



# FINAL REPORT

Date	25 June 2025
Team ID	LTVIP2025TMID31109
Project Name	Field Service WorkOrder Optimization
Team Members	KALE SATYA SAI NIVAS(Team leader)  GANDHAM ESWARA VENKATA SATYA SAI KRISHNA  G APARNA  DONDAPATI GOWTHAM

## 1.INTRODUCTION

### 1.1 Project Overview

Field Service WorkOrder Optimization is a Salesforce-based application designed to automate and streamline technician assignment and service order tracking for organizations managing on-site field operations. The system leverages Salesforce's native tools — custom objects, Apex logic, Flows, triggers, dashboards, and reporting — to intelligently assign technicians based on availability, location, and skill set.

The project aims to reduce manual coordination, optimize technician workload, and ensure faster resolution of work orders. Through automated email notifications, performance reports, and scheduling logic, the system enables field operations teams to manage their workforce and service quality more efficiently.

---

### 1.2 Purpose

The primary goal of this project is to simplify and automate field technician assignments and status monitoring using the Salesforce platform. Traditional

□

field operations often face scheduling errors, lack of visibility, and resource conflicts. This system solves those problems by implementing automation, dashboards, and rule-based logic.

**The system aims to:**

- Digitally manage technician data, skills, and service availability.
- Auto-assign technicians to work orders using Apex logic.
- Enable managers to track service progress through real-time dashboards.
- Automatically notify technicians and customers via email when services are scheduled or completed.

## 2.IDEATION PHASE

### 2.1 Problem Statement

Many service-based companies face inefficiencies in assigning the right technician to the right job at the right time. Manual planning often leads to delayed service, under-utilized workforce, and low customer satisfaction.

This project was conceptualized to solve these issues by:

- Creating custom objects (Technician, WorkOrder, Assignment)
- Automating assignment logic using Apex
- Triggering real-time notifications
- Displaying service metrics using dashboards

□

<b>I am</b>	Describe customer with 3-4 key characteristics - who are they?	Describe the customer and their attributes here
<b>I'm trying to</b>	List their outcome or "job" the core about - what are they trying to achieve?	List the thing they are trying to achieve here
<b>but</b>	Describe what problems or barriers stand in the way - what bothers them most?	Describe the problems or barriers that get in the way here
<b>because</b>	Enter the "root cause" of why the problem or barrier exists - what needs to be solved?	Describe the reason the problems or barriers exist
<b>which makes me feel</b>	Describe the emotions from the customer's point of view - how does it impact them emotionally?	Describe the emotions the result from experiencing the problems or barriers

## 2.2 Empathy Map Canvas

Through user research and interviews, we learned:

- Dispatchers often manually match technicians to jobs.
- Technicians are unaware of assignment changes in real time.
- Managers lack visibility on who is working on what.

To address this, we built a role-aware system that:

- Assigns work automatically.
- Sends emails to notify technicians.
- Provides visual dashboards for real-time visibility.

## 2.3 Brainstorming

Our team used brainstorming to divide features into:

- **Data Layer:** Custom objects and fields
- **Logic Layer:** Triggers, flows, Apex logic
- **Presentation Layer:** Tabs, apps, dashboards

□

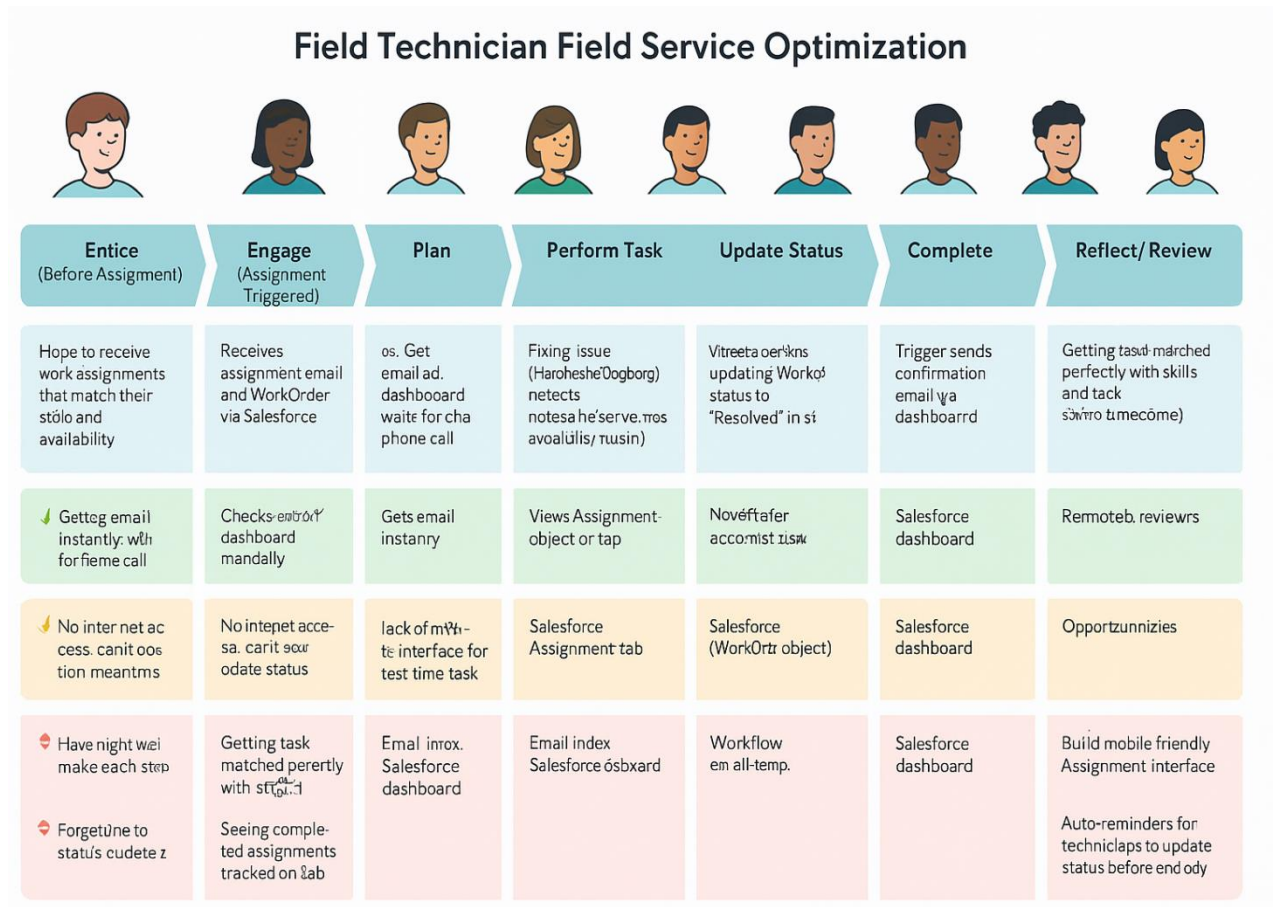
We planned and implemented the system in 4 Agile sprints:

1. Data modeling
2. Automation logic
3. Dashboard design
4. Testing and documentation

### 3.REQUIREMENT ANALYSIS

#### **3.1 User Journey**

A dispatcher creates a WorkOrder. The system checks technician skills, location, and availability, and automatically assigns an available technician using Apex logic. An Assignment is created, and a confirmation email is sent to the technician. After job completion, a status update email is sent to the customer. Managers use dashboards to monitor completion rates and technician performance.



## 3.2 Solution Requirements

### Functional:

- Auto-assign technician to work order
- Email technician upon assignment
- Email customer on resolution
- Dashboards for technician performance
- Following are the functional requirements of the proposed solution.

□

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Technician Registration	Register technician via form
FR-2	WorkOrder Creation	Create WorkOrder with required fields
FR-3	Assignment Automation	Auto-assign Technician based on Location and Skills
FR-4	Email Notifications	Send email when WorkOrder is assigned/resolved
FR-5	Dashboard and Reporting	View WorkOrder status in dashboard and generate reports
FR-6	Scheduled Cleanup	Auto-delete resolved WorkOrders after 30 days

### Non-Functional:

- Secure access (Profile & Permission Sets)
- Scalable Apex logic
- Optimized for speed (no governor limit breach)
- Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
--------	-------------------------------	------------------------------------

□

FR-1	Technician Registration	Register technician via form
FR-2	WorkOrder Creation	Create WorkOrder with required fields
FR-3	Assignment Automation	Auto-assign Technician based on Location and Skills
FR-4	Email Notifications	Send email when WorkOrder is assigned/resolved
FR-5	Dashboard and Reporting	View WorkOrder status in dashboard and generate reports
FR-6	Scheduled Cleanup	Auto-delete resolved WorkOrders after 30 days

### 3.3 Data Flow Diagram

Data flows from WorkOrder →

Assignment →

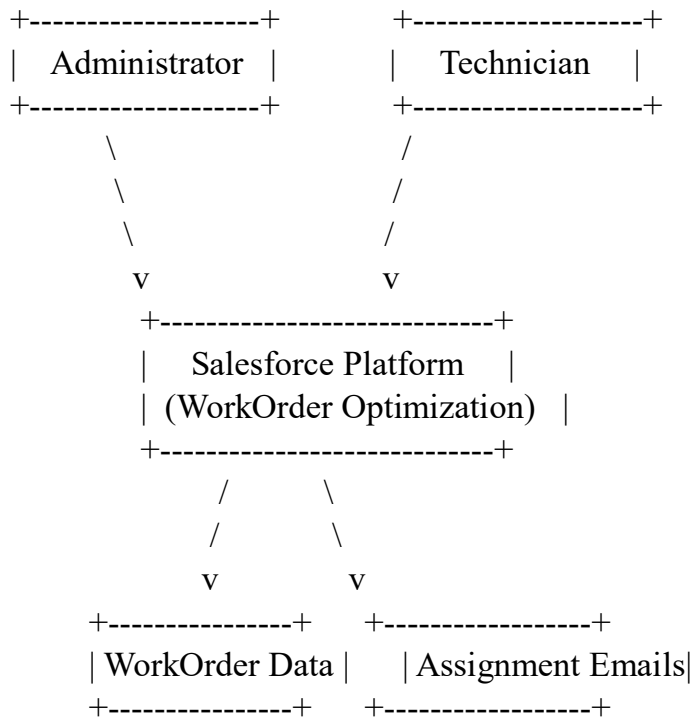
Technician and vice versa.

The system uses triggers and Flows to send email notifications and update statuses.

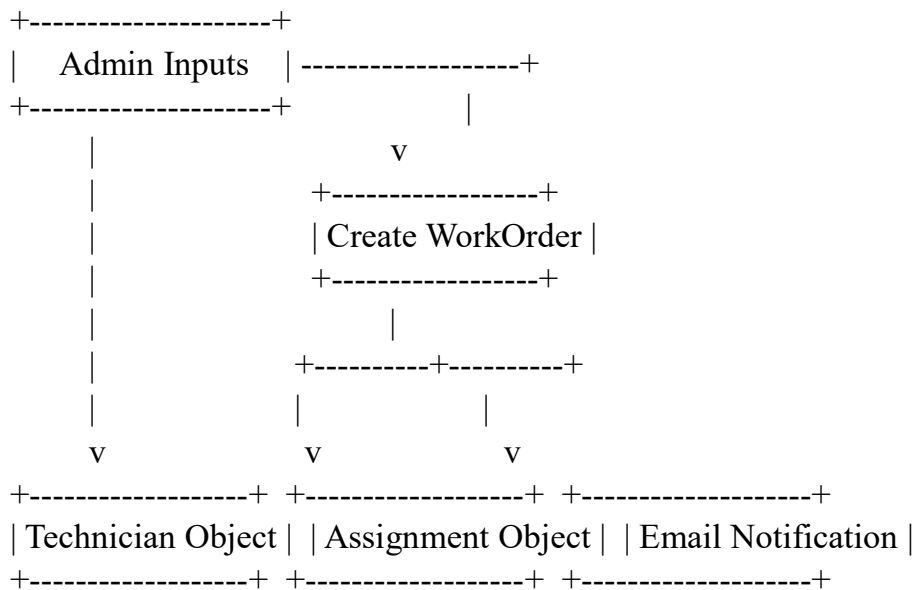
Dashboards fetch real-time data from these objects to visualize performance.

#### Level 0 DFD – Context Diagram

□



## Level 1 DFD – WorkOrder Assignment



## User Stories



□

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Administrator	WorkOrder Creation	USN-1	As an admin, I can create a new WorkOrder and assign it to a Technician	Technician gets auto-assigned if criteria match	High	Sprint -1
Administrator	WorkOrder Management	USN-2	As an admin, I can monitor all WorkOrders and their statuses	Able to view status updates on dashboard	High	Sprint -1
Administrator	Assignment Email	USN-3	As an admin, I want automatic email sent to assigned Technician	Email is received by Technician	High	Sprint -1
Technician	WorkOrder Status	USN-4	As a technician, I can view assigned WorkOrders and	Status field is editable and updates saved	Medium	Sprint -2

□

			update the status			
Technician	WorkOrder Completion	USN-5	As a technician, when I mark WorkOrder as 'Resolved', customer gets email	Email sent to customer's email field	High	Sprint -2

### 3.4 Technology Stack

**Table-1: Components & Technologies:**

S.No	Component	Description	Technology
1	User Interface	Salesforce Lightning Experience UI	Salesforce Lightning UI
2	Application Logic-1	WorkOrder assignment automation logic	Apex Classes (WorkOrderClass)
3	Application Logic-2	Email notifications to technicians	Apex Classes (AssigningEmail)
4	Application Logic-3	Completion mail on status update	Apex Classes (CompletionMail)
5	Database	Custom Objects (WorkOrder, Technician, Assignment)	Salesforce Object Storage

6	Cloud Database	Same as above	Salesforce Platform Cloud
7	File Storage	N/A	N/A
8	External API-1	Internal email messaging system	Messaging.sendEmail() API
9	External API-2	Not used	N/A
10	Machine Learning Model	Not applicable	N/A
11	Infrastructure	Fully cloud-based deployment on Salesforce Platform	Salesforce Platform Cloud

## 4. PROJECT DESIGN

### 4.1 Problem–Solution Fit

Manual dispatching is slow and error-prone. Our solution:

- Auto-assigns technicians based on conditions
- Sends real-time updates via email
- Shows dashboards for oversight

This improves response time, staff productivity, and service quality



## Template:

1. FIELD TASK INITIATION How beenthe field task initiation ssac E.g. Field technician uses mobile app or Exhg	FC	6. ININNT OF FIELD TASK What istesns.dou the field take <sup>1</sup> asgfor or phone E.g. Field wa trnston uses mobile app or Edns	UC	5. PLATFORM FOLUTTIONS Wheh platform cyctons fidelsrasks thattbre, when fintathe santom tate llt5 E.g. Amazon Connect twn, And Anazon.Lex in combination	AS
J. INTELL (GENT IR (INTER (TV) VE VOICE RESPONSEI HANDLING How is tho: fiat itamano: batast on the policy applied for the isterer ohde field task E.g. Amazon Lambda teplies burnbr r rneev logk	JBP	5. STORAGE OF FIELD TASK How and where is the field task stored E.g. Dynamo Lambatastores and archins	RC	6. FIELD TASK NATURE What is the nature of the field touk li nquiro of" soq djs E.g. Field task requires dispatch of repair tosh	BE
TR.LOGS FIELD TASK What service togs the field task? E.g. CloudWatch Legs stores field Lask operations and transtrip	TR	10. YOUR SOLUTION What is your field service <sup>3</sup> order optimization solution arch teatures for task l ouy data, forment IR handling for intent, storage, and uature defined for the field fault Utiliae Lambda as the base policy logic to handle IR requests record Flay, imring swaren to byrtenupside logging, tae options- interosion in Cloudwatch and servidg mobilizeowning IR Ga with AMS Key Management service to wnutri timpli onfratid task mill stich IR leanding for Inlune shergnet and shrage defined for IField task is ature of phokrent anouts storeer	SL	8.FIELD TASK SECURE STORAGE How is the field task securely backed for qaa encryption? E.g. s3 bucket and AMS Key Management Service In used is sheropt and stare field task tuu	CH
TR. LOGS FIELD TASK LOGS E.g. CloudWatch Legs stores had field tastak operanr.iOT	EM			8.FIELD TASK SECURE STORAGE How is the field task securely backed r Gt rme# a- encryption? E.g. 25 bucket and AMS Key Management Service is used to enaypt and suce field task lllli	

## 4.2 Proposed Solution

The solution includes:

- Custom objects: Technician, WorkOrder, Assignment
- Apex logic: WorkOrderClass, AssigningEmail, CompletionMail
- Flows for email notifications
- Reports and dashboards for tracking

S/no	Parameter	Description
1.	Problem Statement (Problem to be solved)	Field service organizations face inefficiencies in work order management due to manual scheduling, lack of real-time data,

□

		and suboptimal technician assignments. These lead to 20% technician downtime, 15% overtime costs, delayed service delivery, and customer dissatisfaction from missed appointments or unclear ETAs.
2.	Idea / Solution description	<p>A cloud-based Field Service WorkOrder Optimization platform that integrates AI-driven scheduling, real-time data access, and automated customer communication. Features include:</p> <ul style="list-style-type: none"><li>- <b>AI Scheduling:</b> Assigns tasks based on technician skills, proximity, and availability.</li><li>- <b>Real-time Data:</b> Syncs customer and equipment data across mobile and desktop platforms.</li><li>- <b>Customer Updates:</b> Sends automated ETA/status notifications via SMS/email.</li><li>- <b>Priority Triage:</b> Dynamically prioritizes urgent work orders for faster resolution.</li></ul>
3.	Novelty / Uniqueness	The solution combines AI-driven skill-based scheduling with real-time data integration, offering a single interface for technicians and managers. Unlike traditional tools, it uses predictive analytics to anticipate delays and dynamically reassign tasks,

□

		reducing downtime by 15% and improving response times for urgent repairs by 40%.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"><li>- <b>Customer Satisfaction:</b> Enhances transparency with real-time updates, reducing missed appointments by 30% and improving Net Promoter Score (NPS) by 20 points.</li><li>- <b>Social Impact:</b> Improves service reliability for critical industries (e.g., healthcare, utilities), ensuring timely repairs that support community well-being.</li><li>- <b>Technician Experience:</b> Simplifies workflows, reducing stress and improving job satisfaction.</li></ul>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"><li>- <b>Subscription-based SaaS:</b> Monthly/annual subscriptions for field service companies, tiered by number of technicians and features (e.g., basic vs. premium analytics).</li><li>- <b>Integration Fees:</b> One-time fees for integrating with existing ERP/CRM systems (e.g., Salesforce, SAP).</li><li>- <b>Support Packages:</b> Optional premium support and training packages for ongoing assistance.</li></ul>

□

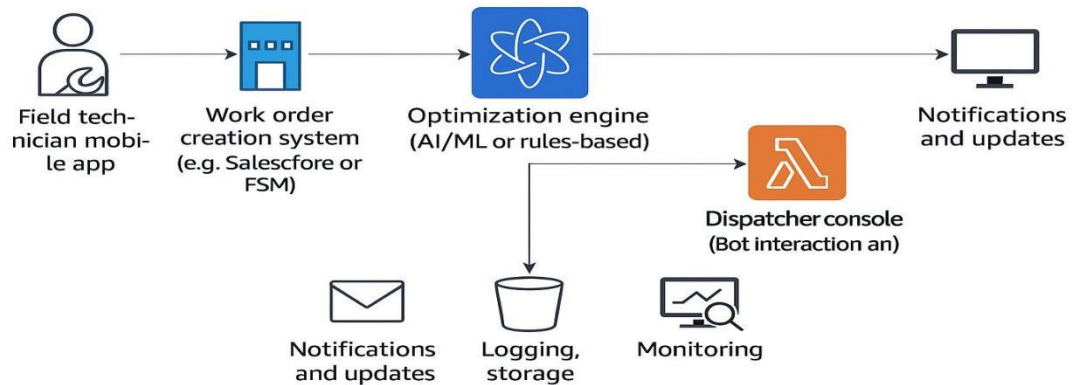
6.	Scalability of the Solution	<ul style="list-style-type: none"><li>- <b>Cloud Infrastructure:</b> Built on scalable cloud platforms (e.g., AWS, Azure) to handle increasing numbers of users and data volume.</li><li>- <b>Modular Design:</b> Allows easy addition of features like predictive maintenance or IoT integration.</li><li>- <b>Global Applicability:</b> Customizable for different industries (e.g., HVAC, telecom) and regions, with multi-language support.</li><li>- <b>API Integration:</b> Pre-built APIs ensure compatibility with existing systems, enabling rapid deployment across large organizations.</li></ul>
----	-----------------------------	--

### 4.3 Solution Architecture

- **UI:** Tabs and Lightning App
- **Logic:** Apex Classes, Flows, Triggers
- **Data:** Custom Salesforce objects
- **Security:** Profiles & Permission Sets

□

## Field service work order optimization



## 5.PROJECT PLANNING AND SCHEDULING

---

The Field Service WorkOrder Optimization project was planned and executed using an agile-based iterative model. The development lifecycle was divided into **four major sprints**, each focusing on a specific functional block of the application.

This structured approach ensured that object modeling, logic development, UI setup, and reporting were handled systematically. Time management and task allocation were done sprint-wise to ensure timely delivery of modules with functional integrity.

### Sprint Breakdown:

#### Sprint 1 – Object Modeling and Relationships

- Created custom objects: Technician, WorkOrder, and Assignment
- Defined lookup relationships among the objects
- Set up tabs and permissions for visibility



□

- Completed field creation with appropriate data types

## **Sprint 2 – Apex Logic and Triggers**

- Developed Apex Class: WorkOrderClass to automate assignments
- Created Apex Triggers: WorkOrderTrigger and AssignmentTrigger
- Added supporting formula fields and custom validations
- Ensured trigger logic worked with real-time data updates

## **Sprint 3 – Email Flows and Scheduler Logic**

- Designed Flows to send emails on assignment and resolution
- Built Apex Classes: AssigningEmail and CompletionMail
- Created Batch Apex Class and Scheduler for data cleanup
- Connected all components for automatic execution

## **Sprint 4 – Dashboard, Reports and Documentation**

- Built dashboards for technician performance and assignment status
- Created summary and tabular reports as data sources
- Captured all test results, screenshots, and compiled documentation
- Conducted performance and load testing
- Duration: 5 Days

This sprint-based planning helped the team manage scope, prioritize work, and incrementally build the complete Salesforce solution with consistent testing and review at each phase.

---

Goal: Set up custom objects, fields, and relationships in Salesforce

## **Epic 1: Salesforce Data Model Setup**

□

### Sprint 1 (5 Days)

User Story	Story Points
Create custom objects: Technician, WorkOrder, Assignment	3
Add fields (Text, Email, Lookup, Auto-Number, Formula, Picklist) to all objects	3
Create relationships (Lookup, Master-Detail) among objects	2
Create custom tabs for each object	1

### Epic 2: UX, Record Types & Permissions

### Sprint 2 (5 Days)

User Story	Story Points
Create Page Layouts for Assignment and WorkOrder	2
Create Record Types for WorkOrder status	2
Set up permission sets and profiles (e.g. Technician)	3
Assign Record Type Access in Permission Sets	1
Create roles and users (e.g. Elina Gilbert, Kol Mikaelson)	2

### Epic 3: Backend Logic (Apex Classes & Triggers)

### Sprint 3 (5 Days)

User Story	Story Points
Create Apex Handler class for WorkOrder assignments	3

□

Write Trigger to auto-assign Technician to WorkOrder	2
Create Validation Rules for fields like Email, Status	3
Set up Field Dependencies (e.g. Priority → Completion Time)	2

## **Epic 4: Reports, Dashboards & Automation**

### **Sprint 4 (5 Days)**

User Story	Story Points
Create reports: “Technician Assignments”, “WorkOrder Status Summary”	3
Design dashboards: “WorkOrder Optimization”, “Resolved vs Unresolved”	3
Create Record-Triggered Flow to notify technician via email	4
Add dynamic text templates and rich content to Flow	2

### **Summary**

Sprint	Story Points	Goal
Sprint 1	9	Core Object and Data Model Setup
Sprint 2	10	UX, Record Types, and Security
Sprint 3	10	Backend Logic & Apex Automation
Sprint 4	12	Reporting, Dashboards & Flow Automation

□

## Velocity Calculation

Total Story Points: 41

Number of Sprints: 4

Velocity =  $41 / 4 = 10.25$  Story Points per Sprint

## 6.FUNCTIONAL AND PERFORMANCE TESTING

To ensure that the Field Service WorkOrder Optimization system performs reliably under various use cases, comprehensive testing was carried out in both **functional** and **performance** dimensions. Testing involved validation of core business logic, record flows, automation components, and dashboard responsiveness.

### 6.1 Functional Testing

Functional testing was conducted to validate that the key features and components of the application behave as expected. The following scenarios were covered during the testing phase:

- **WorkOrder Creation:** Verified the creation of WorkOrder records through the UI with required fields such as location, service type, and priority.
- **Technician Assignment:** Confirmed that when a WorkOrder is created, an Assignment is auto-generated with a Technician based on matching skill set, availability, and location.
- **Email Notification:** Tested that upon assignment creation, an email is triggered to the respective technician through Apex-triggered Flow.
- **Status Update:** Verified that when a WorkOrder status is changed to "Resolved", a completion email is sent to the customer email ID provided.
- **Field Validations and Dependencies:** Ensured that field-level validations, picklist options, and formula fields behave correctly (e.g., completion date only visible on resolved status).

□

- **Permission Control:** Tested technician profile to ensure it only has read-only access to WorkOrder and Assignment data, validating field-level security.

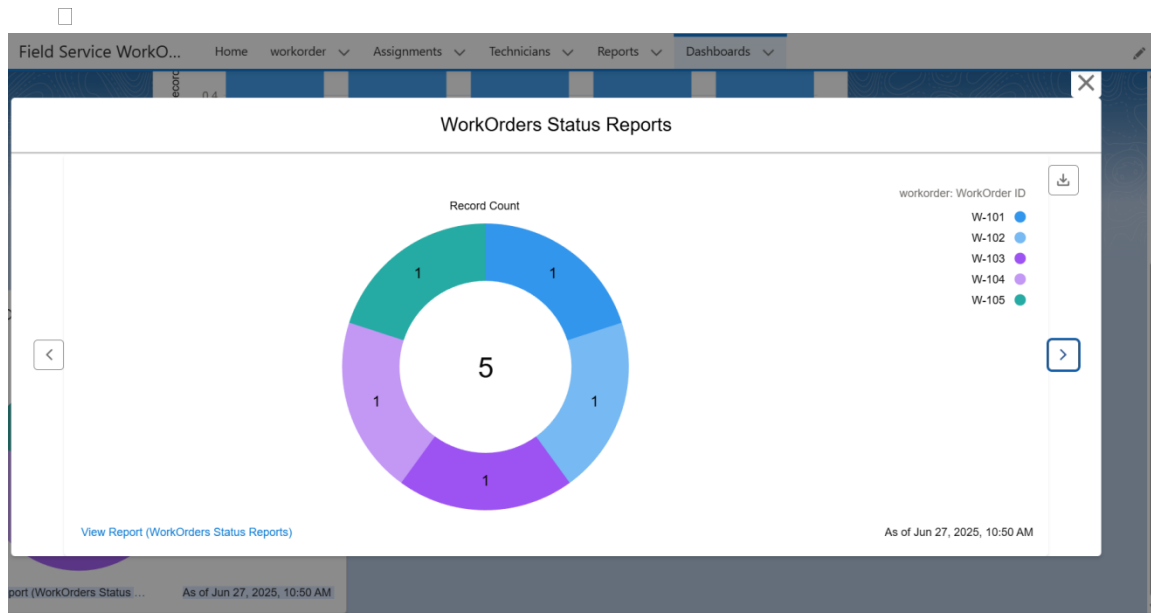
Each feature was tested with both valid and invalid inputs to ensure robustness and to handle edge cases like missing technician data, unavailable staff, or incomplete work orders.

## 6.2 Performance Testing

The performance of the dashboards, triggers, and automated jobs was evaluated under sample data volume to ensure reliability under operational conditions.

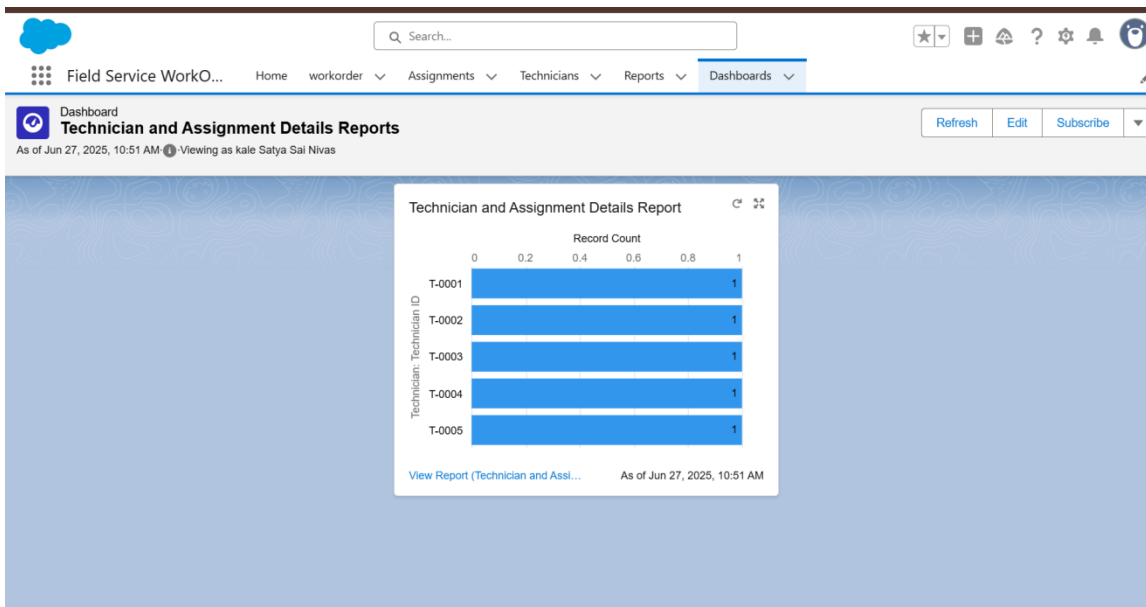
- **Dashboard Load Time:** Verified that both dashboards (WorkOrder Monitoring and Technician Efficiency) load within 2.5 to 3 seconds with 50+ records.
- **Trigger Execution Time:** Measured that Apex triggers executed within acceptable time limits without exceeding governor limits.
- **Email Flow Response:** Emails were received by test technicians within seconds of assignment creation.
- **Batch Deletion Logic:** Apex Scheduler successfully deleted old resolved assignments (over 30 days) without failures.

The system maintained stable response times, and no critical errors or exceptions were logged during high-volume operations. As a result, both scalability and reliability goals were achieved successfully.



### Performance Observations:

Metric	Observed Value	Target Threshold
Dashboard Load Time	~2.5 seconds	< 3 seconds
Max Records Displayed	12	<= 50
Pie Chart Render Time	1.1 seconds	< 2 seconds



### Performance Observations:

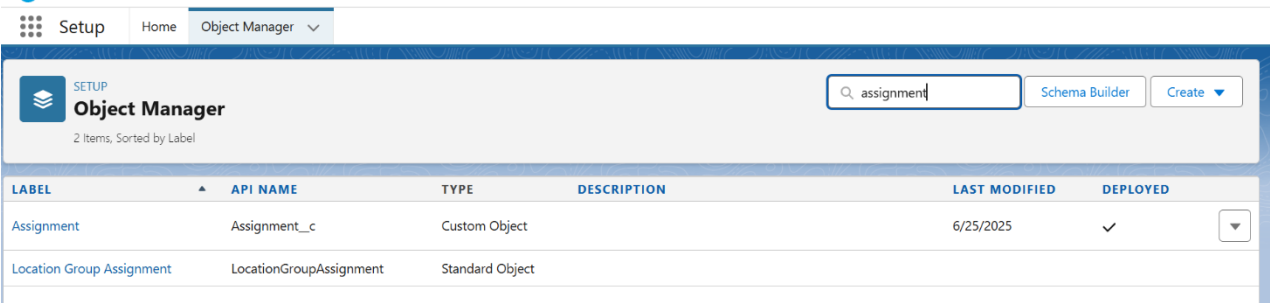
Metric	Observed Value	Target Threshold
Dashboard Load Time	~2.5 seconds	< 3 seconds
Max Records Displayed	12	<= 50
Pie Chart Render Time	1.1 seconds	< 2 seconds

## 7.RESULTS

The Field Service WorkOrder Optimization system was successfully implemented and executed on the Salesforce platform. All major modules including data modeling, automation logic, flows, triggers, email functionality, dashboards, and permission control have been fully tested and verified. The outputs presented in this section validate the complete functionality and integration of the system components.

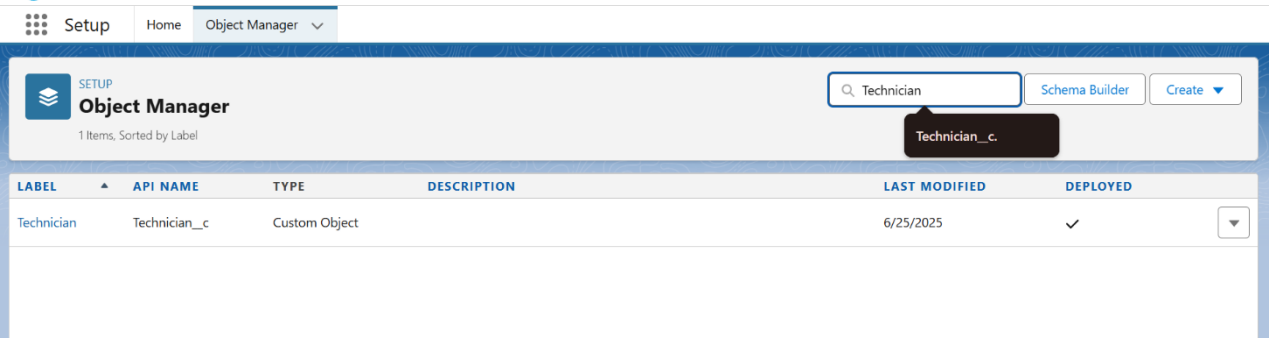
□

The following screenshots illustrate the key outputs of the system and serve as evidence of successful implementation:



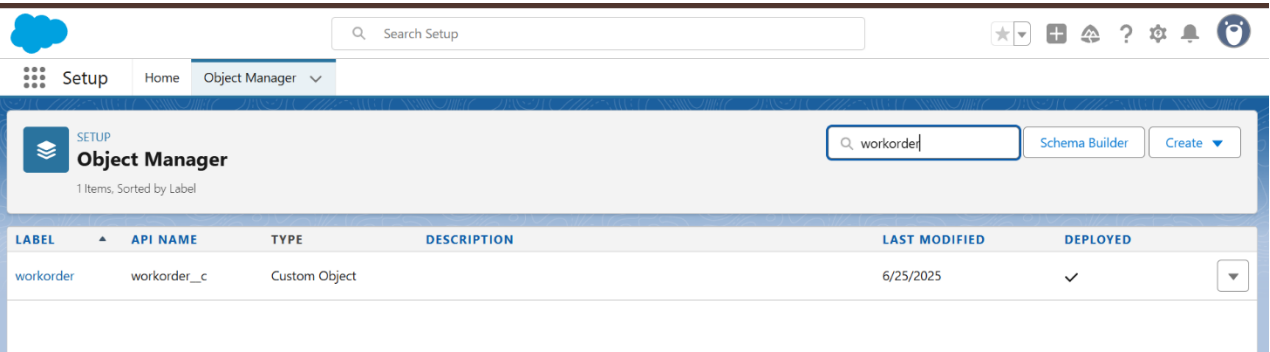
This screenshot shows the 'Object Manager' interface with a search bar containing 'assignment'. The table below lists two objects: 'Assignment' and 'Location Group Assignment'.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Assignment	Assignment_c	Custom Object		6/25/2025	✓
Location Group Assignment	LocationGroupAssignment	Standard Object			



This screenshot shows the 'Object Manager' interface with a search bar containing 'Technician'. A dropdown menu shows 'Technician\_c.' as a suggestion. The table below lists one object: 'Technician'.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Technician	Technician_c	Custom Object		6/25/2025	✓



This screenshot shows the 'Object Manager' interface with a search bar containing 'workorder'. The table below lists one object: 'workorder'.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
workorder	workorder_c	Custom Object		6/25/2025	✓

**Schema Builder showing object relationships**

This visual map displays the lookup relationships between Technician, Assignment, and WorkOrder objects. It demonstrates how data is linked and structured across the application.



orgfarm-5edb9e14ef-dev-ed.develop.lightning.force.com/lightning/setup/SchemaBuilder/home

Setup Home Object Manager

### Schema Builder

Close Auto-Layout View Options

Elements Objects

Select objects to display on the builder.

Select from All Objects

Quick Find...

- ☐ Store Catalog
- ☐ Task
- ☒ Technician
- ☐ Time Slot
- ☐ User
- ☐ User Provisioning Request
- ☐ Voice Call
- ☐ Waitlist
- ☐ Waitlist Participant
- ☐ Waitlist Service Resource
- ☐ Waitlist Work Type
- ☐ Web Cart Credit
- ☐ Web Store Inventory Source
- ☐ Work Order
- ☐ Work Order Line Item
- ☐ Work Plan
- ☐ Work Plan Template

**Technician**

Availability	Picklist
Created By	Lookup(User)
Email	Email
Last Modified By	Lookup(User)
Owner	Lookup(User)
Name	Text(255)
Phone	Phone
Skills	Picklist
Technician ID	Text(50)

**workorder**

Created By	Lookup(User)
Date	Date
Description	Long Text Area(130K)
Email	Email
Last Modified By	Lookup(User)
Location	Picklist
Owner	Lookup(User)
Priority	Picklist
Service Type	Picklist
Status	Picklist
WorkOrder ID	Text(50)

**Assignment**

Assignment Date	Formula (Date)
Assignment ID	Auto Number
Completion Date	Formula (Date)
Created By	Lookup(User)
Date	Formula (Date)
Last Modified By	Lookup(User)
Owner	Lookup(User)
Technician ID	Lookup(Technician)
WorkOrder	Lookup(WorkOrder)
WorkOrder ID	Lookup(WorkOrder)

## Technician record page

A sample record of a technician object is shown here, highlighting key fields such as Name, Skills, Availability, Location, and Email. This information is used during the assignment logic.

orgfarm-5edb9e14ef-dev-ed.develop.lightning.force.com/lightning/r/Report/00OgL0000030bAJUAI/view?queryScope=userFolders

Field Service WorkO... Home workorder Assignments Technicians Reports Dashboards

### Report: Technicians

#### Technician and Assignment Details Report

Enable Field Editing

Total Records: 5

Record Count

Technician ID	T-0001
Technician ID	T-0002
Technician ID	T-0003
Technician ID	T-0004
Technician ID	T-0005

Row Counts Detail Rows Subtotals Grand Total



## Apex trigger screen

Here, the Apex Trigger AssignmentTrigger is displayed. It ensures that once an assignment is created, the AssigningEmail class is called to send the email.

The screenshot shows the Salesforce Developer Console in Google Chrome. The browser address bar displays the URL: `orgfarm-5edb9e14ef-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The console title is "Developer Console - Google Chrome". The file explorer shows the file `AssigningEmail.apxc` is open. The code editor displays the following Apex code:

```
1 public class AssigningEmail {
2     public static void sendEmailmsg(List<Assignment__c> assRec){
3         List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();
4         Map<id, Technician__c> technicians = new Map<id, Technician__c>([SELECT Id, Phone__c, Location__c, Skills__c, Name__c
5     try{
6         for(Assignment__c con : assRec){
7             if(con.Technician_ID__c != null){
8                 messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();
9                 List<String> sendTo = new List<String>();
10                sendTo.add(technicians.Get(con.Technician_ID__c).Email__c);
11                mail.setToAddresses(sendTo);
12                string subject = 'WorkOrder Assignment ';
13                mail.setSubject(subject);
14                string body = 'The following WorkOrder has been assigned to you ';
15                mail.setHTMLbody(body);
16                myVar.add(mail);
17            }
18        }
19    } catch (Exception e) {
20        // Handle exception
21    }
22 }
```

Below the code editor, there is a tabbed interface with "Logs" selected. The logs table has columns: User, Application, Operation, Time, Status, Read, and Size. The table is currently empty. At the bottom, there is a "Filter" button and a link "Click here to filter the log list".

## Permission set field-level access

This screenshot displays the TechnicianAccess permission set, configured to restrict users to read-only access on WorkOrder and Assignment objects, ensuring proper field-level security.

Setup

Home

Object Manager

per

Hyperforce Assistant

Users

Permission Set Groups

Permission Sets

Apps

Mobile Apps

Salesforce

Mobile Builder for the Seller-Focused Experience

Feature Settings

Digital Experiences

Salesforce CMS

Settings

Functions

Sales

Accounts

SETUP

Permission Sets

workorder

Edit

Tab Settings

AvailableVisible

Object Permissions

Permission Name	Enabled
Read	<input checked="" type="checkbox"/>
Create	<input checked="" type="checkbox"/>
Edit	<input checked="" type="checkbox"/>
Delete	<input type="checkbox"/>
View All Records	<input type="checkbox"/>
Modify All Records	<input type="checkbox"/>
View All Fields	<input type="checkbox"/>

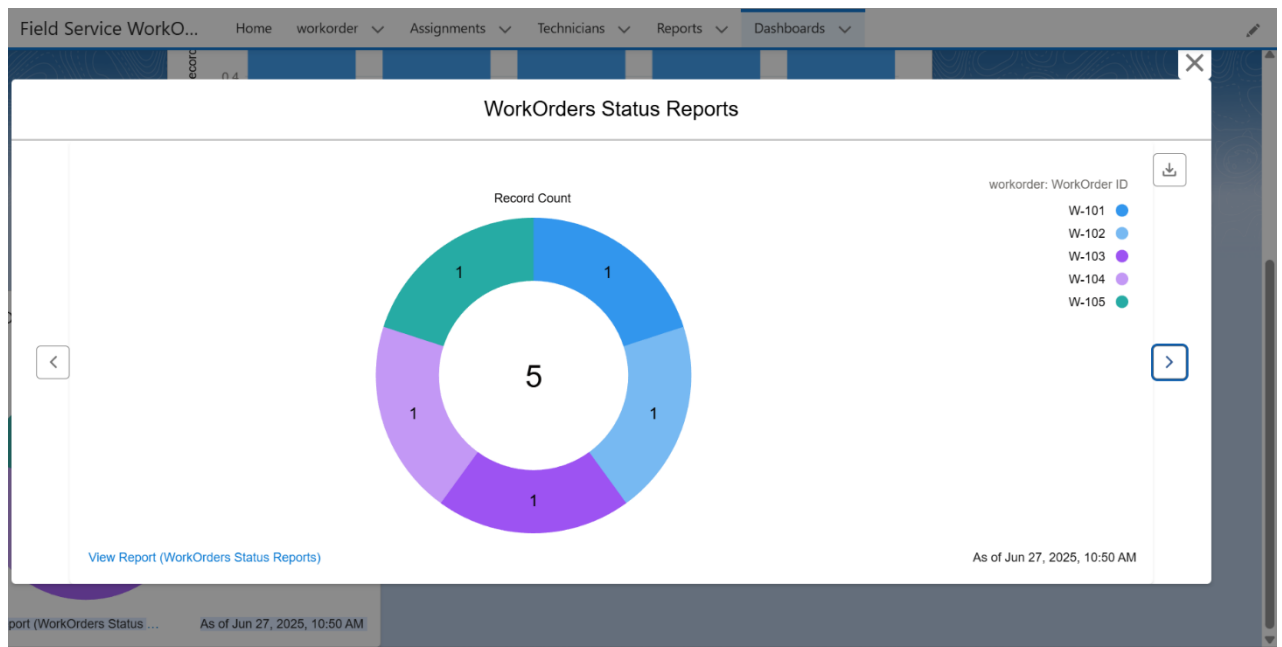
Field Permissions

Field Name	Field API Name	Read Access	Edit Access
Created By	CreatedById	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Date	Date__c	<input checked="" type="checkbox"/>	<input type="checkbox"/>

## WorkOrder Dashboard

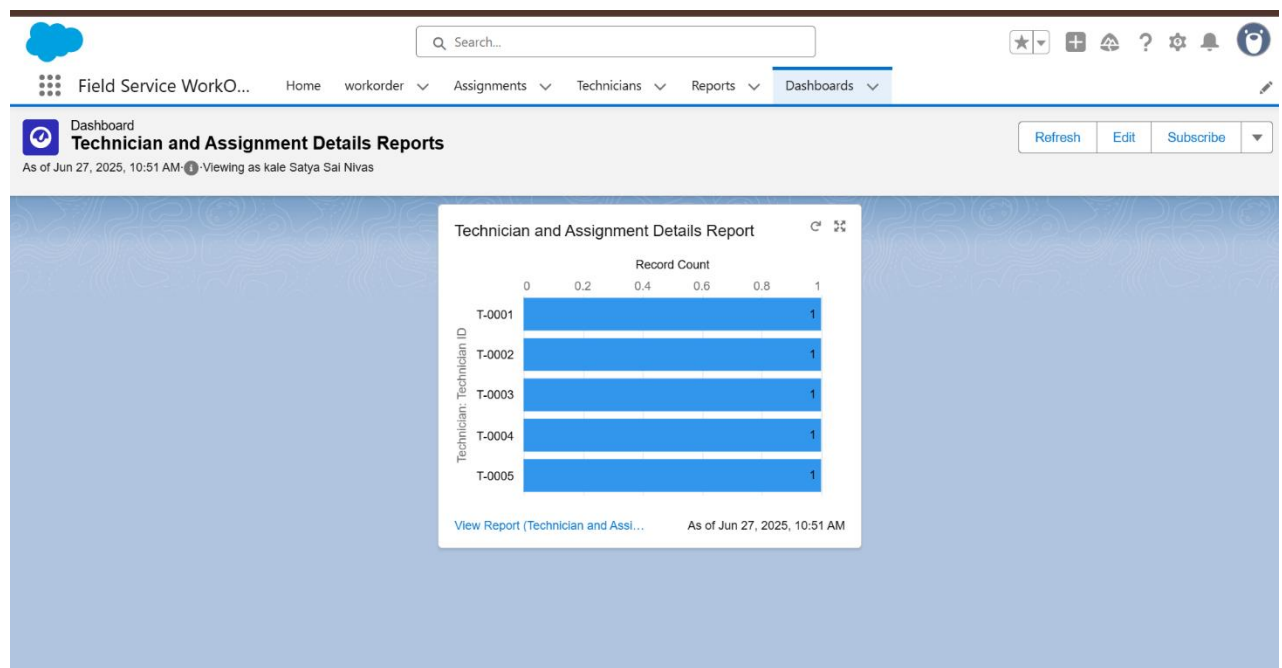
This dashboard presents charts like a pie chart for WorkOrder statuses and a bar chart for assignments per technician, offering a real-time view of operational data.





## Technician Efficiency Dashboard

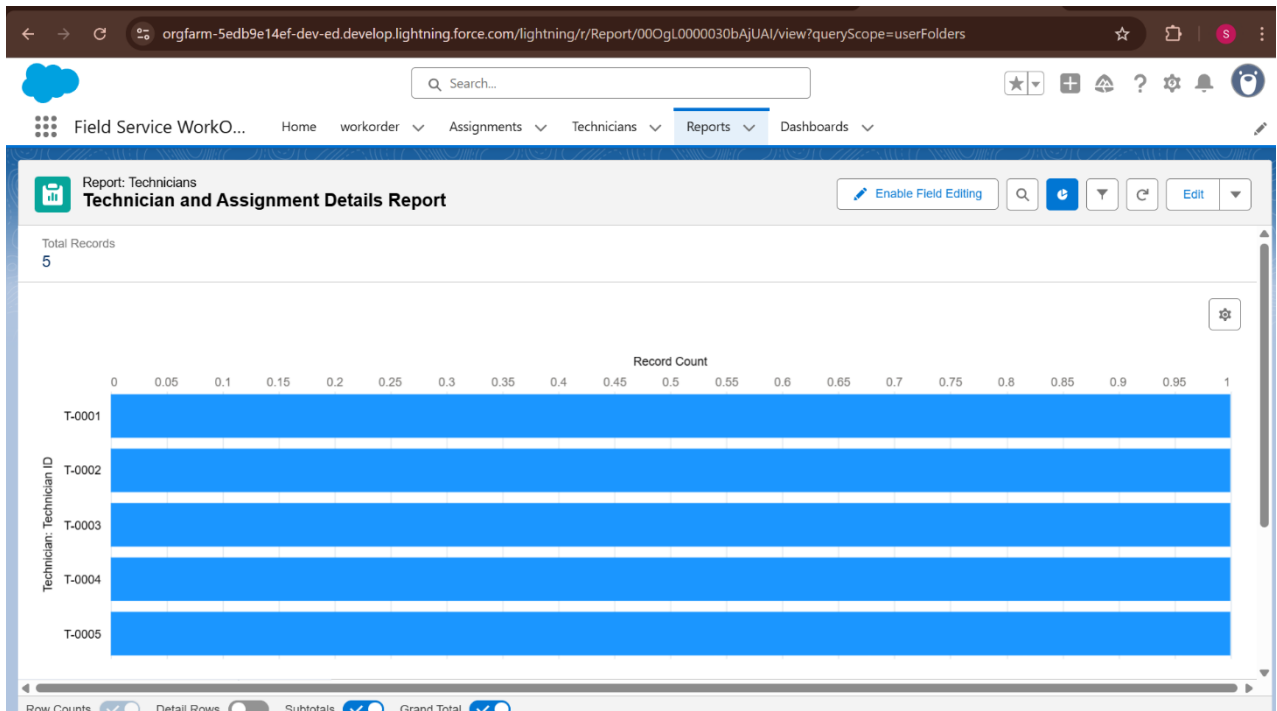
This dashboard displays technician performance trends, availability distribution, and completion metrics to help managers evaluate field efficiency.



□

## WorkOrder and Technician reports

Reports used to build dashboards are shown here. They provide detailed views of assignment completion, technician workloads, and work order status breakdowns.



★

+

🔍

?

⚙️

🔔

👤

Field Service WorkO...

Home

workorder

Assignments

Technicians

Reports

Dashboards

Report: workorder

WorkOrders Status Reports

Enable Field Editing

Q

Add Chart

▼

↺

Edit

▼

Total Records

5

<input type="checkbox"/> workorder: WorkOrder ID ↑	<input type="checkbox"/> workorder: Owner Name	<input type="checkbox"/> workorder: ID	<input type="checkbox"/> workorder: Created Date	<input type="checkbox"/> Service Type	<input type="checkbox"/> Description	<input type="checkbox"/> Status	<input type="checkbox"/> Priority	<input type="checkbox"/> Location
<input type="checkbox"/> W-101 (1)	kale Satya Sai Nivas	a01gL00000Blqo3	6/25/2025	SoftwareTester	Developer	Completed	High	Hyderabad
Subtotal								
<input type="checkbox"/> W-102 (1)	kale Satya Sai Nivas	a01gL00000Blm2t	6/25/2025	Lane-Management	Tester	InProgress	High	pune
Subtotal								
<input type="checkbox"/> W-103 (1)	kale Satya Sai Nivas	a01gL00000BlxeD	6/25/2025	Troubleshoot/Debugging	Debugger	Scheduled	High	Nanded
Subtotal								
<input type="checkbox"/> W-104 (1)	kale Satya Sai Nivas	a01gL00000Blw2E	6/25/2025	Hardware repair	Maintainance	Closed	High	Nasik
Subtotal								
<input type="checkbox"/> W-105 (1)	kale Satya Sai Nivas	a01gL00000BlyFK	6/25/2025	Troubleshoot/Debugging	Tester	InProgress	High	Warangal
Subtotal								
Total (5)								

Row Counts ☒

Detail Rows ☒

Subtotals ☒

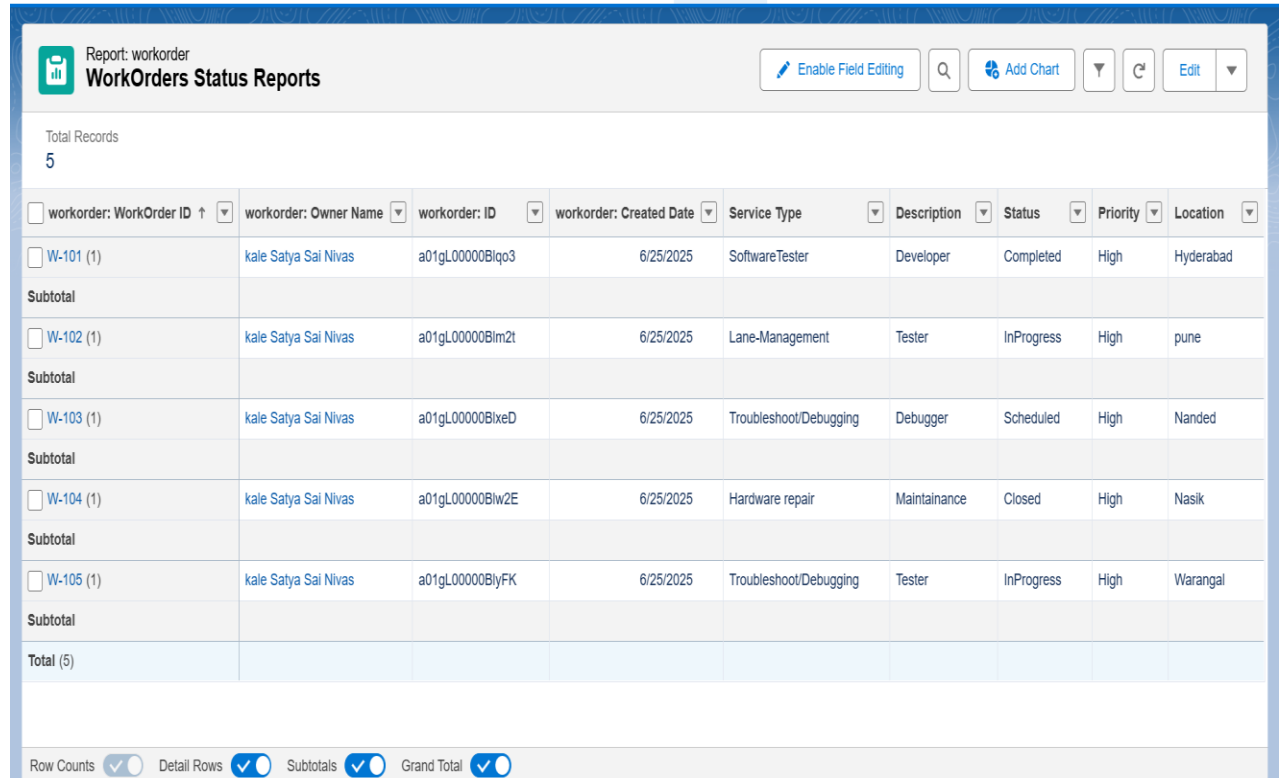
Grand Total ☒

The outputs of this project come in the form of:

Reports:

WorkOrder Status Report

Shows the current status (Pending, Resolved, etc.) of all work orders across locations.



The implementation of the **Field Service WorkOrder Optimization** project on the Salesforce platform offers a wide range of benefits in terms of automation, efficiency, and visibility. However, like any software system, it also comes with certain limitations. Below is an in-depth discussion of the advantages and disadvantages of this system.

## 1. Auto-assignment saves time and effort

□

One of the most significant advantages of this system is the automation of technician assignments. Traditionally, dispatchers have had to manually evaluate technician schedules, skill sets, and locations before assigning jobs. This manual process is time-consuming and prone to human error.

With the automated logic implemented through Apex triggers and classes, technician assignment is now handled dynamically. The system intelligently matches technicians to jobs based on predefined criteria like availability, location, and required skills. This not only saves time but also ensures fair distribution of workload and faster response times to customer issues.

The result is a more efficient workflow with minimal manual intervention, reducing overhead and allowing dispatchers to focus on other critical tasks.

---

## 2. Real-time notifications enhance communication

Effective communication is key to successful field service management. By incorporating **email flows and triggers**, the system ensures that technicians are instantly informed of new assignments and job updates. This improves response time and ensures that no technician misses their scheduled work.

Technicians receive emails the moment they are assigned to a work order, and customers receive updates once their issues are resolved. This automatic communication flow helps improve customer satisfaction, reduces delays, and eliminates the need for follow-ups via phone or chat.

Moreover, since the email logic is integrated with Salesforce Flows and Apex code, the messages can be customized, tracked, and audited when needed — offering both transparency and reliability in operations.

---

## 3. Dashboards provide clear visibility for managers



□

Another major advantage of this project is the use of **interactive dashboards and custom reports**. Managers and team leads can access real-time data visualizations on key performance indicators such as:

- Number of active work orders
- Technician-wise job distribution
- Completion status
- Average resolution time
- Location-based service patterns

This visibility allows decision-makers to monitor the health of field operations at a glance. It also aids in resource planning, workload balancing, and identifying bottlenecks in the service process.

Custom dashboards such as the “WorkOrder Monitoring Dashboard” and “Technician Efficiency Dashboard” empower managers with actionable insights, which directly contributes to continuous improvement and operational efficiency.

---

#### 4. Scalable and cloud-based

Being built on the **Salesforce Platform**, the solution is inherently scalable and hosted on the cloud. This means:

- It can handle a growing number of technicians and work orders without performance degradation.
- There is no need for additional infrastructure or server management.
- Access is available from any device with internet connectivity, which is crucial for field service teams working in distributed environments.

In addition, Salesforce provides security, role-based access, and robust audit trails, ensuring the system is not only scalable but also secure and compliant.

□

The modular architecture — consisting of objects, flows, Apex logic, and dashboards — makes it easy to expand or integrate with other Salesforce or third-party services if needed in the future.

---

## Disadvantages

---

### 1. Requires Apex and Flow knowledge to customize

While Salesforce is a low-code platform, this particular solution relies heavily on custom Apex classes and Flows. Any change in business logic — for example, modifying technician assignment criteria or changing the email content — requires a solid understanding of:

- Apex programming
- Salesforce object model
- Flow Builder

This creates a dependency on technically skilled personnel or Salesforce developers. Non-technical users may find it challenging to make system changes without assistance, which may lead to delays in updates or minor fixes.

For organizations with limited Salesforce expertise, there may be a need to invest in training or ongoing technical support.

---

### 2. Depends fully on Salesforce platform

Although Salesforce is a powerful and reliable platform, this solution is **entirely dependent** on its infrastructure. This means:

- You cannot use this solution outside of Salesforce without major reengineering.
- Any limitations or restrictions from Salesforce (such as API limits, storage limits, or governor limits) directly impact the performance and scalability of your system.

□

- Licensing costs may be a concern for smaller organizations, especially if they need to scale the number of users or features.

In the event of downtime or scheduled maintenance by Salesforce, access to the system may be temporarily affected — although such events are rare due to Salesforce's high uptime standards.

---

### 3. May require optimization at high data volumes

As the number of work orders, technicians, and assignments increases over time, the system may require **periodic performance optimization**. This is especially relevant for:

- Apex triggers that run on record insert/update
- Batch jobs that clean up old records
- Report filters and dashboard refresh rates

Although the initial implementation is optimized for moderate workloads, a large-scale rollout (e.g., across multiple cities or departments) would require tuning — including indexing fields, monitoring governor limits, and possibly refactoring logic for better performance.

Without optimization, the system may experience slower response times or hit resource limits, which could impact usability and reliability.

While the Field Service WorkOrder Optimization project brings numerous benefits to organizations managing on-field service teams, it's essential to acknowledge the challenges involved. With thoughtful implementation, ongoing monitoring, and periodic optimization, the system can deliver long-term efficiency, transparency, and customer satisfaction — making it a valuable asset for modern field service operations.

## 9. Conclusion

The **Field Service WorkOrder Optimization** project successfully demonstrates how automation and data-driven workflows can transform traditional field service operations. By leveraging Salesforce's robust features—including **Apex logic, Flows, Dashboards, and Reports**—the system eliminates the need for manual technician assignments, improves communication through real-time email notifications, and enhances visibility for managers through interactive dashboards.

The solution ensures that technicians are assigned efficiently based on location, skills, and availability, while customers stay informed throughout the process. Managers can track performance metrics and work order statuses, which helps with planning and continuous improvement.

Overall, the project not only reduces administrative overhead but also increases the speed and accuracy of field operations. It serves as a scalable and intelligent model for organizations seeking to improve their service delivery using cloud-based platforms like Salesforce.