Ahmedabad University

CSE300 Software Engineering

Group 17

Real Estate Management System

Version : 1.0

Date: 02/11/2023

Submitted to: Prof. Khushru Doctor

S. No.	Name	Enrolment No.
1	Dev Bhalodia	AU2140208
2	Parva Parmar	AU21400083
3	Krishna Patel	AU2140170
4	Tirth Soni	AU2140183
5	Sahil Kotadiya	AU2040224

Table of Content

1. **Introduction** **page no**
   1.1 Overview……4
   1.2 Purpose of the SRS……..5
   1.3 Scope of the SRS……6
   1.4 Intended audience……7

2. **Description**
   2.1 Description……...8
   2.2 Objectives and goals……9
   2.3 Constraints……10

3. **Functional requirements**
   3.1 Functional requirements……...11
   3.2 Usecase Diagram……..12
   3.3 Sequence Diagram……13

4. **Non-functional requirements**
   4.1 Non-functional requirements……15

5. **External interface**……18

6. **User interface**……19

7. **System features**……..20

8. **Data requirements**……22

9. **Testing requirements**……24

10. **Performance requirements** ……27

11. Security requirements ……28

12. Documentation requirement ……..30

1. Introduction

1.1 Overview

Purpose:

The primary purpose of a real estate management system is to streamline, automate, and enhance the processes related to buying, selling, and managing properties. The software aims to make the management of listings, clients, and transactions more efficient.

Features:

1. **Property Listings Management:** Add, update, or delete property listings complete with photographs, videos, floor plans, and descriptions.
2. **Customer Relationship Management (CRM):** Maintain records of clients, buyers, sellers, renters, and landlords. Helps in tracking interactions, setting reminders, and following up.
3. **Transaction Management:** Monitor and manage all transactions including offers, agreements, and closings.
4. **Document Management:** Store and manage important documents such as agreements, title deeds, and tenant contracts.
5. **Financial Management:** Handle all financial aspects like invoicing, receipts, and ledgers. Some systems integrate with accounting software.
6. **Lease Management:** Manage rental properties, tenant details, lease agreements, and rental payment records.
7. **Reports & Analytics:** Generate reports on sales, listings, financial data, and other performance metrics.
8. **User Access Control:** Different levels of access for agents, managers, and administrators.
9. **Communication Tools:** Send emails or SMS notifications directly from the platform.
10. **Integration:** API support for integration with other platforms like MLS (Multiple Listing Service), payment gateways, and other business tools.

Benefits:

- **Efficiency:** Reduces paperwork and automates repetitive tasks.
- **Accuracy:** Minimizes manual data entry errors.
- **Accessibility:** Offers access from anywhere at any time.
- **Enhanced Customer Service:** Faster response times and better tracking of client interactions.
- **Scalability:** Suitable for both individual realtors and large real estate agencies.

Considerations:

- **Usability:** User-friendly interface for ease of use.
- **Customization:** Ability to customize features based on specific needs.
- **Security:** Ensure the software adheres to data protection standards, especially since it handles sensitive property and personal data.
- **Cost:** Evaluate cost-effectiveness based on features offered, size of the organization, and usage patterns.

1.2 Purpose and scope of SRS

Purpose of the SRS:

1. **Clear Communication:** The SRS serves as a bridge between stakeholders (developers, designers, project managers, clients, and end-users) ensuring everyone has a shared understanding of the system's functionalities and constraints.
2. **Foundation for Development:** It provides developers with a clear idea of what functions and features need to be implemented.
3. **Basis for Validation:** After implementation, the system can be validated against the SRS to ensure all requirements are met.
4. **Project Scope Management:** By defining the system's features and constraints, it helps in managing the scope of the project and prevents scope creep.

5. **Cost Estimation:** Based on the requirements laid out, it becomes easier to estimate the time, resources, and cost for the project.

Scope of the SRS:

1. **Functional Requirements:** Detailed description of all functionalities the system should possess. For this project, it will include things like property listings management, transaction management, CRM, and more.
2. **Non-functional Requirements:** Describes the performance attributes of the system, such as usability, reliability, scalability, and security.
3. **System Interfaces:** Describes how the software will interact with other software systems, hardware, or external systems.
4. **Data Management and Database Design:** Explains how the software will handle, store, and retrieve data.
5. **Constraints:** Lists any limitations or restrictions on the system's design or functionalities. This could be in terms of technology, budget, time, regulations, etc.
6. **User Requirements:** Details the needs of the end-users, and how the system will address those needs.
7. **Operational Scenarios:** Presents possible real-life scenarios of how the system will be used. This helps in understanding the flow of operations.
8. **Assumptions and Dependencies:** Lists any factors that the system's successful operation might depend on or any assumptions made while defining requirements.
9. **Acceptance Criteria:** Specifies the criteria that need to be met for the client to accept the software.
10. **Appendices:** Contains any other supplementary information, reference material, or detailed explanations that support the main content of the SRS but might be too detailed for the main sections.

1.3 Intended audience

- **Real Estate Agencies and Agents**: To keep track of their listings, manage clients, handle commissions, and maintain records.
- **Property Owners and Landlords**: For managing their properties, tracking tenant details, handling maintenance requests, and managing lease agreements.
- **Property Managers**: For overseeing multiple properties, handling tenant relations, and ensuring maintenance tasks are completed.
- **Real Estate Investors**: To manage their property portfolios, track income and expenses, and evaluate the performance of their investments.
- **Real Estate Developers**: To track the development and sale of new properties.
- **Government and Municipal Agencies**: To manage public housing units, collect property data, and ensure regulatory compliance.
- **Financial Institutions**: Like banks and mortgage lenders, to track property loans, property values, and risk assessments.
- **Legal Professionals**: To manage and access property-related legal documents and track ongoing property litigation.
- **Property Marketing and Advertising Firms**: For campaign management and tracking of property advertisements.
- **Potential Property Buyers**: To search for properties, compare features, schedule viewings, and access financing options.
- **Real Estate Appraisers and Valuation Experts**: To access property details and compare them with market trends to determine property values.
- **Property Insurers**: To evaluate risks associated with particular properties and to determine appropriate coverage levels.

2. Overall Description

2.1 Description

Real Estate Management System (REMS)

The Real Estate Management System (REMS) is an integrated digital platform designed to serve the multifaceted world of real estate. At its core, REMS seeks to simplify, automate, and optimize the operations of real estate professionals, offering a centralized solution that caters to various property-related needs.

Key Features:

- Property Listings Management: A dynamic module that allows realtors to effortlessly add, modify, and showcase properties with multimedia enhancements, ensuring that potential buyers or renters receive a comprehensive view of listings.
- Customer Relationship Management (CRM): This feature streamlines client communication, tracks interactions, and ensures timely follow-ups, fostering strong and lasting relationships between realtors and their clients.
- Transaction and Lease Management: An end-to-end solution for managing property sales, purchases, and rentals, ensuring that all parties have clarity on terms, agreements, and timelines.
- Financial Tools: A comprehensive suite that handles everything from invoicing to accounting integrations, making financial management a breeze.
- Reporting & Analytics: With just a few clicks, users can generate insightful reports on sales trends, property demands, client feedback, and more, empowering decision-makers with data-driven insights.

Target Audience:

REMS is meticulously crafted for a spectrum of users, including individual real estate agents, large real estate agencies, property managers, and even real estate

investors. Its intuitive design ensures ease of use, regardless of the user's technical proficiency.

Benefits:

Adopting REMS translates to a transformative experience for real estate professionals. Not only does it significantly diminish the need for cumbersome paperwork, but it also promises enhanced efficiency, real-time data accessibility, and a notable elevation in client satisfaction levels.

Integration & Compatibility:

REMS is built on a robust foundation using HTML, CSS, and JavaScript, ensuring a responsive and interactive user interface that's compatible across various browsers and devices. With PHP powering its backend operations, REMS offers reliable, server-side processing and database connectivity. This combination not only ensures swift and secure data transactions but also offers flexibility for potential future enhancements. Thanks to its web-based nature, users can access REMS from anywhere, provided they have an internet connection. Additionally, its architecture is primed for easy integration with other web services or platforms as the need arises, allowing the system to adapt and evolve in the dynamic realm of real estate management.

2.2 Goals and objectives

Goals:

1. **Streamlined Operations:** Provide an integrated platform to automate and optimize real estate operations, reducing manual efforts and enhancing efficiency.
2. **User-Centric Design:** Ensure an intuitive, user-friendly interface that caters to both tech-savvy and non-tech-savvy users.
3. **Data Centralization:** Create a single source of truth for all property-related data, making information access and management seamless.

4. **Improved Decision Making:** Offer real-time data analytics and reporting tools to assist stakeholders in making informed decisions.
5. **Enhanced Client Engagement:** Foster improved communication and relationship management with clients through the CRM module.

Objectives:

1. **Property Listings:** Allow users to add, edit, or delete property listings with multimedia support efficiently.
2. **Client Management:** Enable tracking of client interactions, setting reminders, and automating follow-ups.
3. **Transaction Oversight:** Provide tools to manage property sales, purchases, and leases, ensuring transparency and clarity.
4. **Financial Management:** Integrate financial tools for invoicing, payments, and accounting to maintain accurate financial records.
5. **Scalability:** Ensure the system can accommodate growth, both in terms of data volume and feature enhancements.

2.3 Constraint

1. **Platform Dependency:** Being web-based, the system requires an active internet connection for access.
2. **Technological Limitations:** The current system is built using HTML, CSS, JavaScript, and PHP. Any integration with platforms or technologies that aren't compatible with these might require additional efforts.
3. **Database Scalability:** Depending on the chosen database solution, there might be limits to how much data can be stored or how fast data can be accessed.

3. Functional Requirements

---

1. **User Registration & Profile Management**
    - **Justification**: Allows users to create profiles, manage personal details, and track their activities on the site.
2. **Property Listing Feature**
    - **Justification**: The main functionality of a real estate website is to allow agents or homeowners to list properties for sale or rent.
3. **Search & Filter Functionality**
    - **Justification**: Enables potential buyers or renters to find properties based on specific criteria such as location, price, number of bedrooms, etc.
4. **Property Details Page**
    - **Justification**: Provides comprehensive details about a property, including photos, descriptions, and amenities.
5. **Booking or Inquiry Form**
    - **Justification**: Allows interested parties to directly book a viewing or send an inquiry about a property.
6. **Ratings and Reviews for Properties/Agents**
    - **Justification**: Provides feedback mechanism for properties or agents, thereby increasing trust and transparency on the platform.
7. **Payment Gateway Integration (if required)**
    - **Justification**: Enables users to make payments for services like premium listings or other value-added services.

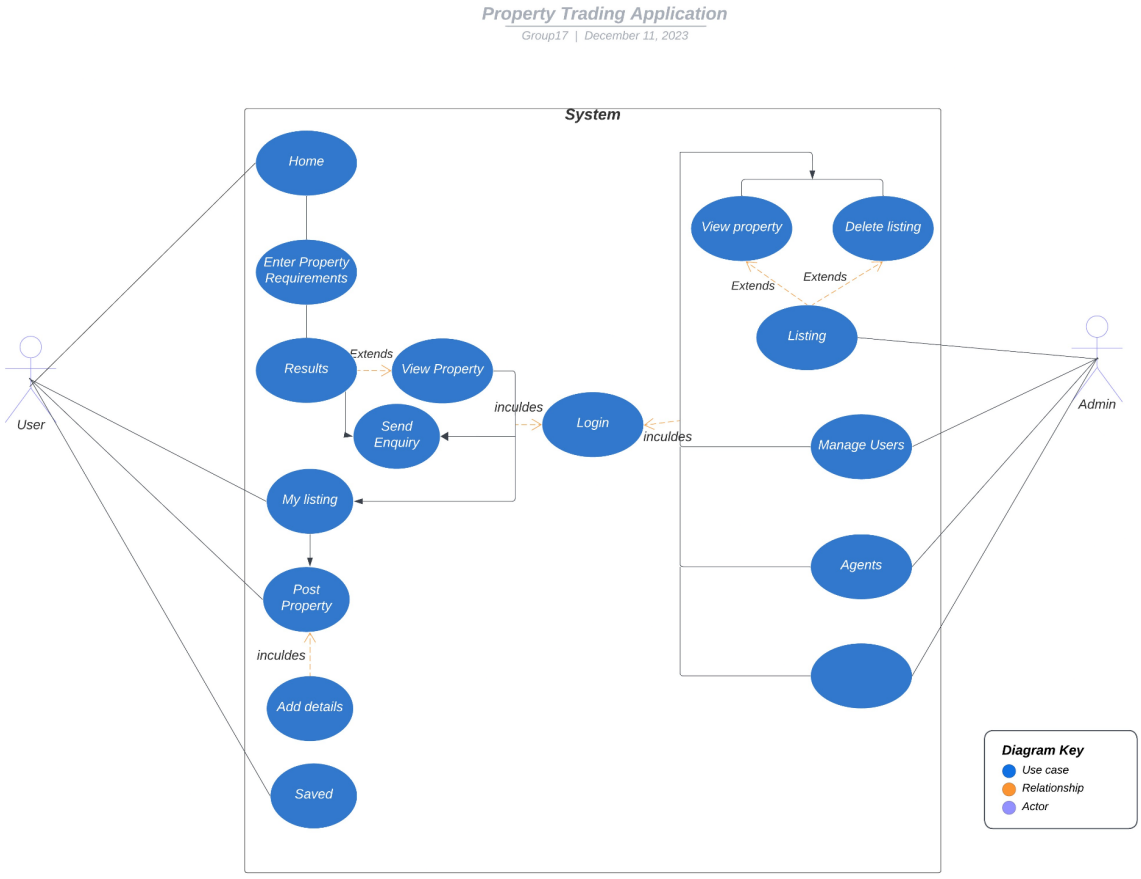
8. **Notification System**

- **Justification**: Keeps users informed about relevant properties, inquiries, booking confirmations, etc.

9. **Admin Dashboard**

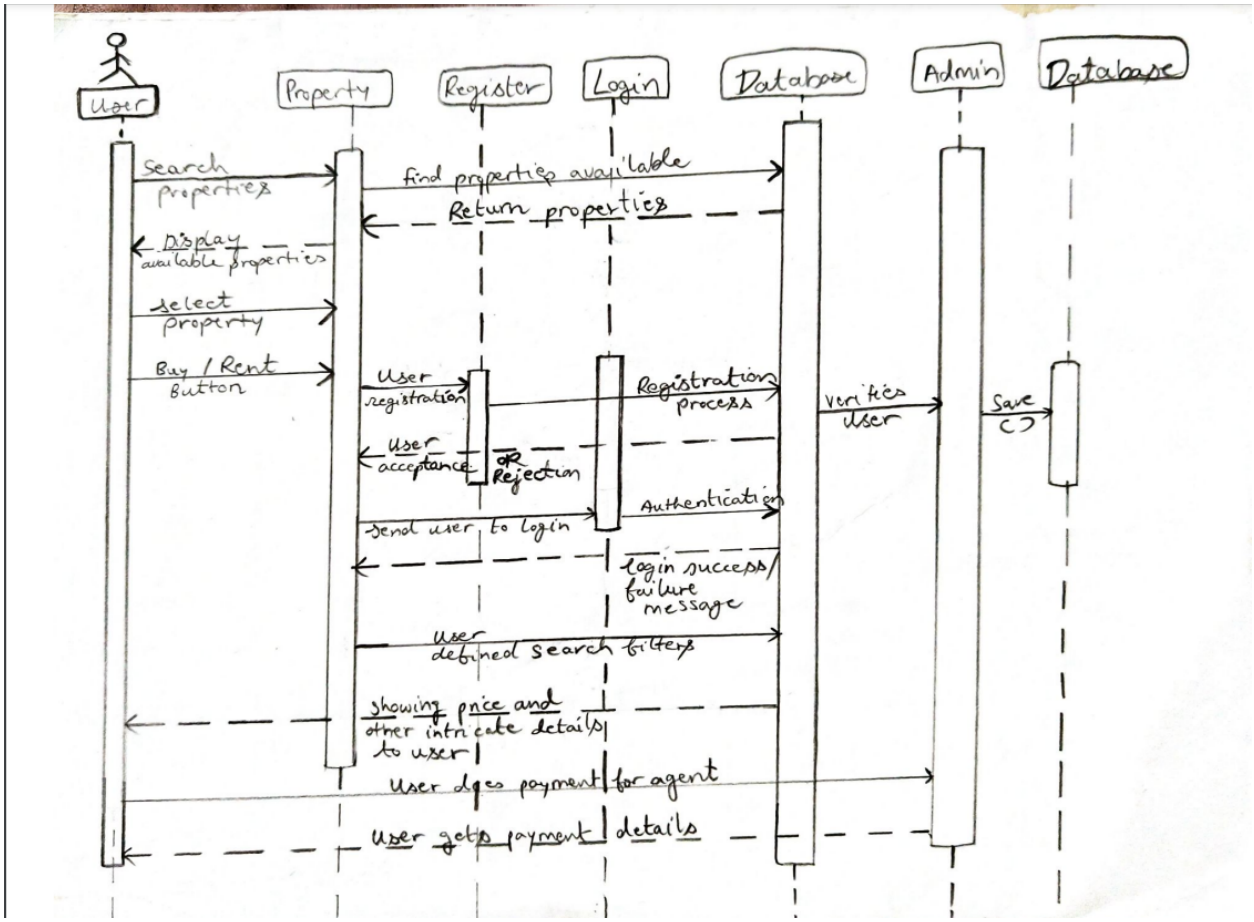
- **Justification**: Allows administrators to oversee the platform, manage listings, and monitor user activity.

Use case diagram:-

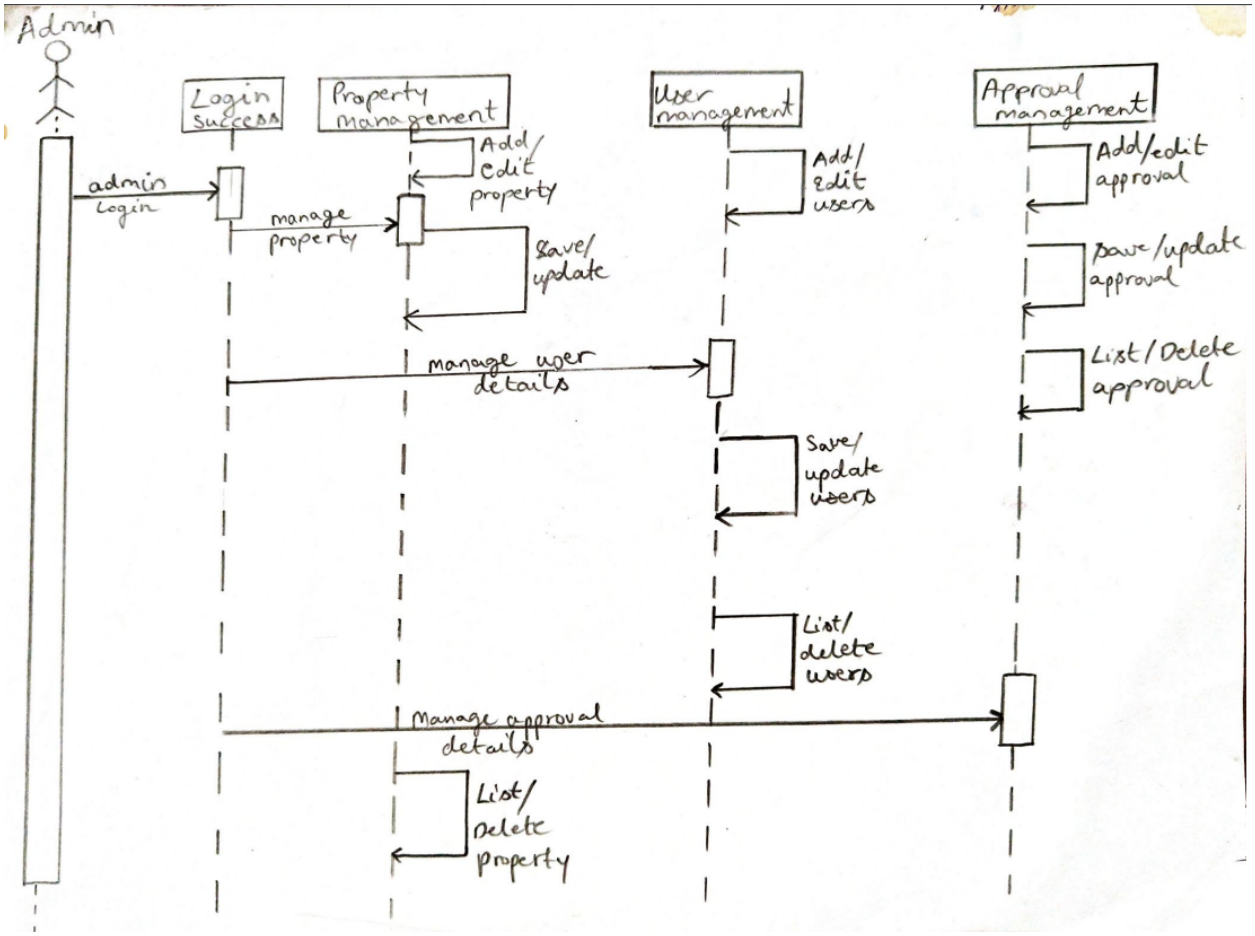


[Use case diagram link]

Sequence diagram:-
User side:-


User
Property
Register
Login
Database
Admin
Database
Search properties
find properties available
Return properties
Display available properties
Select property
Buy / Rent Button
User registration
Registration process
verifies user
Save
User acceptance
OR
Rejection
Send user to login
Authentication
login success/ failure message
User defined search filters
showing price and other intricate details to user
User does payment for agent
User gets payment details

Admin side:-


Admin
Login success
Property management
User management
Approval management
admin login
Add/ Edit property
manage property
Save/ update
Add/ edit users
Add/edit approval
Save/update approval
List/Delete approval
manage user details
Save/ update users
List/ delete users
Manage approval details
List/ Delete property

4. Non-Functional Requirements

---

1. **Performance:**
    - **Response Time:** should be as much less as possible
2. **Reliability:**
    - **System Uptime:** Ensure a high level of availability, with a goal of 99.9% or higher, to minimize downtime and disruptions to users.
3. **Security:**
    - **Data Encryption:** All sensitive data, including user credentials, financial information, and property records, should be encrypted at rest and in transit using strong encryption protocols.
    - **Access Control:** user should be able to access specific areas unless access given to them.
    - **Authentication and Authorization:** Use secure authentication mechanisms like multi-factor authentication (MFA). Implement fine-grained authorization to ensure that users can only access the data and perform actions they are allowed to.
4. **Scalability:**
    - Ensure that the system can handle an increasing number of properties and users over time without a significant drop in performance. Define scalability targets for property listings, users, and concurrent sessions.
5. **Usability:**
    - Provide an intuitive and user-friendly interface to make it easy for users to navigate and perform tasks within the system.
    - Perform usability testing to ensure that the system meets user expectations.
6. **Maintainability:**
    - Make the system easy to maintain and update. Consider modular architecture and version control practices to streamline updates and bug fixes.
7. **Backup and Recovery:**
    - Implement regular automated backups of the system's data to prevent data loss in case of failures or disasters.

8. **Logging and Auditing:**
    - Implement comprehensive logging of system activities to track user actions and system events. Ensure that logs are secure and reviewable for auditing purposes.
9. **Performance Testing:**
    - Conduct load testing and stress testing to evaluate how the system performs under peak loads, ensuring that it meets performance requirements.
10. **Compliance:**
    - Ensure the system complies with relevant industry standards, regulations, and best practices, especially in the real estate sector.
11. **Accessibility:**
    - Ensure that the system is accessible to individuals with disabilities, in compliance with accessibility standards
12. **Disaster Recovery and Business Continuity:**
    - Develop a plan for disaster recovery and business continuity, including backup data centers or cloud resources in case of a catastrophic event.

5. External Interfaces:

---

1. **User Interfaces (UI):** The primary external interface is the user interface through which real estate professionals, clients, and other stakeholders interact with REMS. It includes web-based forms, dashboards, and mobile applications.
2. **Database Management System (DBMS):** REMS interfaces with a database management system (e.g., MySQL, PostgreSQL, or others) for data storage and retrieval. This includes creating, reading, updating, and deleting data in the database.
3. **Customer Support and Communication Tools:** Integration with communication and customer support tools (e.g., live chat, support ticket systems) to facilitate interactions with clients and users.

6. User Interfaces:

Figma wireframe:-

https://www.figma.com/file/6BDdE455AZEoiNpeGRh9bo/G17_WireFrame?type=design&node-id=0%3A1&mode=design&t=YV9RZLfRs6T1MdU8-1

Design documentation:-

https://docs.google.com/document/d/1knkmHVS9_RmLyb5_vx_tqfQwfL_jHaRNNu3nU4ekBNo/edit

7. System Features:

---

1. **Property Listings Management:**
    - Add, edit, and showcase property listings with multimedia support.
    - Include comprehensive property details such as images, videos, and descriptions.

2. **Customer Relationship Management (CRM):**
    - Efficiently manage client interactions, communications, and histories.
    - Automate follow-ups and set reminders for important client interactions.
3. **Reporting & Analytics:**
    - Generate detailed reports on sales trends, property performance, and financial data.
    - Provide analytics tools for data-driven decision-making.
4. **User Access Control:**
    - Define different levels of access for agents, managers, and administrators.
    - Ensure data security and privacy through role-based access control.
5. **Lease Management:**
    - Manage rental properties, tenant details, and lease agreements.
    - Keep track of rental payment records and lease terms.
6. **Document Management:**
    - Store and manage important documents, including agreements and contracts.
    - Facilitate easy access and retrieval of critical paperwork.
7. **Integration with MLS:**
    - Interface with Multiple Listing Service (MLS) platforms for broader property exposure.
    - Ensure seamless synchronization of property listings across different platforms.
8. **Communication Tools:**
    - Send emails or SMS notifications directly from the platform.
    - Enhance communication and responsiveness with clients and stakeholders.
9. **Scalability:**
    - Adapt to the growing needs of users and data volume.
    - Ensure the system remains effective as the real estate portfolio expands.

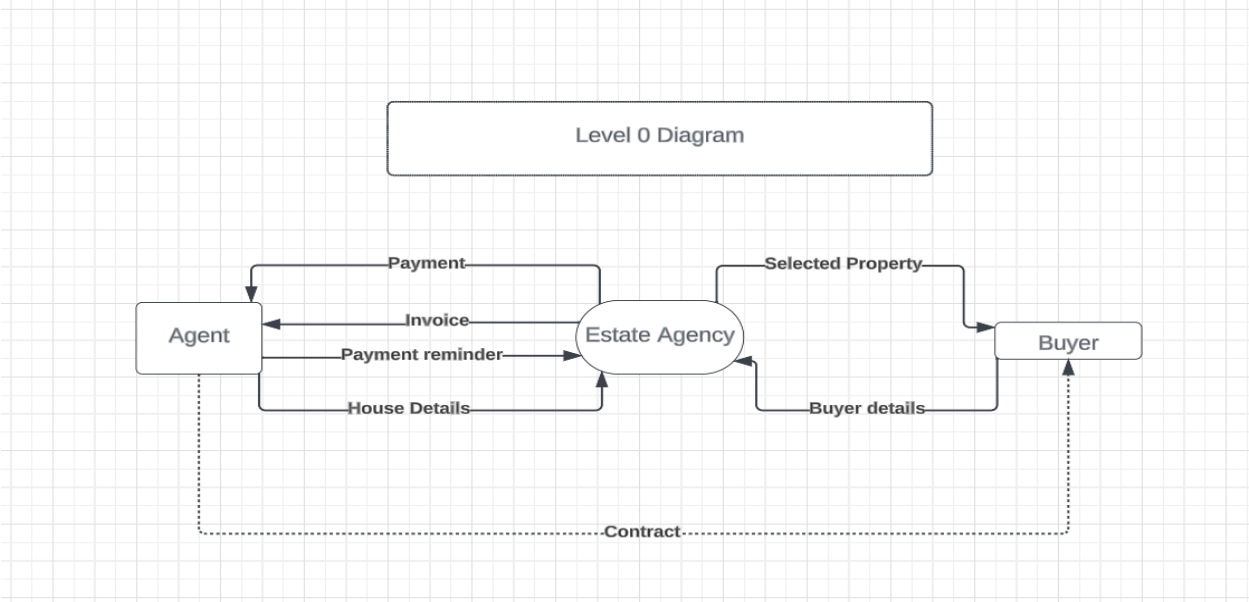
8. Data Requirements:

ER diagram:-

https://lucid.app/lucidchart/f5c2c0ce-41f8-49d2-820e-251cfe7e2051/edit?invitationId=inv_a96fa9b0-d7b0-4b95-ba84-16ade5858bc8&page=0_0#

Data flow diagram:-

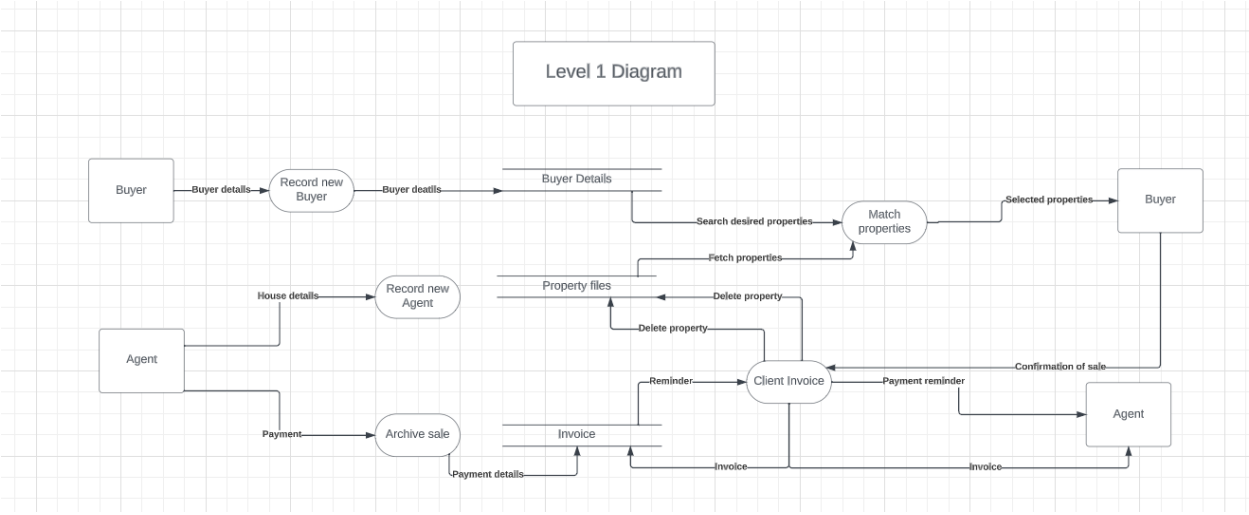
Level 0:-



For clear view:-

https://lucid.app/lucidchart/a4429aec-dd99-4139-aedd-b78b8cc46e80/edit?beaconFlowId=EC634049DE6579B0&invitationId=inv_d9ad7064-8345-4732-8d6c-9fb2d70b8ac4&page=0_0#

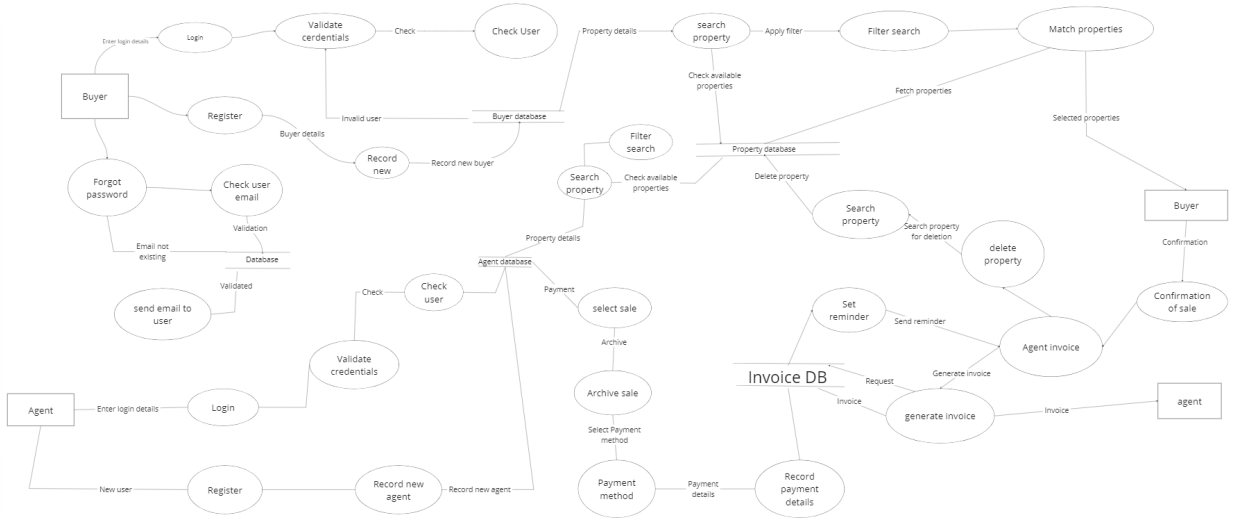
Level 1:-



For clear view:-

https://lucid.app/lucidchart/b4371171-3604-443c-afa0-530ec7167c37/edit?beaconFlowId=F9415FC31AA8BC69&invitationId=inv_ac463cf0-3b4a-4b28-ae62-ecf7ca6713e9&page=0_0#

Level 2:-



For clear view:-

https://miro.com/welcomeonboard/ZnROU0FPb3JHckkyOHFINTFRMzFqdVA0M3BOTGVDRGdHbDJ4cXNtRXJrbGxuMUJhWVpYbVI3YVZUWVM3bllyc3wzNDU4NzY0NTQ1Mjc2MjY3MDA2fDI=?share_link_id=448725766807

9. Testing Requirements:

9.1 Testing approach:-

1. Unit Testing:

- **Objective:** Verify the functionality of individual units or components (e.g., functions, methods).
- **Requirements:** Developers should conduct unit tests during the development phase to catch and rectify code-level issues.

2. Integration Testing:

- **Objective:** Confirm that different modules or components of REMS work seamlessly together.
- **Requirements:** Ensure proper data flow, API integrations, and communication between various parts of the system.

3. System Testing:

- **Objective:** Validate the entire system against its specified requirements.
- **Requirements:** Comprehensive testing of all functionalities, user interfaces, databases, and external integrations.

4. User Acceptance Testing (UAT):

- **Objective:** Verify that REMS meets user expectations and business requirements.
- **Requirements:** Involvement of end-users to perform real-world scenarios and validate that the system aligns with their needs.

5. Performance Testing:

- **Objective:** Assess the system's responsiveness, speed, and stability under different loads.
- **Requirements:** Ensure that REMS can handle the expected number of concurrent users and data volume without performance degradation.

6. Security Testing:

- **Objective:** Identify and address vulnerabilities to protect against unauthorized access, data breaches, or other security threats.

- **Requirements:** Test authentication, authorization, encryption, and other security measures.

7. Compatibility Testing:

- **Objective:** Ensure that REMS works correctly across various browsers, devices, and operating systems.
- **Requirements:** Test compatibility with the specified browsers (e.g., Chrome, Firefox, Safari) and devices (desktop, tablet, mobile).

9.2 Testing requirements:-

1. **Test Cases:**
   - Develop comprehensive test cases covering all functionalities and scenarios.
   - Include positive and negative test cases to validate expected behavior and error handling.

2. **Test Data:**
   - Prepare realistic and diverse test data to simulate real-world usage.
   - Include edge cases, boundary values, and data variations.

3. **Testing Environment:**
   - Set up a testing environment that mirrors the production environment.
   - Ensure that databases, servers, and configurations are consistent with the live environment.

4. **Traceability Matrix:**
   - Create a traceability matrix to map test cases to specific requirements.

- Ensure that each requirement has corresponding test coverage.

5. **User Feedback:**
   - Incorporate feedback from users during UAT to address any usability concerns or unmet needs.

Testing documentation:-

https://docs.google.com/document/d/1iVj8T97TBmfYwlfLfL4YXP2F5ihOvbDY8vmAuiPn0-k/edit

10.Performance Requirements:

---

1. **Static**

- **Concurrent user handling:** The system must work smoothly for various users and agents that try to access the site.

- **Storage:** The database needs to store “detailed information” about the property that is to be uploaded to the site.
- **Accessibility:** The website must be online for 24 hours every day and available on various devices like laptops, desktops, Tablets, and Mobile phones.

2. **Dynamic**

- **Property Updates:** The users must be notified about the availability of newly uploaded properties on the website.
- **Deal Updates:** if multiple users are in negotiations for the deal reof the same property upon deal finalization with one user other users must be informed immediately.
- **Agent-User interactions: -** The Agents and users must be able to communicate with each other in real-time.
- **Financial Transactions: -** The agents must be notified immediately about the status of payment.

11. Security Requirements:

---

1. **User Authentication:**
    - **Requirement:** Users must authenticate themselves using secure, unique credentials (e.g., username and strong password) to access the REMS.
    - **Implementation:** Use robust authentication mechanisms, such as multi-factor authentication (MFA), to enhance user account security.

2. **Access Control:**
    - **Requirement:** specific areas of access until access given
    - **Implementation:** Define and enforce access levels for each role to ensure that users can only access information relevant to their responsibilities.

3. **Data Encryption:**
    - **Requirement:** All sensitive data, including user credentials and transaction details, must be encrypted during transmission and storage.
    - **Implementation:** Use secure encryption algorithms

4. **Secure Session Management:**
    - **Requirement:** Implement secure session management to prevent session hijacking or unauthorized access.
    - **Implementation:** Use secure session tokens, implement session timeouts, and ensure secure cookie attributes.

5. **Secure File Uploads:**
    - **Requirement:** If the system allows file uploads (e.g., property images, documents), ensure a secure and controlled mechanism.
    - **Implementation:** Validate file types, size limits, and implement malware scanning to prevent malicious uploads.

6. **Data Integrity:**

- **Requirement:** Ensure the integrity of stored data to prevent unauthorized modifications.
- **Implementation:** Implement hashing algorithms and regular data integrity checks to detect and mitigate data tampering.

7. **Incident Response Plan:**
   - **Requirement:** Have a documented incident response plan to address security breaches or incidents.
   - **Implementation:** Define roles and responsibilities, establish communication protocols, and conduct regular drills to ensure a swift and effective response.

8. **Data Backups:**
   - **Requirement:** Regularly backup critical data to prevent data loss due to accidental deletion or malicious activities.
   - **Implementation:** Store backups securely, test the restoration process periodically, and ensure backups comply with data protection regulations.

12. Documentation Requirements:

---

1. **Requirements Specification:**
   - Purpose: Clearly define the functional and non-functional requirements of REMS.
   - Content:

        - Business requirements
        - User stories
        - Functional requirements
        - Non-functional requirements (performance, security, etc.)
2. **System Architecture Documentation:**
    - Purpose: Provide an overview of the system's architecture and design.
    - Content:
        - High-level system architecture diagram
        - Components and modules description
        - Data flow diagrams
        - Dependency relationships
3. **Database Design Documentation:**
    - Purpose: Describe the structure and organization of the database used by REMS.
    - Content:
        - Entity-relationship diagrams (ERD)
        - Database schema
        - Table structures and relationships
        - Indexing and data constraints
4. **User Manual:**
    - Purpose: Guide users on how to use REMS effectively.
    - Content:
        - System login and navigation instructions
        - Feature descriptions and usage guidelines
        - Troubleshooting tips
        - FAQs
5. **Technical Documentation:**
    - Purpose: Aid developers, administrators, and support teams in understanding the technical aspects of REMS.
    - Content:
        - Code documentation (comments, inline documentation)
        - API documentation (if applicable)
        - Configuration settings
        - Deployment instructions
6. **Testing Documentation:**
    - Purpose: Document the testing strategy, test cases, and results.
    - Content:

        - Test plan outlining testing scope and objectives
        - Test cases and test scripts
        - Test results and defect reports
        - User acceptance testing (UAT) documentation
7. **Security Documentation:**
    - Purpose: Outline the security measures implemented in REMS.
    - Content:
        - Security requirements and specifications
        - Threat models and risk assessments
        - Security architecture details
        - Incident response plan
8. **Release Notes:**
    - Purpose: Communicate changes, new features, and bug fixes in each release.
    - Content:
        - Summary of changes
        - Known issues
        - Upgrade instructions (if applicable)
9. **Maintenance and Support Documentation:**
    - Purpose: Assist support teams in maintaining and troubleshooting REMS.
    - Content:
        - Troubleshooting guides
        - FAQs for common issues
        - Contact information for support
        - SLA (Service Level Agreement) if applicable
10. **Training Materials:**
    - Purpose: Provide training resources for users and administrators.
    - Content:
        - Training manuals
        - Video tutorials
        - Webinar recordings (if applicable)
        - Training schedules