Pages**

- **Use Case:** Custom record pages have been created for all major custom objects, including Registered_Product__c, Service_Case__c, Dispatch__c, Feedback__c, and Engineer__c. These pages are designed to provide users with a clear and organized view of all relevant information for a given record.
- **Implementation Details:**
 - The layouts are customized to show the most important fields at the top of the page.
 - Related lists are included to provide easy access to related records (e.g., viewing all service cases for a registered product).
 - The welcomeMessage LWC is included on the home page to greet users.



Registered Product				
Registered Product Name				
Serial Number	Product ID	Purchase Date	Warranty Expiry	AMC Expiry
Serial Number	Product ID	Purchase Date	Warranty Expiry	AMC Expiry

Details		Service Cases	
<div>Information</div> <div> <div>Product</div> <div>Product</div> <div>Contact</div> <div>Contact</div> <div>Customer Email</div> <div>Customer Email</div> <div>AMC Active</div> <div>AMC Active</div> <div>Warranty Expired</div> </div> <div> <div>Created By</div> <div>Created By</div> <div>Defective</div> <div>Defective</div> </div>			

Service Agent

Agent Name

Agent ID	Email ID	Region	Active
Agent ID	Email ID	Region	Active

Open Cases

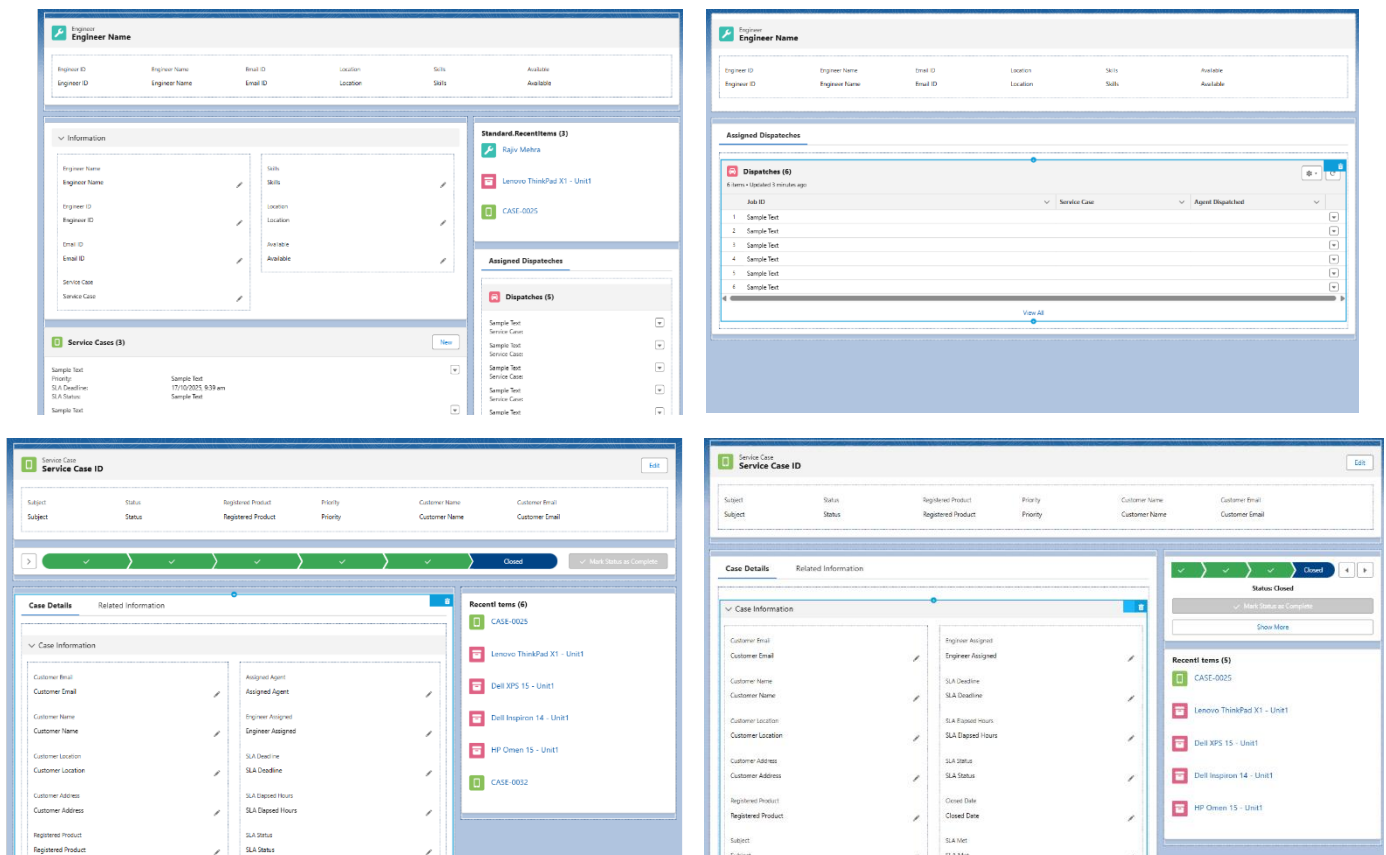
Assigned Dispatches

Service Cases (6)

6 items • Updated 2 minutes ago

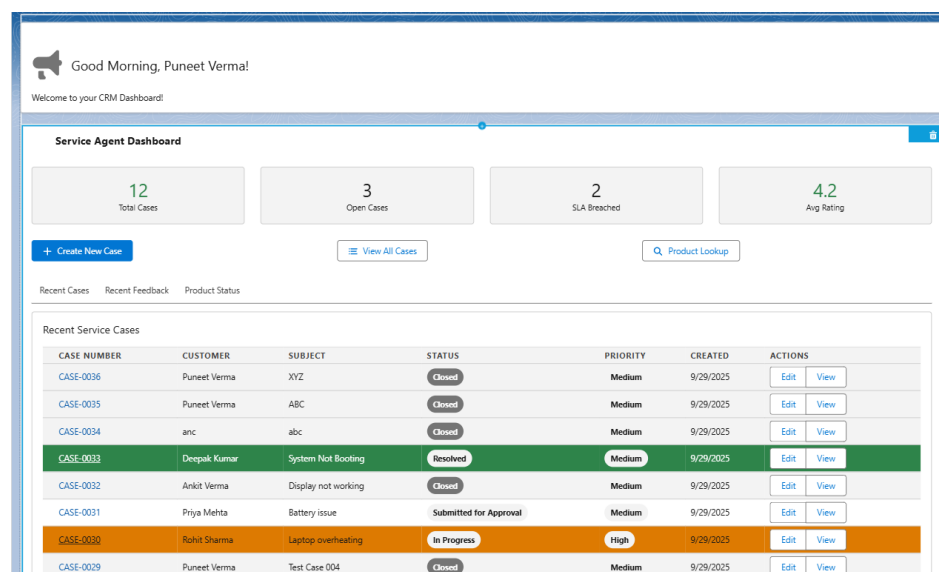
Service Case ID	Status
1 Sample Text	Sample Text
2 Sample Text	Sample Text
3 Sample Text	Sample Text
4 Sample Text	Sample Text
5 Sample Text	Sample Text
6 Sample Text	Sample Text

View All

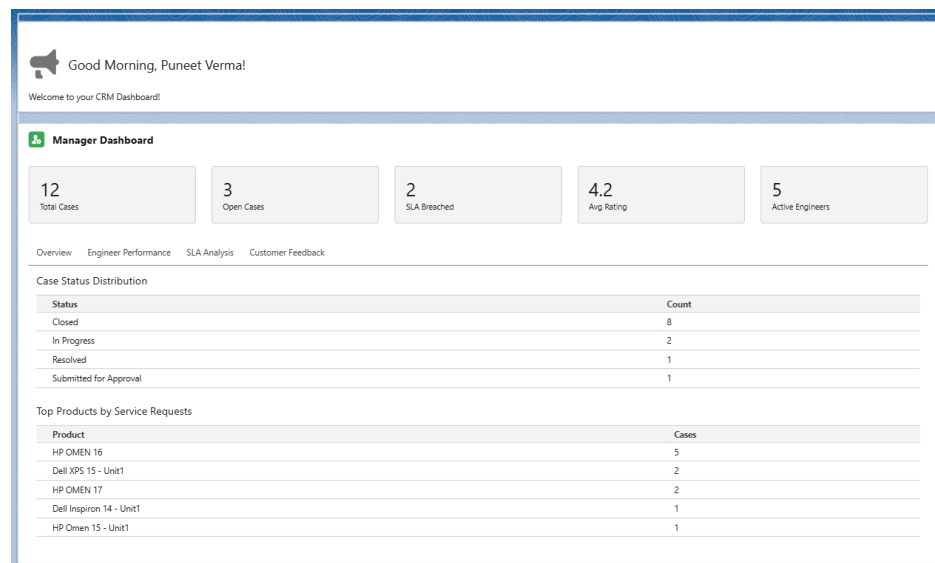


- **Home Page Layouts**

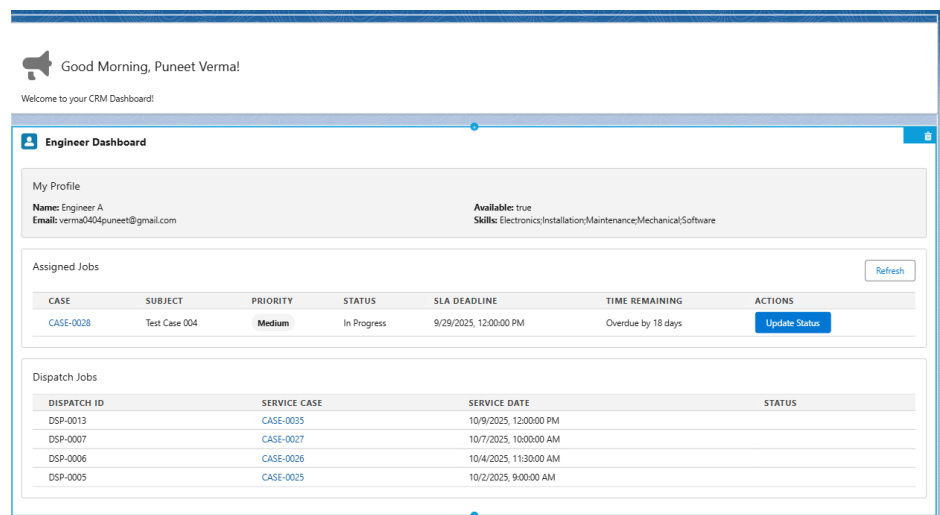
- **Use Case:** The home page is customized for different user roles to provide them with the most relevant information and quick access to their daily tasks.
- **Implementation Details:**
 - **Service Agent Home Page:** Features the serviceAgentDashboard LWC.



- **Manager Home Page:** Features the managerDashboard LWC.



- **Engineer Home Page:** Features the engineerDashboard LWC.



Navigation and Usability

- **Tabs:** Custom tabs have been created for all custom objects to ensure they are easily accessible from the App Launcher and the navigation bar of the custom app.
- **Utility Bar:** The utility bar can be configured to include quick-access components, such as a "Recent Items" list or a notes utility, to further enhance user productivity.