

SDLC Model

Real Estate with 360 website



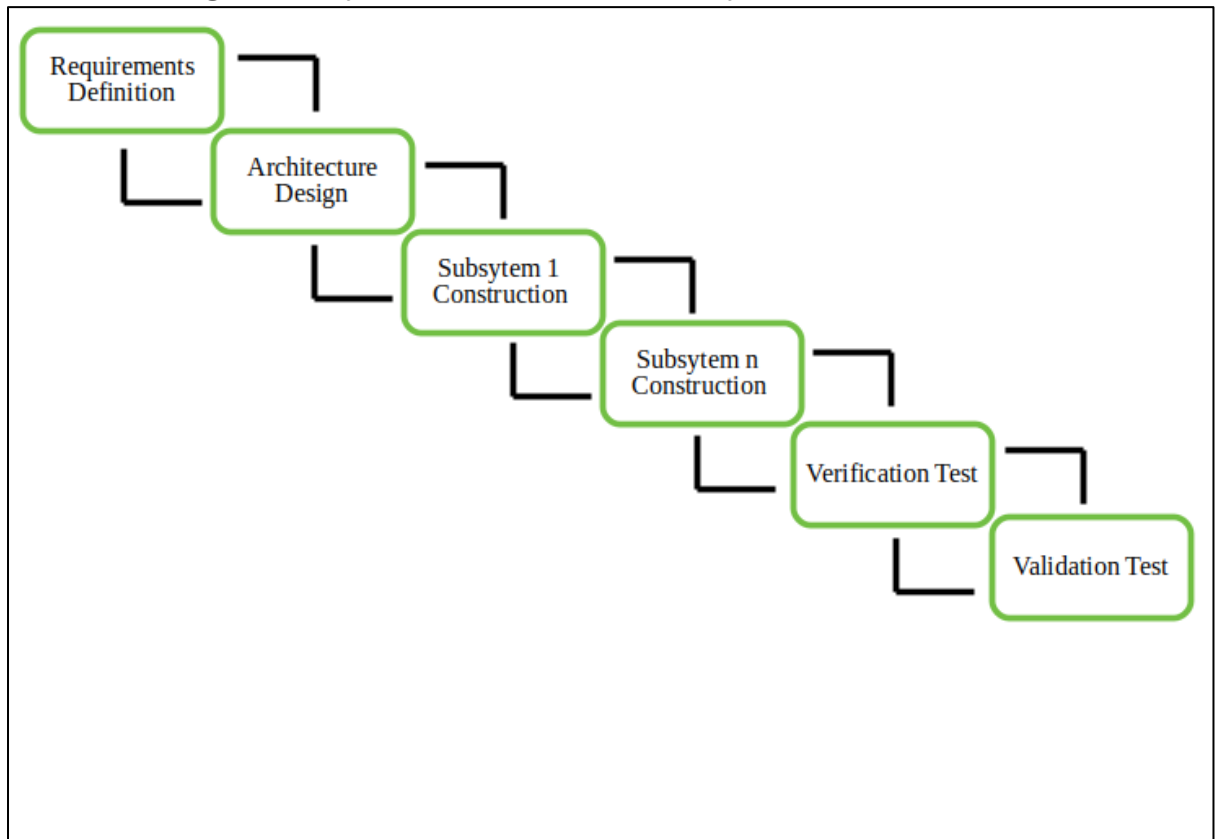
Definition:

The Incremental Model is a software development approach where the system is built and delivered in small, manageable pieces, called increments. Each increment adds functionality to the system until the complete system is implemented.

Key Features:

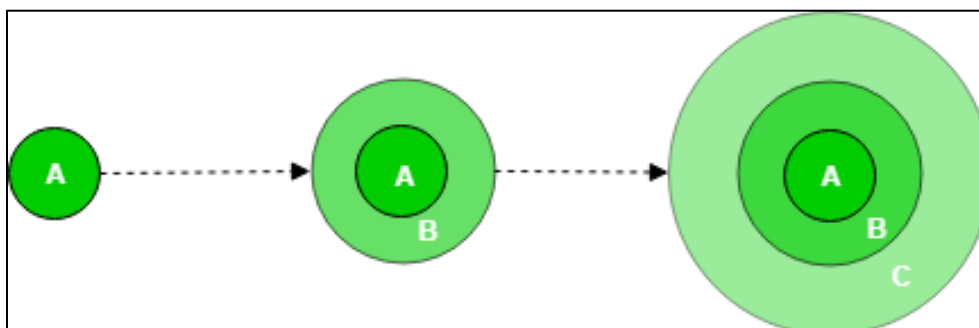
- **Incremental Delivery:** The system is developed in parts, with each increment adding new features or functionality.
- **Client Feedback:** Regular client involvement and feedback are integral, allowing for adjustments throughout development.

- Iteration: Development is iterative, refining and expanding the system with each cycle



Phases:

1. Requirement Analysis: Define the requirements for the first increment.
2. Design: Create a design for the increment.
3. Implementation: Develop and implement the increment.
4. Testing: Test the increment to ensure it works correctly.
5. Integration: Integrate the increment into the existing system.
6. Feedback: Gather client feedback for the next iteration.



Advantages:

- **Flexibility:** Easier to adapt to changing requirements.
- **Early Delivery:** Parts of the system can be deployed earlier.
- **Risk Management:** Smaller increments make it easier to manage risks.

Disadvantages:

- **Complex Integration:** Integrating new increments with the existing system can be challenging.
- **Incomplete System:** Early increments might not provide full functionality.

Use Cases:

Ideal for complex systems, customer-driven projects, and situations where early deployment is crucial.

List of Increments in project:

Increment 1: Project Planning and Requirements Gathering

- Defined project scope and objectives.
- Identified core features (property listings, 360-degree view integration, search functionality, etc.).
- **Deliverables:** Finding a solution for the given problem statement

Increment 2: Setting Up Development Environment and Basic Website Structure

- **Frontend:**
 - Create initial wireframes and UI design for the homepage, property listing page, and 360-view page.
 - Use HTML, CSS, and JavaScript for basic layouts and navigation.

- **Backend:**
 - Set up the server environment (e.g., Node.js, Python Flask/Django, or PHP).
 - Configure the database (MySQL, PostgreSQL, or MongoDB) for storing property details.
- **Deliverables:**
 - Functional skeleton of the website with simple navigation.
 - Database schema for property listings.

Increment 3: 3D Model Creation and Integration

- **3D Model Preparation:**
 - Use Blender to create detailed 3D models of properties.
 - Capture 360-degree views with Matterport for selected properties.
- **Frontend:**
 - Display basic property details with images on the listing page.
 - Integrate static 3D models and 360-degree views into property detail pages.
- **Deliverables:**
 - Basic frontend showing property details with integrated 3D models and 360-degree views.

Increment 4: Property Listing and Search Functionality

- **Backend:**
 - Develop a robust database schema to store property listings, including filters like price, location, and size.
 - Implement APIs for retrieving filtered property listings.
- **Frontend:**

- Create a dynamic property listing page that displays properties from the database.
 - Implement a search and filtering system based on user input (location, price, size, etc.).
- **Deliverables:**
 - Fully functional property listing with real-time search and filter capabilities.

Increment 5: User Interface Enhancements and Responsive Design

- **Frontend:**
 - Enhance the UI/UX by refining the design with better color schemes, typography, and responsive layouts.
 - Improve the responsiveness of the website to ensure it works smoothly across different devices (mobile, tablet, desktop).
 - Add interactive elements like property galleries and animations for a better user experience.
- **Deliverables:**
 - Polished and responsive UI across all web pages.

Increment 6: Testing, Performance Optimization, and Bug Fixing

- **Backend:**
 - Test the API endpoints for performance and ensure that property listings and 3D model data are fetched efficiently.
- **Frontend:**
 - Conduct usability testing to identify UI issues, bugs in 360-view interactions, and search functionality errors.
 - Optimize loading times for 3D models and images.

Deliverables:

- A bug-free, optimized website ready for final deployment.

Increment 7: Final Deployment**Backend**

- Deploy the website to a cloud service (e.g., AWS, Heroku, DigitalOcean) or a dedicated hosting server.
- Set up SSL certificates and ensure secure access to the website.

Frontend:

- Conduct final design touches and prepare the website for public access.

Deliverables:

- Deployed and fully functional website.

Software Requirements Specification (SRS) for Real Estate Website with 360 and 3D Views

1. Introduction

Purpose

The purpose of this document is to define the Software Requirements Specification (SRS) for a web-based real estate platform that provides 360-degree and 3D views of properties. This document is intended for developers, project managers, and stakeholders involved in the project to ensure a clear understanding of the system's requirements.

Scope

This platform aims to provide users with an immersive experience while browsing real estate properties. Users will be able to explore properties through 360-degree and 3D views, enhancing their decision-making process. The website will offer features such as property searches, virtual tours, contact with agents, and user accounts for saving favorite properties.

Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **360 View:** A panoramic image view that allows users to look around a scene as though they were physically present.
- **3D View:** A three-dimensional model allowing users to explore the depth and detail of a property.
- **User:** Refers to anyone using the system, including property buyers, sellers, and real estate agents.

References

- Web development frameworks: ReactJS, NodeJS
- Virtual tour libraries: Three.js, A-Frame
- Database: MongoDB
- Previous SRS templates

2. Overall Description

Product Perspective

The Real Estate website is part of a larger ecosystem where potential buyers and real estate agents interact digitally. The website will be accessible on multiple platforms, including desktop and mobile, allowing users to explore properties virtually without needing to physically visit each location.

Product Functions

- Search properties by location, price, and property type
- 360-degree and 3D virtual tours of listed properties
- User registration, login, and profile management
- Save and share favorite properties
- Contact real estate agents through integrated chat
- Upload and manage property listings for agents

User Characteristics

There are two types of users for the system:

- **Property Buyers:** Individuals looking to explore properties, usually with limited technical knowledge.
- **Real Estate Agents:** Individuals or companies listing properties for sale, with moderate technical understanding for uploading and managing listings.

Constraints

- Must be mobile-responsive to support various screen sizes
- Integration with real estate databases
- High-resolution images and models might increase the load time, thus optimizations are needed for performance.

Assumptions and Dependencies

- Assumed that users will have internet access and a modern browser.
- The system depends on the availability of external APIs and services for mapping, 3D model rendering, and messaging.

3. Specific Requirements

Functional Requirements

- Users must be able to search properties using filters (price, location, etc.).
- The system must support uploading high-quality 360-degree images and 3D models.
- Property details should include images, 3D views, price, location, and contact information for the agent.
- Users must be able to create accounts, log in, and save properties to a favorites list.
- Agents should be able to upload new property listings, including photos and 3D models.
- A messaging feature should be available for users to communicate with agents.

Non-Functional Requirements

- **Performance:** The website should load within 3 seconds, even when displaying high-resolution images or 3D models.
- **Security:** User data, especially personal information, must be encrypted and securely stored.
- **Usability:** The website must be intuitive, with a user-friendly interface.
- **Reliability:** The platform should have at least 99.5% uptime.

External Interface Requirements

- Integration with payment gateways for future premium services.
- Mapping APIs for displaying property locations.
- Use of cloud storage services for storing 3D models and images.

System Features

- **Property Search:** Users can search for properties based on various criteria.
- **360-degree View:** Users can explore property interiors through panoramic views.
- **3D Property Tours:** Users can walk through a property in 3D.
- **User Profiles:** Registered users can save and revisit favorite properties.
- **Messaging:** Users and agents can communicate through the website.

4. Other Requirements

Performance Requirements

- The system must handle at least 100 concurrent users without significant performance degradation.
- 3D models should load within 5 seconds on standard broadband internet speeds.

Safety Requirements

- There are no specific safety concerns beyond standard data protection regulations.

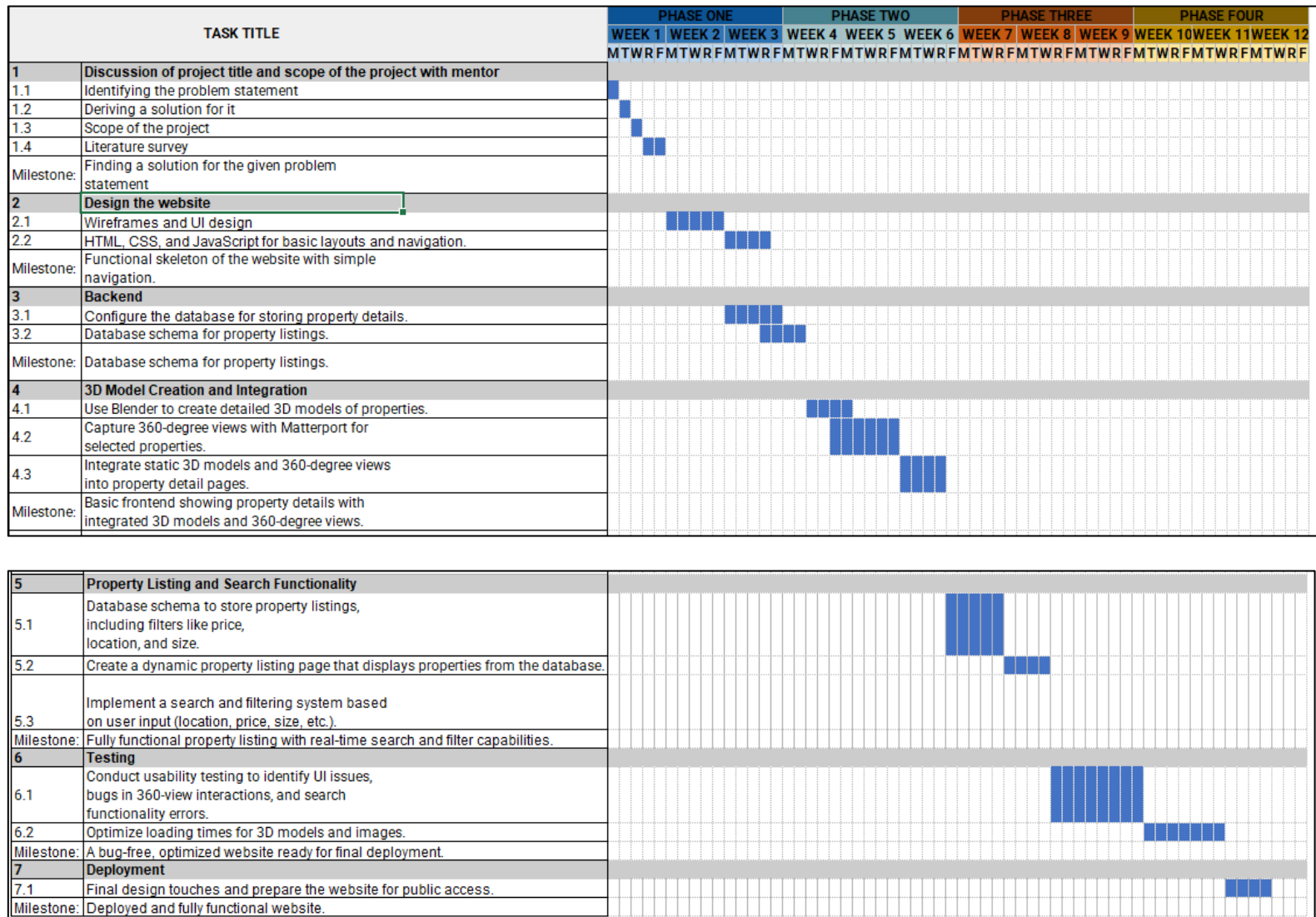
Security Requirements

- All user passwords must be stored securely using hashing algorithms.
- Data transferred between the client and server must use HTTPS.

Software Quality Attributes

- **Maintainability:** The codebase should be modular, allowing easy updates and bug fixes.
- **Portability:** The system must run across different browsers and operating systems.
- **Scalability:** The system should be able to handle increasing loads by scaling the server or database capacity as needed.

Gantt Chart:



RMMM Plan for the Project

1. Risk Identification:

- **Technical Risks:**

- Incompatibility of 360-view feature with certain browsers/devices.
- Issues with hosting 3D models or large media files, leading to slow loading times.
- Website/App Downtime: The platform for virtual tours might crash or experience downtime.
- Virtual Tour Quality: Poor quality of virtual tours could lead to negative feedback and reduced user engagement.

- **Operational Risks:**

- Limited expertise in 360 virtual tour or 3D rendering technologies.
- Inadequate server resources to handle high traffic or data-heavy requests.

- **Security Risks:**

- Vulnerabilities in third-party plugins for 360 or 3D views.
- Data breaches due to insecure handling of sensitive customer or property information.

- **Project Management Risks:**

- Timeline delays due to unexpected technical issues or resource shortages.
- Miscommunication among team members about feature requirements.

2. Risk Analysis:

- **Impact on Project:**

- **High Impact:** Website downtime, security breaches, or poor performance can lead to a failure in user adoption.
- **Medium Impact:** Browser incompatibility or integration issues could limit accessibility but may have workarounds.
- **Low Impact:** Minor bugs or feature delays could affect the user experience but are generally manageable.

- **Likelihood:**

- **High Probability:** Incompatibility and slow load times are common in media-heavy websites.
- **Medium Probability:** Security issues, if proper measures aren't followed, could occur.
- **Low Probability:** Major breakdowns due to miscommunication or total failure of integration with the CMS.

Risk Mitigation Strategies:

1. Technical Risks:

- **Incompatibility of 360-view feature with certain browsers/devices:**
 - **Mitigation:** Test the 360 virtual tour on multiple browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile) during development.
 - **Backup Plan:** Provide an alternative image gallery or video walkthrough for unsupported browsers/devices.
- **Issues with hosting 3D models or large media files (slow load times):**
 - **Mitigation:** Optimize media files by compressing 360 images and videos, and using a Content Delivery Network (CDN) to speed up content delivery.

- **Backup Plan:** Implement lazy loading techniques where images load progressively as the user scrolls.
2. **Website/App Downtime:** Ensure regular maintenance and use a robust hosting service with scalable infrastructure.
- **Limited expertise in 360 virtual tour or 3D rendering technologies:**
 - **Mitigation:** Invest time in learning through tutorials and online courses. Leverage external libraries or plugins that simplify the creation of 360 virtual tours.
 - **Backup Plan:** Outsource the specific 360 or 3D rendering aspects to a freelancer if time or expertise becomes a bottleneck.
 - **Inadequate server resources to handle high traffic:**
 - **Mitigation:** Estimate resource needs early and choose a scalable hosting plan (such as cloud hosting with flexible bandwidth).
 - **Backup Plan:** Monitor site traffic closely post-launch and upgrade to higher capacity if needed.
3. **Security Risks:**
- **Vulnerabilities in third-party plugins:**
 - **Mitigation:** Only use trusted and frequently updated plugins. Regularly check for updates and patches. Run penetration tests on the site to identify weak points.
 - **Backup Plan:** Have a detailed security recovery plan, including backup and restoration procedures in case of breaches.
 - **Data breaches or insecure handling of sensitive information:**
 - **Mitigation:** Implement HTTPS, encryption for data storage, and secure authentication methods. Ensure proper access control to sensitive information.

- **Backup Plan:** Ensure daily backups of all user data and have a restoration plan ready in case of any breach.

4. Project Management Risks:

- **Timeline delays due to unexpected technical issues:**
 - **Mitigation:** Set realistic deadlines and build in buffer time for unexpected challenges. Regularly monitor progress and adjust timelines as needed.
 - **Backup Plan:** Prioritize core features (real estate listings, contact forms) in case secondary features (like 360 tours) face delays.
- **Miscommunication among team members:**
 - **Mitigation:** Hold regular team meetings, define clear roles, and use project management tools (e.g., Trello, Slack) to track tasks and updates.
 - **Backup Plan:** Create documentation for every phase of the project to ensure clarity and reference if communication breaks down.

3. Risk Monitoring and Control:

- **Risk Tracking:**
 - Maintain a risk register where each identified risk is logged, along with its likelihood and mitigation measures.
 - Set up automated tests for website responsiveness, load times, and browser compatibility.
- **Control Measures:**
 - Regularly test the 360 and 3D views on multiple platforms.
 - Perform security audits, particularly for third-party integrations.
 - Monitor hosting performance to avoid downtime during site launch.

4. Reporting:

- **Risk Status Updates:**

- Weekly team meetings to assess risk status, update the risk register, and address any new risks that have surfaced.

- **Reporting Channels:**

- Assign risk owners who will be responsible for updating stakeholders (project leads, mentors, or clients) about the progress or escalation of risks.

- **Documentation:**

- Create detailed reports after key milestones, showing risk mitigation actions and their outcomes.

5. Continuous Improvement:

- **Post-Project Review:**

- After project completion, conduct a review of the risk management process to identify lessons learned.

- **Feedback Loops:**

- Gather feedback from all stakeholders and team members to improve risk identification and control measures in future projects.

RIS Sheet:

Risk ID	Risk Description	Impact (High/Medium/Low)	Likelihood (High/Medium/Low)	Mitigation Strategy	Risk Owner	Status
R1	Incompatibility of 360-view feature on certain browsers/devices	High	Medium	Test on all browsers/devices, offer image galleries as a fallback.	Development Team	Ongoing
R2	Slow loading times due to large 3D models and media files	High	High	Optimize media, use CDN, apply lazy loading techniques.	Web Developer	Ongoing
R3	Website downtime due to server or hosting issues	High	Medium	Use a reliable hosting service, monitor uptime continuously, implement load balancing, and have backup servers ready.	Cloud Engineer	Planned

R4	Limited expertise in 360 or 3D rendering	Medium	Medium	Invest in learning/training, outsource specific tasks if necessary.	Project Lead	Ongoing
R5	Server unable to handle high traffic	High	Medium	Use scalable cloud hosting, monitor traffic closely, upgrade hosting as needed.	Cloud Engineer	Planned
R6	Vulnerabilities in third-party plugins	High	Medium	Use trusted plugins, run regular updates, and conduct security audits.	Security Lead	Ongoing

R7	Data breach due to insecure handling of sensitive information	High	Low	Use HTTPS, encrypt data, implement secure authentication methods, and have a recovery plan ready.	Security Lead	Planned
R8	Timeline delays due to technical issues	Medium	High	Set realistic deadlines with buffer time, prioritize core features.	Project Lead	Ongoing
R9	Miscommunication between team members	Medium	Medium	Regular meetings, use project management tools (Trello/Slack), document tasks and updates.	Project Lead	Ongoing

