

VIT-AP
UNIVERSITY

**Property Management Application
using Salesforce**

SPJ 2001

SUMMER INTENSHIP - PROJECT FIRST REVIEW

Guided by: Prof. Nihar Ranjan Pradhan

TEAM MEMBERS

CHILUKA SAKETH – 20MIS7006
M.SURYA GOPAL – 20MIS7097
K.V.K.R.NANDINI – 20MIS7002
NAGELLA SAKETH – 20MIS7078

1.1 ABSTRACT:

The aim of our project is to develop a comprehensive software solution that streamlines and automates various aspects of property management processes. The application will leverage the capabilities of the Salesforce platform to provide a robust and scalable solution for property managers and real estate professionals.

1.2 INTRODUCTION:

It is a software solution developed using Salesforce to help property managers and real estate professionals manage their properties more efficiently. It simplifies tasks such as listing properties, managing tenants and leases, tracking maintenance, and handling finances. By automating these processes, the application saves time and reduces errors. It also provides valuable insights through reporting and analytics, allowing property managers to make informed decisions. Overall, the Property Management Application streamlines property management operations and improves the experience for both property managers and tenants.

1.3 PROBLEM IDENTIFICATION:

The property management industry faces challenges such as manual processes, scattered information, limited tenant engagement, complex lease management, inefficient maintenance processes, and limited financial insights. These issues result in inefficiencies, errors, tenant dissatisfaction, and difficulties in decision-making. The Property Management Application using Salesforce addresses these problems by automating processes, centralizing information, improving tenant communication, simplifying lease management, streamlining maintenance workflows, and providing comprehensive financial insights.

2. SCOPE:

The scope of the Property Management Application project encompasses the development, implementation, and deployment of a comprehensive software solution using Salesforce. The project aims to address various aspects of property management, including property listing and marketing, tenant and lease management, maintenance and work order tracking, financial management, and reporting.

Financial management capabilities will be incorporated, allowing property managers to track rent payments, manage expenses, generate financial reports, and integrate with accounting systems. Reporting and analytics functionalities will provide insights into property performance, occupancy rates, rental income, and expenses, enabling data-driven decision-making.

The key features of the Property Management Application project using Salesforce include:

2.1 Usability and User Experience: The application should prioritize ease of use and provide a seamless user experience for property managers, tenants, and maintenance staff. It should have an intuitive interface, clear navigation, and well-organized features. User-centric design principles should be employed to ensure that users can efficiently perform tasks and access the information they need.

2.2 Scalability and Flexibility: The application should be designed to accommodate the growing needs of property management businesses. It should have the ability to handle a large volume of properties, tenants, and maintenance requests. Additionally, it should be flexible enough to adapt to changes in business requirements, such as adding new features or integrating with external systems.

2.3 Data Security and Privacy: The application should prioritize the security and privacy of user data. Robust security measures, such as encryption, secure authentication, and role-based access control, should be implemented to protect sensitive information. Compliance with relevant data protection regulations, such as GDPR or CCPA, should be ensured to maintain user trust and adhere to legal requirements.

3. FUNCTIONAL REQUIREMENTS:

3.1 Property Listing and Marketing: The application allows property managers to create and manage property listings, including details like descriptions, images, pricing, and availability. It provides marketing capabilities to reach potential tenants or buyers through various channels.

3.2 Tenant and Lease Management: The application facilitates the management of tenant information, lease agreements, rent collection, and lease renewals. It automates rent reminders, tracks payment history, and generates reports related to lease terms and financials.

3.3 Maintenance and Work Order Management: The application enables property managers to track and manage maintenance requests, work orders, and service vendors. It streamlines communication and coordination between property managers, tenants, and maintenance staff for timely resolution of maintenance issues.

4. NON-FUNCTIONAL REQUIREMENTS:

4.1 Performance and Scalability: The application should be highly responsive and able to handle a large volume of users, data, and transactions. It should be optimized for speed and performance to ensure quick response times, even during peak usage periods. Additionally, the system should be designed to scale seamlessly as the user base or property portfolio grows.

4.2 Security and Privacy: The application should have robust security measures in place to protect sensitive data. This includes encryption of data at rest and in transit, secure authentication mechanisms, and role-based access control. Data privacy should be ensured by implementing appropriate measures to comply with relevant regulations and protect user information.

4.3 Reliability and Availability: The application should be reliable and available to users whenever they need it. It should have a high uptime, minimal downtime for maintenance, and a solid backup and disaster recovery strategy. Regular system monitoring and proactive measures should be in place to minimize potential disruptions and ensure continuous availability.

4.4 User Interface and User Experience: The application should have an intuitive and user-friendly interface that enables users to easily navigate and perform tasks. It should provide clear instructions, helpful error messages, and efficient workflows. Consistency in design and responsiveness across different devices and screen sizes is also important for a seamless user experience.

4.5 Integration and Interoperability: The application should support integration with external systems and platforms commonly used in the property management industry. This includes property listing websites, accounting software, payment gateways, or other third-party systems. It should have well-defined APIs or integration mechanisms to ensure smooth data exchange and interoperability.

4.6 Compliance and Regulation: The application should comply with relevant industry regulations and legal requirements. This may include data protection regulations (e.g., GDPR, CCPA), accessibility standards (e.g., WCAG), and any other applicable laws or regulations specific to the property management industry.

UML DIAGRAMS:

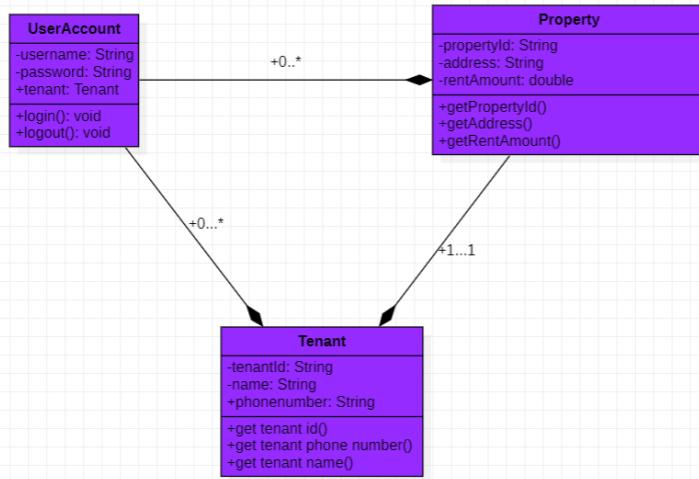
USE CASE DIAGRAM



- The "User" actor has access to the following use cases: User Login, View Properties, Apply for Property, Make Payment, Submit Maintenance Request, and User Registration.
- The "Admin" actor has access to the following use cases: Admin Login, Manage Properties, Create Tenant, Manage Maintenance Requests, Generate Reports, Manage Users, and System Configuration.
- By separating the use cases for the "User" and "Admin" actors, it distinguishes their respective roles and responsibilities within the property management application. Users can perform actions related to their own account, such as logging in, viewing properties, applying for properties, making payments, submitting maintenance requests, and registering as new users. On the other hand, administrators (admins) can perform additional administrative tasks such as logging in as an admin, managing properties, creating tenant profiles, managing maintenance

requests, generating reports, managing user accounts, and configuring the system.

CLASS DIAGRAM



- The "UserAccount" class has an optional association with the "Tenant" class, represented by a solid line with an open diamond. This means that a user account may or may not be associated with a tenant, and a tenant must be associated with a user account. The "UserAccount" class has a reference to the "Tenant" class through the "tenant" attribute.
- The "Tenant" class has a composition relationship with the "UserAccount" class, represented by a solid line with a filled diamond. This means that each tenant belongs to a single user account, and the tenant cannot exist without a user account. The "Tenant" class has a reference to the "UserAccount" class through the "userAccount" attribute.

These relationships illustrate the connection between user accounts, tenants, and properties in the property management application. A user account can be associated with a tenant, and each tenant is associated with a user account. Additionally, each property is associated with a tenant.

LITERATURE SURVEY

INTRODUCTION

- Provide an overview of property management applications and their significance in the real estate industry. Discuss the challenges faced by property managers and the need for efficient and automated solutions.
- Introduce Salesforce as a leading cloud-based CRM platform widely used in various industries for developing business applications.
- Highlight the potential benefits of leveraging Salesforce for property management, such as improved collaboration, scalability, and data security.

PROPERTY MANAGEMENT APPLICATION FEATURES

- Identify and describe the key features and functionalities commonly found in property management applications built on the Salesforce platform. This may include modules for tenant management, lease tracking, document management, maintenance requests, financial reporting, and analytics.
- Discuss how these features streamline property management processes, enhance data accuracy and accessibility, and enable better decision-making for property managers.

BENEFITS PROPERTY MANAGEMENT APP ON SALESFORCE

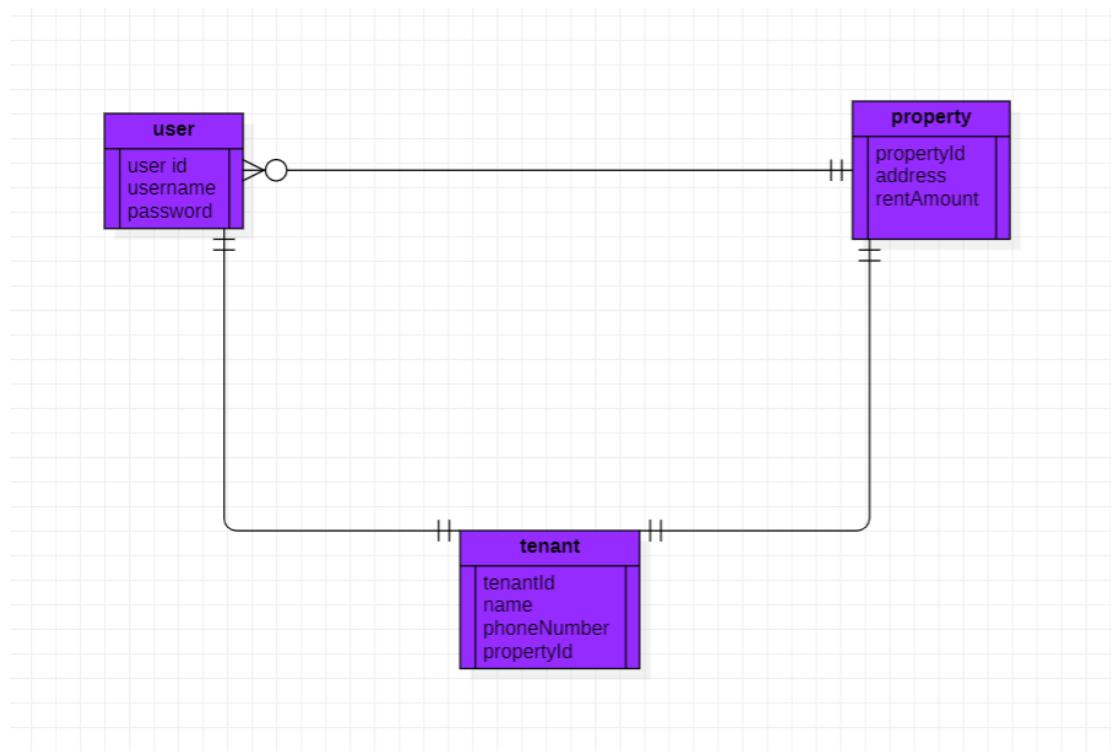
- Conduct a thorough analysis of the benefits offered by Salesforce-based property management applications.
- Discuss how these applications contribute to improved operational efficiency, cost savings, and time management by automating routine tasks, streamlining workflows, and providing real-time access to information.
- Highlight the impact of Salesforce's reporting and analytics capabilities on generating actionable insights and supporting data-driven decision-making in property management.

INTEGRATION AND CUSTOMIZATION

- Explore literature on integrating property management applications developed on Salesforce with other systems commonly used in the real estate industry, such as accounting software, marketing automation tools, or customer relationship management (CRM) platforms.
- Discuss the advantages of seamless data exchange between different systems, elimination of data silos, and improved collaboration across departments.
- Explore the customization capabilities of Salesforce, including the use of declarative tools and development resources, to tailor the property management application to specific business requirements and workflows.

DESIGN

ENTITY RELATIONSHIP MODEL



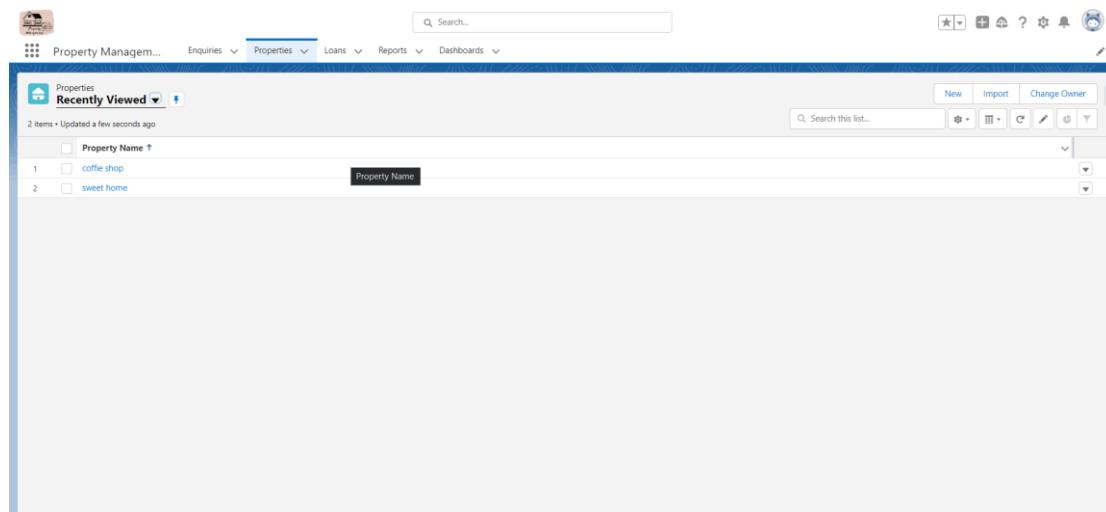
- The "User" entity represents a user account in the property management application. It has attributes such as userId (as the primary key), username, and password.

- The "Property" entity represents a property listing. It has attributes such as propertyId (as the primary key), address, rentAmount, and a foreign key userId referencing the "User" entity. This relationship indicates that a property is associated with a user (one-to-many relationship).
- The "Tenant" entity represents a tenant profile. It has attributes such as tenantId (as the primary key), name, phoneNumber, and a foreign key propertyId referencing the "Property" entity. This relationship indicates that a tenant is associated with a property (one-to-one relationship).

These relationships illustrate the connections between users, properties, and tenants in the property management application. Each user can have multiple properties, and each property can have a single tenant associated with it.

DESIGN AND ARCHITECTURE:

FRONT END: The front-end of the CRM application will focus on providing an intuitive and user-friendly interface for customers to interact with the system.



Property Management... Enquiries Properties Loans Reports Dashboards

Enquiries Recently Viewed

4 items • Updated a few seconds ago

	Customer Name
1	abhi
2	sai
3	keenthi
4	seetha

New Import Change Owner

Property Management... Enquiries Properties Loans Reports Dashboards

Loans Recently Viewed

2 items • Updated a few seconds ago

	Loan Id
1	LN-0002
2	LN-0001

New Import Change Owner

BACK END: The back-end architecture is responsible for processing and managing data, handling business logic, and integrating with external systems

Setup Home Object Manager

SETUP > OBJECT MANAGER Enquiry

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
City	City__c	Picklist	State	
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Lead Number	Lead_Number__c	Auto Number		
Owner	OwnerId	Lookup(User,Group)		✓
State	State__c	Picklist		

SETUP > OBJECT MANAGER			
Property			
Details		Page Layouts	
Fields & Relationships		PAGE LAYOUT NAME	CREATED BY
Page Layouts		Buy	SAKETH CHILUKA, 27/06/2023, 12:33 am
Lightning Record Pages		Property Layout	SAKETH CHILUKA, 26/06/2023, 1:51 pm
Buttons, Links, and Actions		Rent	SAKETH CHILUKA, 27/06/2023, 12:23 am
Compact Layouts			SAKETH CHILUKA, 27/06/2023, 12:24 am
Field Sets			
Object Limits			
Record Types			
Related Lookup Filters			
Search Layouts			
List View Button Layout			

SETUP > OBJECT MANAGER			
Loan			
Details		Fields & Relationships	
Fields & Relationships		4 Items, Sorted by Field Label	
Page Layouts		FIELD LABEL	FIELD NAME
Lightning Record Pages		Created By	CreatedBy
Buttons, Links, and Actions		Last Modified By	LastModifiedBy
Compact Layouts		Loan Id	Name
Field Sets		Owner	OwnerId
Object Limits			
Record Types			
Related Lookup Filters			

*****END*****