NAAN MUDHALVAN SALESFORCES PROJECT

**A CRM Application to Manage the Mall**

TEAM MEMBERS

B. ARCHANA

K.S. BHARATHI

Y. MOMINA BEE

S. SUMITHA





**1. Project Overview : CRM Application for Mall Management**

This project is focused on developing a **CRM Application for Mall Management**, designed to address the primary challenge of **streamlining and enhancing the management of commercial malls**. The goal is to deliver a comprehensive solution by leveraging the **Salesforce platform**. Through this project, we aim to enhance **operational efficiency**, **tenant and visitor experience**, and **data accuracy** to support the long-term goals of effective mall management.

# 2. Objectives :

# To develop a comprehensive CRM application to streamline and enhance the management of commercial malls, focusing on lease tracking, tenant interactions, and actionable insights for decision-making.

# Here are some specific and measurable goals that the CRM application intends to achieve:

# 1. Improve Lease Management Efficiency:

# - Reduce the time spent on manual lease tracking by 50% within the first six months.

# - Automate lease renewal reminders for 100% of the tenants.

# 2. Enhance Tenant Communication and Satisfaction:

# - Achieve a 90% tenant satisfaction rate through improved communication channels within the first year.

# - Resolve 95% of tenant issues within 24 hours by utilizing the issue tracking system.

# 3. Boost Sales and Revenue Tracking Accuracy:

# - Integrate sales data from 100% of tenant point-of-sale systems within the first quarter.

# - Generate detailed revenue reports with 99% accuracy on a monthly basis.

# 4. Optimize Visitor Management:

# - Register 100% of mall visitors digitally and track their flow within the first year.

# - Increase visitor engagement through targeted marketing campaigns by 20% within six months.

# 5. Streamline Maintenance and Facility Management:

# - Schedule and complete 95% of maintenance tasks on time within the first six months.

# - Reduce incident resolution time by 50% through efficient incident reporting and tracking.

# 6. Enhance Data Analytics and Reporting Capabilities:

# - Develop and deploy comprehensive dashboards for real-time insights within the first three months.

# - Deliver 90% of custom reports requested by mall management within 24 hours.

# These goals are designed to be specific, measurable, achievable, relevant, and time-bound (SMART) to ensure the project's success and alignment with the mall's overall objectives

**Business Goals**:

* Tenant Object:

The Tenant Object in Salesforce will serve as the core repository for all tenant-related information. This includes contact details, lease agreements, payment history, and any communication logs. Key attributes might include:

Tenant ID: A unique identifier for each tenant.

Tenant Name: The name of the tenant (individual or business).

Contact Information: Email, phone number, and address.

Lease Details: Associated lease agreements, lease start and end dates, and renewal status.

Payment History: Records of all payments made by the tenant.

Issues and Feedback: Logs of any reported issues or feedback provided by the tenant.

* Lease Tracking

This feature automates and simplifies the management of lease agreements. Key functionalities include:

Automated Reminders: Notifications for upcoming lease renewals or expirations.

- \*\*Digital Documentation\*\*: Secure storage of all lease agreements in digital format, ensuring easy access and retrieval.

- \*\*Lease Status Tracking\*\*: Real-time updates on the status of leases—active, expired, or pending renewal.

- \*\*Customizable Templates\*\*: Pre-built templates for lease agreements that can be tailored to specific needs.

* Tenant Issues

Efficient management of tenant issues is crucial for maintaining tenant satisfaction and operational efficiency. This component includes:

- Issue Reporting: Tenants can easily report issues through a dedicated portal or app.

- Issue Tracking: A system to log, monitor, and resolve reported issues.

- Communication Logs: Keep track of all communication with tenants regarding their reported issues.

- Resolution Workflows: Automated workflows to assign and manage resolution tasks, ensuring timely and efficient resolution of issues.

- Feedback Collection: Post-resolution feedback to gauge tenant satisfaction and identify areas for improvement.

These components work together to create a cohesive and efficient system for managing tenant relationships, lease agreements, and any issues that arise, ensuring smooth operations and high tenant satisfaction.

**Specific Outcomes**: CRM Application for Mall Management

Here are the key deliverables and outcomes expected from this project:

1. Automated Lease Management System:

- An automated system to track lease agreements, send renewal reminders, and store digital lease documents securely.

- Expected Outcome: \*\*50% reduction in manual lease tracking tasks\*\* and timely renewal processes for 100% of leases.

2. Enhanced Tenant Communication Portal:

- A centralized portal for tenants to access their lease details, report issues, and communicate with management.

- Expected Outcome:”90% tenant satisfaction rate “ through improved communication and faster issue resolution.

3. Integrated Sales Data Tracking:

- Real-time integration of sales data from tenant point-of-sale systems to monitor performance.

- Expected Outcome: “20% increase in revenue accuracy” and detailed performance reports available monthly.

4. Digital Visitor Registration System:

- A system for digital registration and tracking of visitors, including visitor flow analytics.

- Expected Outcome: “100% digital registration of visitors” and actionable insights to improve visitor experience and engagement.

5. Streamlined Maintenance and Facility Management:

- Tools for scheduling and tracking maintenance tasks, managing vendors, and resolving incidents.

- Expected Outcome: “50% reduction in incident resolution time” and timely completion of 95% of maintenance tasks.

6. Comprehensive Analytics and Reporting Dashboard:

- Dashboards and reports providing real-time insights and predictive analytics.

- Expected Outcome: “Informed decision-making"with 90% of custom reports delivered within 24 hours and improved operational transparency.

By achieving these specific outcomes, the CRM application will significantly enhance the efficiency, tenant satisfaction, revenue generation, and overall management of the mall. If there's a particular deliverable you'd like to explore further, let me know!

**3. Salesforce Key Features and Concepts Utilized**

In developing the CRM application for mall management, the project leverages several key features and concepts within the Salesforce platform:

1. Custom Objects:

- Creation of custom objects such as Tenant, Lease, and Issue to store and manage specific data related to mall management.

2. Workflows and Automations:

- Workflow Rules: Automate routine tasks like sending lease renewal reminders and notifying managers of tenant issues.

- Process Builder: Streamline complex business processes by defining triggers and actions for tasks such as lease management and maintenance scheduling.

- Flow: Create more advanced automations with Salesforce Flow to guide users through processes and update records.

3. Salesforce Lightning Experience:

- Utilize the Lightning Experience to provide a modern, user-friendly interface for mall management staff, improving usability and productivity.

4. Reports and Dashboards:

- Develop custom reports and dashboards to provide real-time insights into tenant performance, lease status, visitor analytics, and more.

- Use *Einstein Analytics* for advanced data visualization and predictive analytics.

5. Service Cloud:

- Implement Service Cloud to manage tenant service requests, track issue resolution, and improve overall tenant satisfaction.

- Utilize \*\*Omni-Channel Routing\*\* to ensure that tenant issues are directed to the appropriate personnel for timely resolution.

6. Community Cloud:

- Build a tenant portal using Community Cloud, enabling tenants to access their lease information, report issues, and communicate with mall management.

7. AppExchange Solutions:

- Integrate third-party solutions from the Salesforce AppExchange to extend the functionality of the CRM application, such as digital signature tools for lease agreements.

8. Mobile Capabilities:

- Enable access to the CRM application through Salesforce's mobile app, allowing mall management and staff to perform tasks and access information on-the-go.

9. Security and Data Management:

- Implement robust security measures using Salesforce's security features, including role-based access controls, encryption, and data backups to protect sensitive information.

10. Integration Capabilities:

- Leverage Salesforce's integration capabilities to connect with external systems such as point-of-sale systems for real-time sales data, and IoT devices for facility management.

These features and concepts collectively provide a powerful and flexible solution for managing all aspects of a commercial mall, driving efficiency, improving tenant relationships, and enabling data-driven decision-making. If there's any specific feature you'd like more details on, feel free to ask!

# 4. Detailed Steps to Solution Design

Here is a comprehensive approach to the solution design for the CRM Application to Manage the Mall using Salesforce:

1. Data Models

The data models represent the structure of data stored within Salesforce. They include custom objects, fields, and relationships.

1. Tenant Object (Fields):

- Tenant ID (Auto Number

- Tenant Name (Text)

- Contact Information (Phone, Email, Address)

- Lease Details (Lookup to Lease Object)

- Payment History (Related List)

- Issues Reported (Related List)

*Relationships:*

- One-to-Many with Lease Object

- One-to-Many with Issue Object

2. Lease Object (Fields):

- Lease ID (Auto Number)

- Lease Start Date (Date)

- Lease End Date (Date)

- Renewal Status (Picklist: Active, Expired, Pending Renewal)

- Tenant (Lookup to Tenant Object)

- Documentation (Files)

*Relationships:*

- Many-to-One with Tenant Object

3. \*\*Issue Object (Fields):

- Issue ID (Auto Number)

- Issue Description (Text Area)

- Reported Date (Date)

- Resolution Status (Picklist: Open, In Progress, Resolved, Closed)

- Tenant (Lookup to Tenant Object)

- Assigned To (Lookup to User)

*Relationships:*

- Many-to-One with Tenant Object

- Many-to-One with User (Assignee)

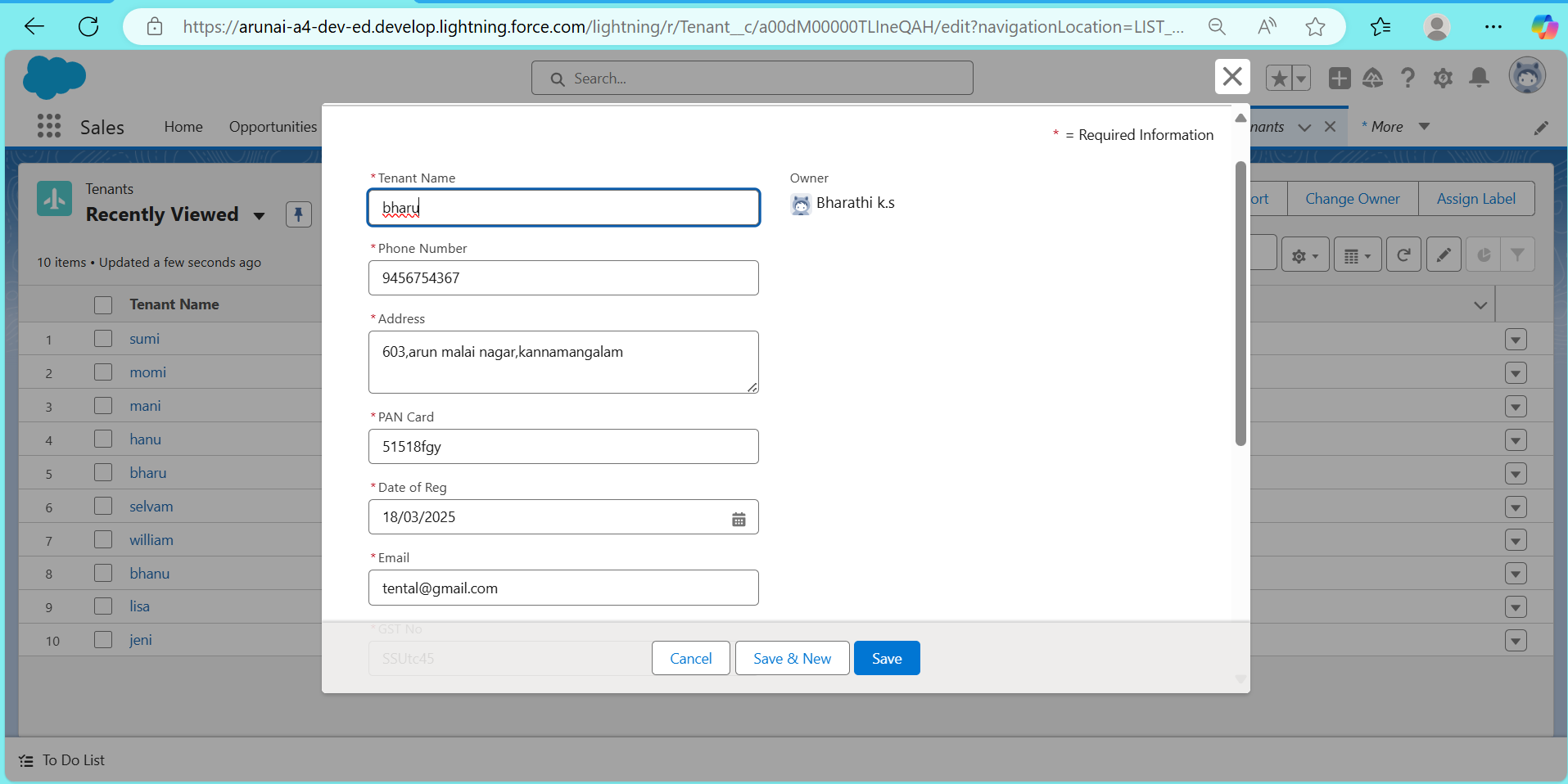
2. User Interface Designs

User interfaces should be intuitive and accessible, ensuring users can easily navigate the application. Below are mockups of key screens.

1. Tenant Management Screen:

- Displays a list of tenants with search and filter capabilities.

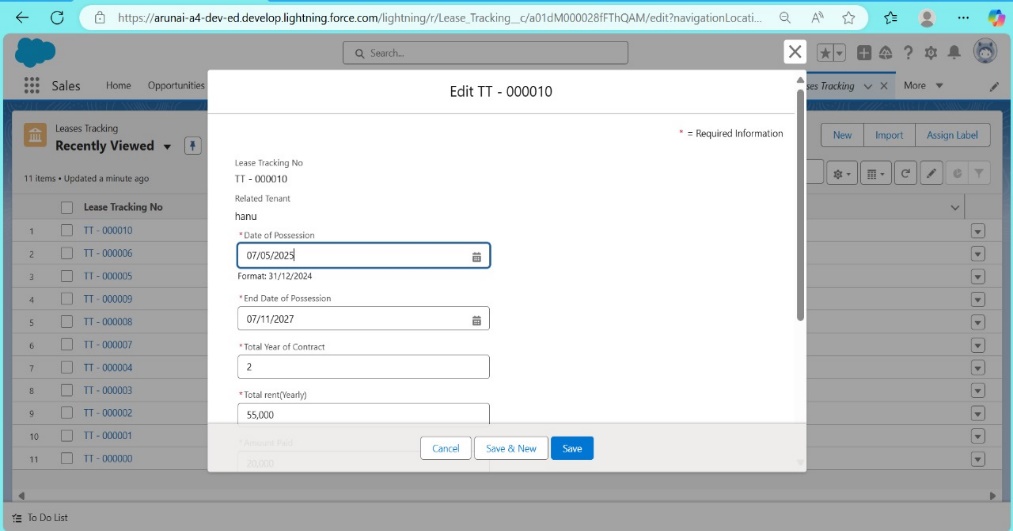
- Allows access to detailed tenant profiles, lease details, payment history, and reported issues.



2. Lease Tracking Screen:

- Provides a visual timeline of lease start and end dates.

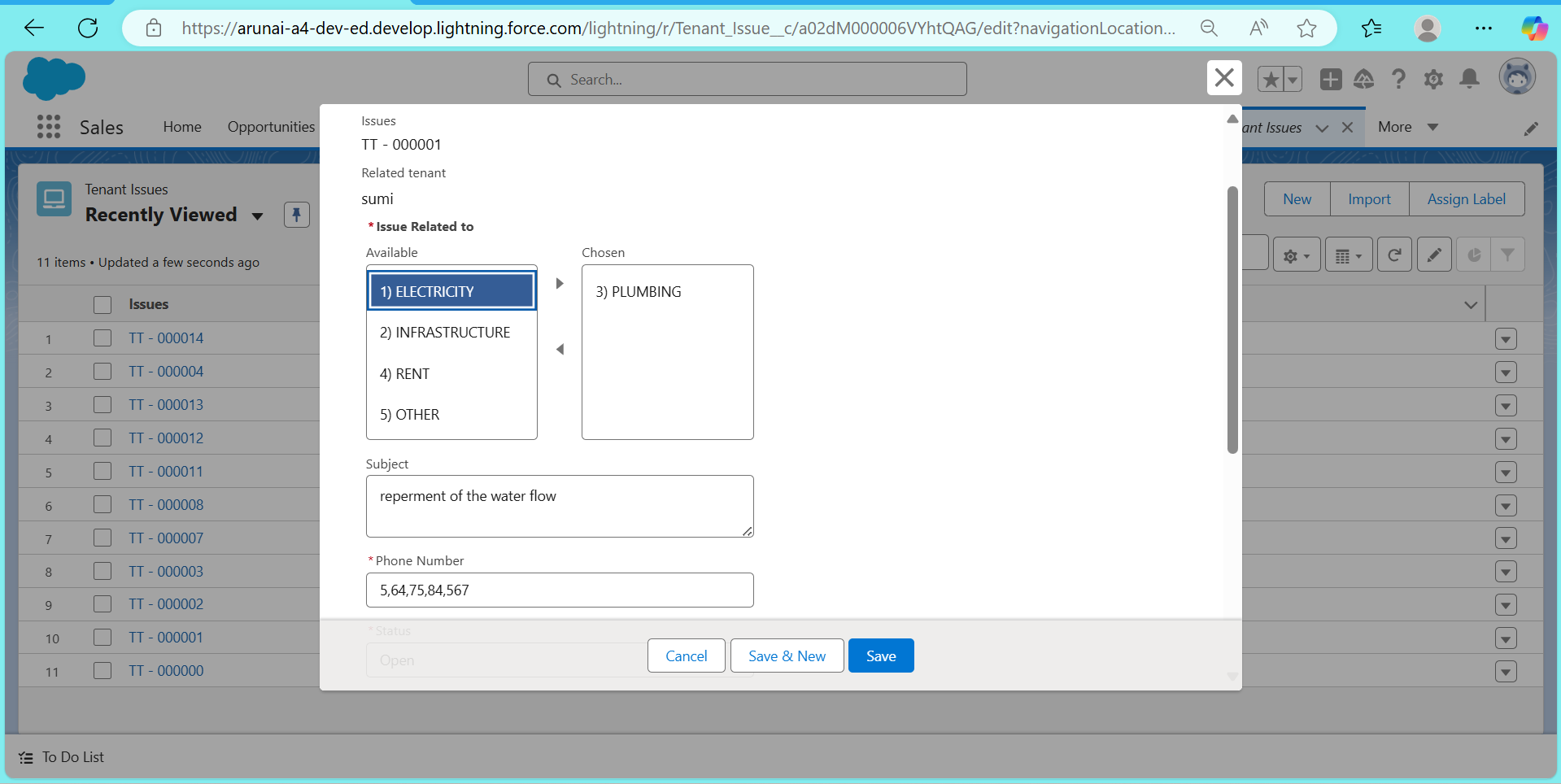
- Highlights leases pending renewal and those nearing expiry.



3. Issue Management Screen:

- Displays a list of reported issues with status filters.

- Allows tracking of issue resolution progress.



3. Business Logic

The business logic includes workflows, process automation, and custom functionality to streamline operations.

1. Automated Lease Reminders:

- Workflow Rule: Trigger an email reminder for lease renewals 30 days before expiry.

- Process Builder: Update lease status automatically based on renewal actions.

2. Issue Tracking and Resolution:

- Workflow Rule: Notify the assigned user when a new issue is reported.

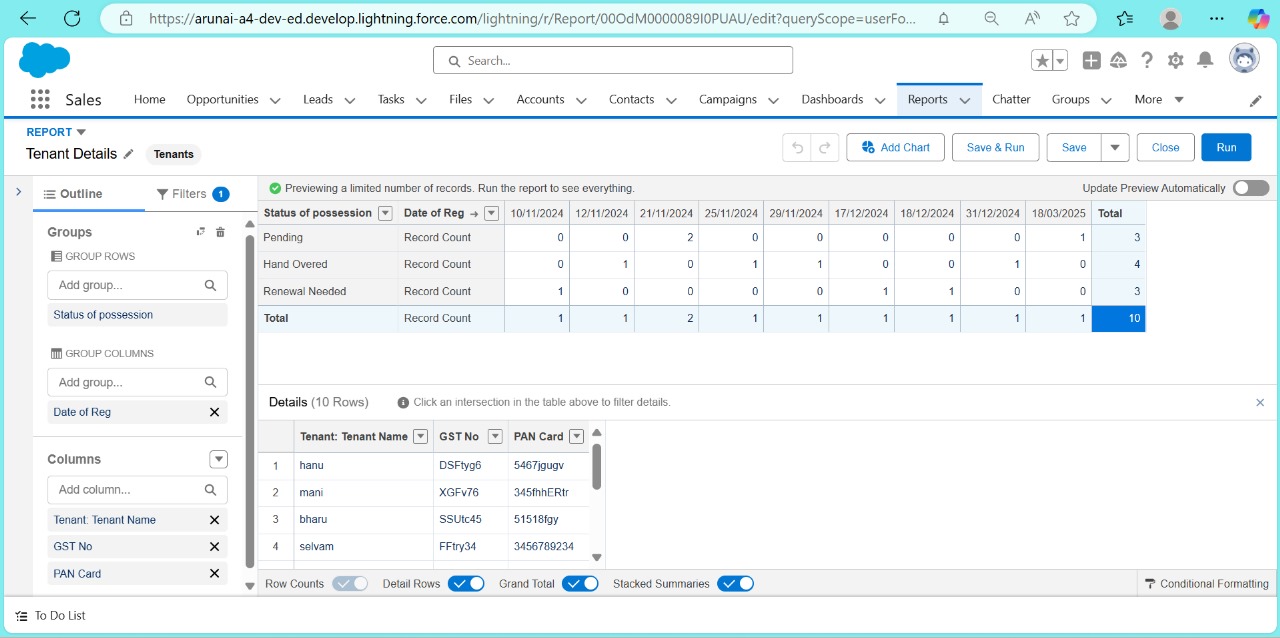
- Salesforce Flow: Guide users through the issue resolution process, updating status and logging actions.

3. Tenant Communication:

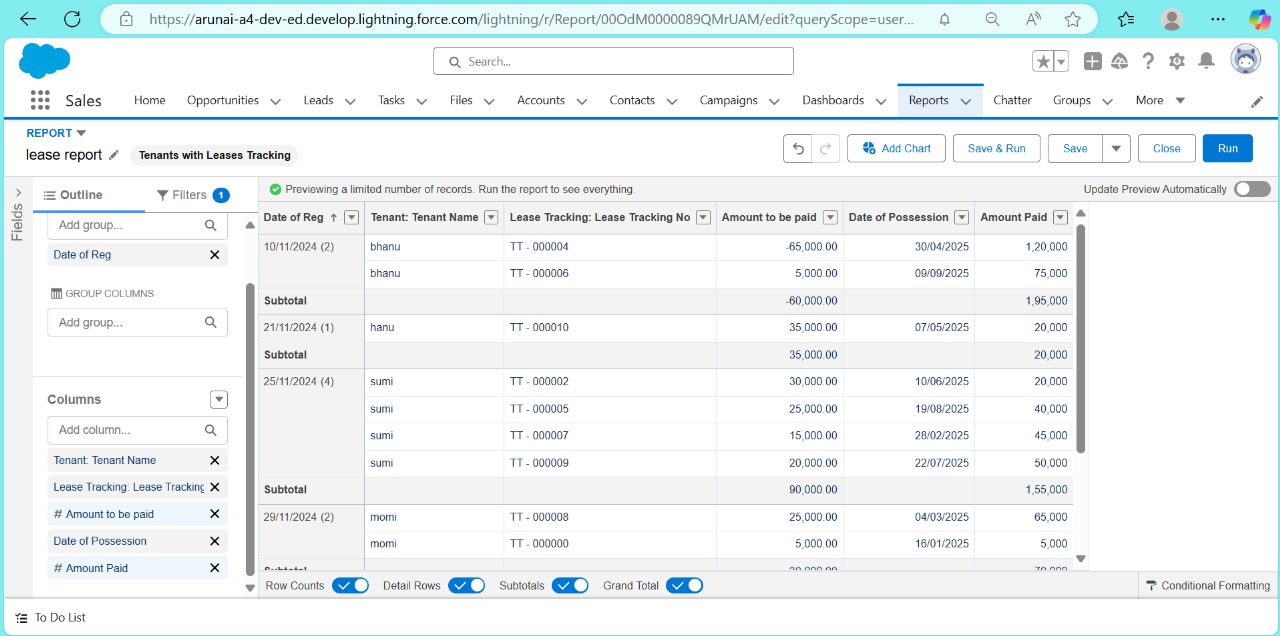
- Automation: Send periodic updates and important notices to tenants via email and SMS.

- Custom Apex Code: Implement complex business logic for specific scenarios that require customization beyond standard tools.

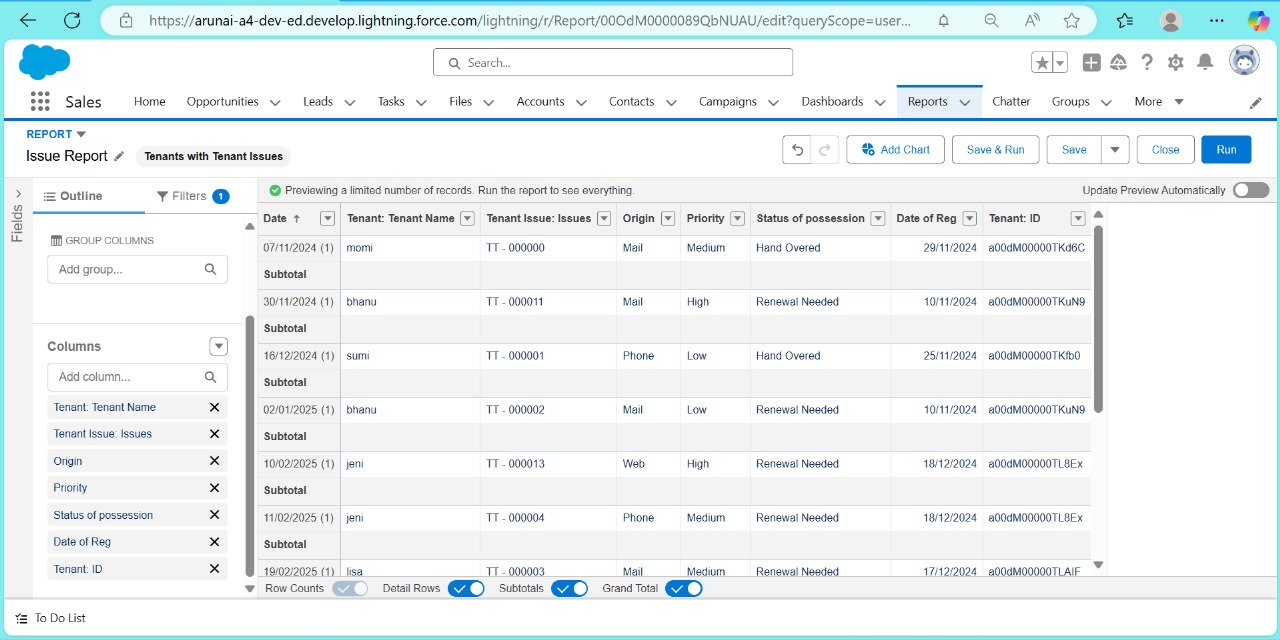
1. Tenant Management Screen:



2. Lease Tracking Screen:



3. Issue Management Screen:



This documentation provides a structured approach to designing the CRM application, ensuring all elements are meticulously planned and executed. 5. Testing and Validation:

# Testing and validation are critical to ensuring that the CRM application for mall management functions as intended and meets all requirements. Here’s a comprehensive approach to the different types of testing:

# Unit Testing (Apex Classes, Triggers)

# Unit testing involves testing individual components of the application to ensure they perform correctly. In Salesforce, this typically involves Apex classes and triggers.

# 1. Apex Classes:

# - Purpose: Validate that the logic written in Apex classes works as expected.

# - Approach:

# - Write test methods in Apex that cover various scenarios, including edge cases.

# - Use the `@isTest` annotation to create test classes.

# - Ensure that tests cover at least 75% of the code (code coverage requirement by Salesforce).

# 2. Triggers:

# - Purpose: Ensure that the triggers execute correctly and efficiently.

# - Approach:

# - Write test methods that simulate the conditions under which the trigger runs.

# - Test various scenarios including insert, update, and delete operations.

# - Validate the expected outcomes by using assertions in the test methods.

# User Interface Testing

# User Interface (UI) testing ensures that the user-facing elements of the application work correctly and provide a good user experience.

# 1. Manual Testing:

# - Purpose: Verify that the UI components work as expected and are intuitive to use.

# - Approach:

# - Create a detailed test plan covering all UI elements.

# - Perform manual testing to validate the functionality, including clicking buttons, entering data, and navigating through the application.

# - Use a checklist to ensure all scenarios are covered.

# 2. Automated UI Testing:

# - Purpose: Automate repetitive testing tasks to ensure consistency and efficiency.

# - Approach:

# - Use tools like Selenium for automated browser testing.

# - Write test scripts to simulate user interactions and validate the outcomes.

# Validation and User Acceptance Testing (UAT)

# 1. Validation:

# - Conduct thorough validation to ensure all functionalities meet the specified requirements.

# - Use feedback from stakeholders to refine and improve the application.

# 2. User Acceptance Testing (UAT):

# - Engage end-users in testing the application to ensure it meets their needs.

# - Collect feedback and make necessary adjustments before the final deployment.

# By following this structured approach to testing and validation, the CRM application will be thoroughly vetted for reliability, functionality, and user satisfaction. public class tenantschedulable implements Schedulable

# {

# public void execute(Schedulablecontext sc)

# {

# list<Tenant\_c> ten = [SELECT Id, Status\_of\_Possessionc FROM Tenant\_c ];

# list<Tenant\_c> tenantstodelete = New List<Tenant\_c>();

# 

# for(Tenant\_\_c te: ten)

# {

# if(te.Status\_of\_Possession\_\_c == 'Closed')

# {

# tenantstodelete.add(te);

# }

# }

# Delete tenantstodelete;

# }

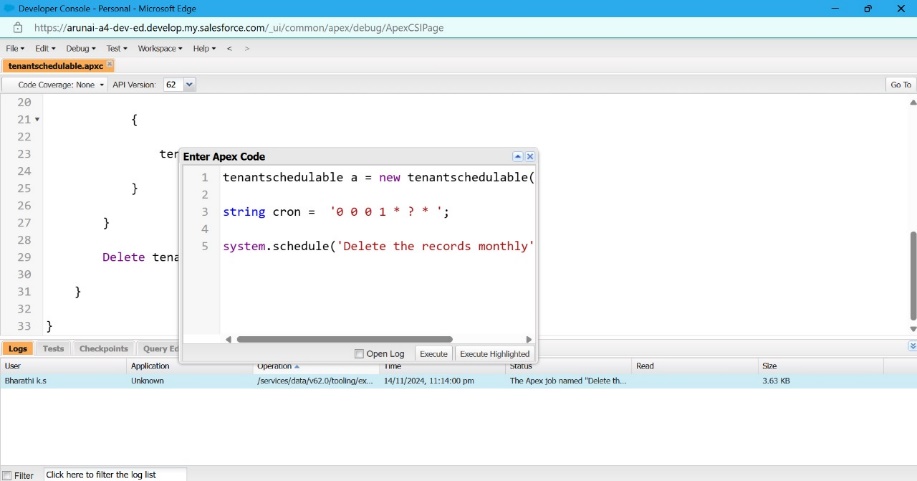
# }

# 

# tenantschedulable a = new tenantschedulable();

# string cron = '0 0 0 1 \* ? \* ';

# system.schedule('Delete the records monthly', cron, a);



## 6. Key Scenarios Addressed by Salesforce in the Implementation Project

This gives clarity that you are addressing various use cases or situations that Sale### Testing and Validation Approach

Effective testing and validation ensure that the CRM application performs reliably and meets all specified requirements. Here is a comprehensive approach to testing both Apex triggers and Asynchronous Apex:

LEASETRACKINGTRIGGER

apex class

public class LeaseTrackingTriggerHandler

{

public static void method1(List<Lease\_Tracking\_\_c> lt1)

{

for(Lease\_Tracking\_\_c lt2: lt1 )

{

if(lt2.Amount\_to\_be\_paid\_\_c > (lt2.Total\_rent\_Yearly\_\_c)/2)

{

Messaging.SingleEmailMessage M = New Messaging.SingleEmailMessage();

List<String> ToADD = New List<String>{lt2.Email\_id\_\_c};

M.setToAddresses(ToADD);

M.setSubject('Regarding the Pending Rent');

M.setPlainTextBody('Hello, This is an Reminder for you to complete your due rent by the end this month, your due rent thatneeds to be paid is' +lt2.Amount\_to\_be\_paid\_\_c);

List<Messaging.Email> AB = New List<Messaging.Email>{}

AB.add(M)

Messaging.sendEmail(AB);

}

}

}

}

Apex triggering

trigger leasetrackingtrigger on Lease\_Tracking\_\_c (After insert,After update) {

if(Trigger.isAfter && Trigger.IsUpdate)

{

LeaseTrackingTriggerHandler.method1(trigger.old);

}

}

Purpose: Ensure asynchronous processes such as Future Methods, Batch Apex, and Scheduled Apex function correctly and handle large data volumes efficiently.

1. Future Methods:

- Testing Approach: Verify that future methods execute correctly and complete their intended tasks.

- Example: Test asynchronous email notifications.

```apex

@isTest

private class TenantServiceTest {

@isTest static void testSendEmailNotifications() {

Tenant\_\_c tenant = new Tenant\_\_c(Name = 'John Doe', Email\_\_c = 'johndoe@example.com');

insert tenant;

Test.startTest();

TenantService.sendEmailNotifications(new List<Id>{tenant.Id});

Test.stopTest();

// Verify email logic or outcome

}

}

```

2. Batch Apex:

- Testing Approach: Ensure batch processing handles large data sets and completes successfully.

- Example: Test batch processing to update tenant statuses.

```apex

@isTest

private class TenantBatchUpdateTest {

@isTest static void testBatchUpdate() {

Tenant\_\_c tenant1 = new Tenant\_\_c(Name = 'John Doe', Status\_\_c = 'Pending');

Tenant\_\_c tenant2 = new Tenant\_\_c(Name = 'Jane Doe', Status\_\_c = 'Pending');

insert new List<Tenant\_\_c>{tenant1, tenant2};

Test.startTest();

Database.executeBatch(new TenantBatchUpdate());

Test.stopTest();

Tenant\_\_c updatedTenant1 = [SELECT Status\_\_c FROM Tenant\_\_c WHERE Id = :tenant1.Id];

Tenant\_\_c updatedTenant2 = [SELECT Status\_\_c FROM Tenant\_\_c WHERE Id = :tenant2.Id];

System.assertEquals('Active', updatedTenant1.Status\_\_c);

System.assertEquals('Active', updatedTenant2.Status\_\_c);

}

}

```

3. Scheduled Apex:

- Testing Approach: Verify scheduled jobs run at specified times and complete tasks.

- Example: Test scheduled job for monthly tenant record deletion.

```apex

@isTest

private class DeleteClosedTenantsSchedulerTest {

@isTest static void testScheduledJob() {

Tenant\_\_c tenant = new Tenant\_\_c(Name = 'John Doe', Possession\_Status\_\_c = 'Closed');

insert tenant;

Test.startTest();

// Simulate scheduling the job

DeleteClosedTenantsScheduler job = new DeleteClosedTenantsScheduler();

String CRON\_EXP = '0 0 0 1 \* ?'; // Run on the first day of the month

System.schedule('Delete Closed Tenants Job', CRON\_EXP, job);

Test.stopTest();

// Verify tenants with 'Closed' status are deleted

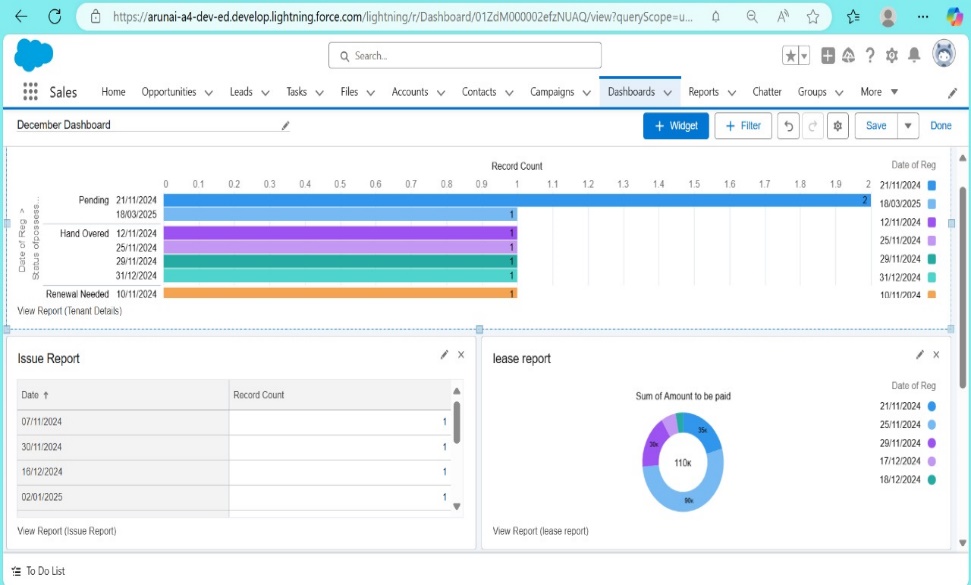
System.assertEquals(0, [SELECT COUNT() FROM Tenant\_\_c WHERE Possession\_Status\_\_c = 'Closed']);

}

}

```

### Conclusion

By thoroughly testing both Apex triggers and asynchronous Apex processes, you can ensure that the CRM application is reliable, efficient, and provides a seamless user experience. 

7. Conclusion

CRM allows you to oversee relationships with your customers and prospects and track information identified with the majority of your connections. It additionally helps groups team up, both inside and

remotely, accumulate experiences from online networking, track essential measurements, and convey

by means of email, telephone, social, and different channels. It helps organizations to anticipate customer

needs based on their past requirements and choices.



