
SOFTWARE REQUIREMENTS SPECIFICATION

For

REAL ESTATE LISTING

Prepared by:-

- *Nasvin.M*
- *Karthick.K*
- *Karmukilan.B.P*

- **Introduction**

1. Purpose

The business model of a Real Estate listing website revolves around connecting property buyers with sellers. It saves time and efforts of both the parties and bridges the gap between them. The broker/seller lists the property on the website which is available to the people looking to buy/rent.

2. Document Conventions

- i. Entire document should be justified.
- ii. Convention for Main title
 - Font face: Times New Roman
 - Font style: Bold
 - Font Size: 14
- iii. Convention for Sub title
 - Font face: Times New Roman
 - Font style: Bold
 - Font Size: 12
- iv. Convention for body
 - Font face: Times New Roman
 - Font Size: 12

3. Scope of Development Project

Creating a successful real estate listings project involves prioritizing user experience with features like detailed listings and advanced search, ensuring accessibility through responsive design, and fostering engagement with features like ratings and contact forms. On the technical side, integration with external APIs and secure payment gateways adds value, legal compliance ensures regulatory alignment, and scalability accommodates potential growth. Implementing SEO practices enhances visibility, while robust security measures safeguard user data. Choosing a flexible technology stack allows for adaptability and evolution in response to market dynamics. In summary, a successful project seamlessly integrates user-centric design with robust technical foundations for a secure and engaging experience.

4. Definitions, Acronyms and Abbreviations

JAVA -> platform independence

SQL-> Structured query Language

ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment

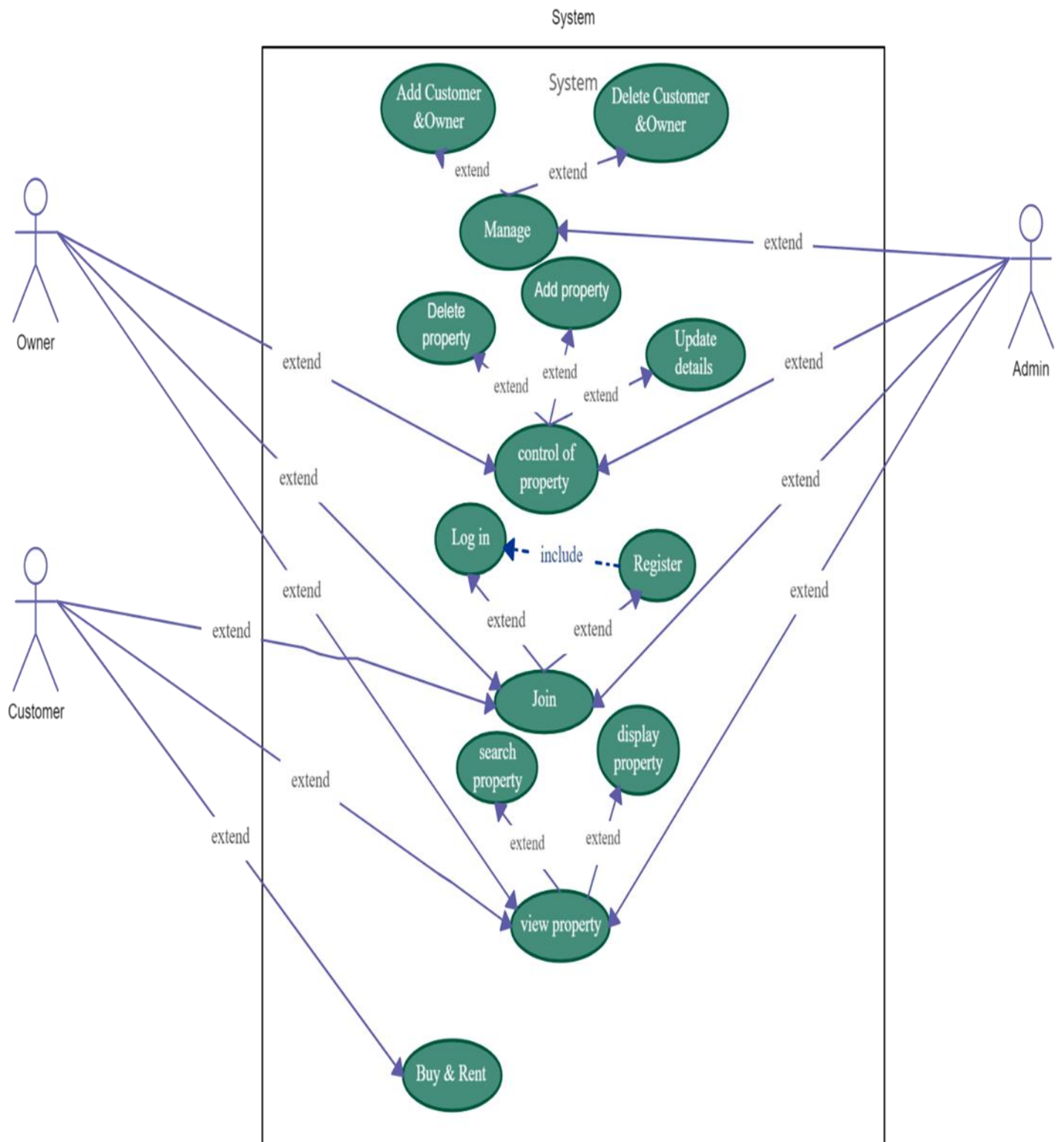
SRS-> Software Requirement Specification

5. Overall Descriptions

a. Product Perspective

Use Case Diagram of Real Estate Listing . . .

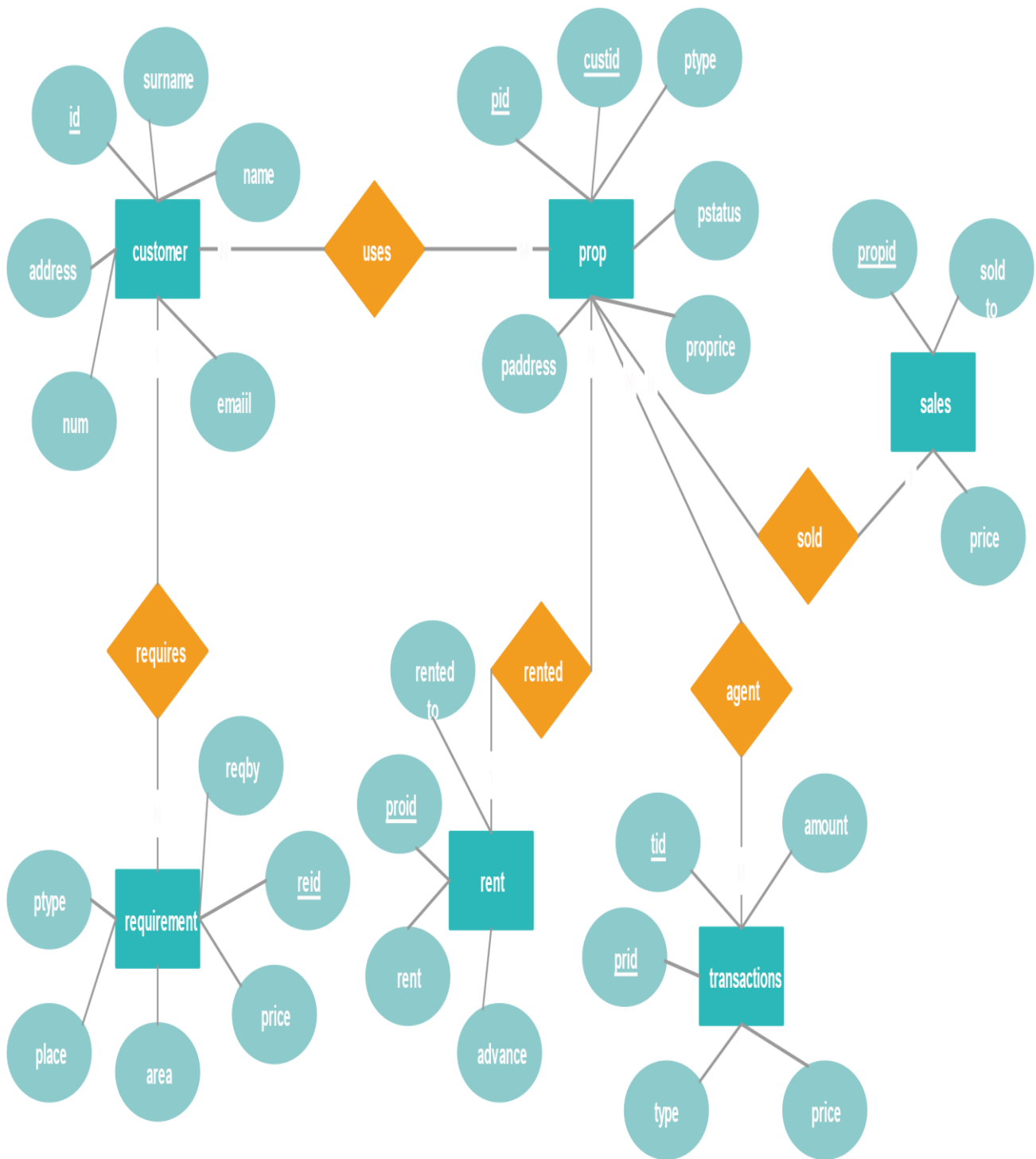
e



The entity-relationship diagram of property management systems depicts all of the visual tools of database tables and the relationships between buyers, properties, registrations, etc.

b. Product Function

Entity Relationship Diagram of Real estate Listing. . .



6. User Classes and Characteristics:

1. Real Estate Agents (Realtors): Property Management: Create and manage property listings. Upload property details, images, and videos. Transaction Handling: Facilitate property transactions, including sales and rentals. Manage property inquiries and negotiations. Analytics and Reports: Generate reports on property performance and market trends. Analyze user behavior and preferences. User Management: Access and manage user accounts, including buyers and sellers. Maintain a database of property owners and potential buyers.
2. Buyers: Property Search: Search for properties based on preferences (location, price range, type, etc.). Utilize advanced search filters for a more refined property search. Account Management: Create and manage a user account securely. Save favorite properties and searches for future reference. Property View and Inquiry: View detailed property listings, including images, descriptions, and pricing. Send inquiries or requests for property viewings. Transaction History: View the history of property inquiries and transactions. Access details of properties currently owned or rented.
3. Sellers: Listing Submission: Submit details of properties they want to sell or rent. Upload property images, videos, and relevant documentation. Transaction Management: Track and manage inquiries and offers on their listed properties. Facilitate negotiations and finalize transactions. Analytics: Access analytics related to their listed properties. Understand market interest and user engagement.
4. Admin: User and Content Management: Administer user accounts, including realtors, buyers, and sellers. Oversee and moderate property listings and content. Analytics and Reporting: Access overall analytics and reports for the platform. Monitor user activities, trends, and system performance. System Configuration: Configure system settings, such as search algorithms and notification preferences. Manage system updates and security protocols.

7.External Interface Requirement

c. GUI

The software provides a robust graphical interface for users, allowing administrators to perform necessary tasks such as property management, updates, and detailed property viewing.

- Users can generate quick reports such as properties listed/sold within a specific time frame.

- The system offers stock verification for properties and a versatile search functionality based on various criteria (e.g., location, price range, property type).
- Administrators have the ability to customize the user interface to meet specific preferences and requirements.
- All modules within the software seamlessly integrate into the graphical user interface, adhering to defined standards.

8.Data and category requirements :

Property Details: Basic Information: Include the property's address, type (e.g., house, apartment, condo), and the number of bedrooms and bathrooms. **Square Footage:** Provide the total living space or square footage of the property. **Year Built:** Specify the year the property was constructed. **Lot Size:** If applicable, mention the size of the land associated with the property. **Property Features: Interior Features:** Describe the interior features, such as flooring type, ceiling height, and any additional amenities (fireplace, walk-in closets, etc.). **Appliances:** List the appliances included in the sale or rental.

Utilities: Specify utilities available, such as heating, cooling, and water sources. **Exterior Features: Parking:** Indicate the availability and type of parking (e.g., garage, driveway). **Outdoor Spaces:** Describe any outdoor features like a garden, patio, balcony, or pool. **Property Images:** Include high-quality images showcasing the property's exterior and interior. Images significantly impact a potential buyer's or renter's decision-making process.

Price and Terms: Clearly state the price or rental rate. Specify any additional costs, such as homeowners association (HOA) fees or property taxes. Outline the terms of the sale or rental, including the duration of the lease and any deposit requirements. **Location Information:** Highlight the property's proximity to key locations such as schools, hospitals, shopping centers, and public transportation. **Contact Information:** Provide accurate contact details for the real estate agent or owner, including phone number and email address.

Listing Status: Clearly indicate whether the property is for sale, rent, or lease. Mention if there are any special conditions or offers. **Property Condition:** Disclose the overall condition of the property. If there are any issues or renovations, it's essential to be transparent. **Additional Information:** Include any relevant information that sets the property apart, such as recent renovations, unique features, or special financing options. **Property History:** If applicable, provide a brief history of the property, including any significant renovations or changes.

9. System Requirements:

Software Requirements: **Operating System:** The server can run on Linux (Ubuntu, CentOS), Windows Server, or another suitable operating system based on your development preferences. **Web Server:** Apache, Nginx, or Microsoft Internet Information Services (IIS) can serve as the web server to handle HTTP requests.

Database Management System (DBMS): Choose a reliable DBMS to store property information. Options include MySQL, PostgreSQL, MongoDB, or Microsoft SQL Server. **Programming Language:** Select a server-side programming language. Common choices include Python (Django or Flask), Ruby (Ruby on Rails), PHP, or Node.js.

Backend Framework: Use a backend framework to expedite development. Frameworks like Django, Ruby on Rails, Laravel (for PHP), or Express.js (for Node.js) are popular choices. **Authentication and Authorization:** Implement secure user authentication and authorization mechanisms to protect user data and ensure secure access.

RESTful API: Design and implement a RESTful API to facilitate communication between the server and client applications. **Data Security:** Implement encryption (e.g., HTTPS/SSL) for secure data transmission. Apply proper validation and sanitation techniques to prevent common web vulnerabilities. **Search Engine:** Integrate a search engine or use database indexing for efficient property searches. **Version Control:** Use version control systems like Git to manage and track changes in the source code. **Database Requirements:** **Data Schema:** Define a clear and efficient database schema to store property details, user information, and related data. **Indexing:** Implement appropriate indexing to optimize database query performance, especially for search functionalities. **Backup and Recovery:** Establish a robust backup and recovery system to prevent data loss in case of failures or accidents.

Client-Side Requirements:

Web Browser Compatibility: Ensure compatibility with major web browsers (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge). **Responsive Design:** Implement a responsive design to ensure a seamless experience across various devices (desktops, tablets, smartphones).

User Interface (UI): Develop an intuitive and user-friendly interface for property search, filtering, and listing. **Map Integration:** Integrate maps (e.g., Google Maps) for property location visualization and navigation. **Image Handling:** Support image uploads and optimize image handling for property photos. **Search and Filters:** Implement effective search and filtering options to allow users to find properties based on various criteria changes, or other relevant information.

10.Security requirements:

Data Encryption: Ensure that all communication, especially sensitive data transmission, is encrypted using secure protocols like HTTPS. **Access Controls:** Implement robust access controls to restrict access to the real estate listing system. Only authorized personnel should have access to sensitive information. Use role-based access control (RBAC) to assign different levels of access based on job responsibilities. **User Authentication:** Enforce strong password policies and encourage the use of multi-factor authentication (MFA) to add an extra layer of security. Regularly audit and update user credentials, especially when there are personnel changes.

Data Backups: Regularly back up all real estate listing data to prevent data loss due to system failures, cyber attacks, or other unforeseen events. **Audit Trails:** Implement audit trails to log and monitor user activities within the system. This helps in identifying any suspicious or unauthorized activities.

Data Privacy Compliance: Ensure compliance with data protection regulations, such as GDPR or similar regional laws, by implementing privacy measures and obtaining necessary consents. **Firewall and Intrusion**

Detection/Prevention: Employ firewalls to monitor and control incoming and outgoing network traffic. Use intrusion detection and prevention systems to identify and respond to potential security threats.

Secure APIs: If the real estate listing platform uses APIs, secure them by implementing proper authentication and authorization mechanisms. **Regular Security Audits:** Conduct regular security audits and vulnerability assessments to identify and address potential security risks. **Physical Security:** If the real estate listing involves physical documents, ensure that they are stored securely, and access is restricted to authorized personnel. **Incident Response Plan:** Develop and maintain an incident response plan to address security incidents promptly and effectively.

Training and Awareness: Provide training to employees on security best practices and raise awareness about the potential risks associated with real estate listing data. **Vendor Security Assessment:** If third-party vendors are involved (e.g., hosting providers, software providers), assess their security measures to ensure they meet your standards. **Encryption of Stored Data:** Encrypt sensitive data at rest to protect it in case of unauthorized access to storage devices.

Mobile Security: If there's a mobile application for real estate listings, ensure it adheres to secure coding practices and implement secure communication protocols. **Regular Software Updates:** Keep all software components, including the operating system, databases, and applications, up-to-date with the latest security patches

