Software Requirements Specification

for

<Project>

Version 1.0 approved

Prepared by <author>

<organization>

<05/02/2025>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Document Conventions

Headings follow a hierarchical numbering structure (1.1, 1.2, 2.1, etc.).

Priority levels are categorized as **High, Medium, or Low** to indicate their importance.

Functional requirements are denoted with "**REQ-#**" identifiers.

Bolded words and phrases highlight **key concepts, system components, and critical actions** within the document for better readability and emphasis.

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

The document is intended to the following stakeholders:

Developers: To allow them to understand system functionalities, database interactions, and API integrations.

Project Managers: To track feature implementation and overall project scope.

End-Users: To review all system capabilities and ensure alignment with expectations.

QA Testers: To validate requirements through test cases and system testing to ensure optimality.

Regulatory Bodies: To ensure compliance with Singapore's housing and data privacy regulations.

Recommended reading order:

1. **Introduction** (Overview of the system)
2. **Overall Description** (System context and user needs)
3. **System Features** (Detailed functional specifications)
4. **External Interface Requirements** (Integration details)
5. **Non-Functional Requirements** (Performance, security, and usability constraints)

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

- *Simplify the flat selection process by allowing users to filter flats based on budget, location, remaining lease, commute tolerance, and amenities.*
- *Provide insights using historical resale price data and plotting it onto a graph*
- *Allow easier user decision-making with side-by-side comparisons of different flats based off criteria above*

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

Designed to replace or complement existing property listing platforms like PropertyGuru. Unlike traditional property platforms, which primarily focus on listing properties, this system uses publicly available government data (e.g., from data.gov.sg) to provide personalized, data-driven recommendations for resale flats.

Limitations of existing property platforms include:

- *Lack of personalized recommendations based on user preferences (e.g., budget, commute tolerance, proximity to amenities).*
- *Limited historical data analysis to help users understand price trends and make informed decisions.*
- *Absence of side-by-side comparisons of resale flats based on user-defined criteria.*

While the system operates independently, it is designed to complement or replace existing property platforms by offering advanced features that are not currently available. It does not rely on other systems for its core functionality but integrates with external APIs to enhance its capabilities.

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

- *Recommendation of flats based off preferences*
- *Comparison feature between flats*
- *User Account Management to store preferences*

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

- *Homebuyers*
- *Description: Individuals or couples looking to purchase a resale flat, possibly for the first time.*

- *Characteristics:*
- *Frequency of Use: Moderate to high (frequent interaction as they explore various options before purchase).*
- *Subset of Functions Used: Account Login, flat recommendation, flat comparison*
- *Technical Expertise: Basic to moderate (may not be tech-savvy but comfortable using simple tools).*
- *Security Levels: Standard access to all features.*

- *Experience: Little to no experience in real estate transactions*
- *Education Level: General public with varying levels of education*

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

Hardware Platform:
The application will be designed to run on typical user devices such as desktop computers, laptops, and mobile devices. It will be optimized for:

- *Desktop and Laptops: Windows, macOS, and Linux platforms.*

- *Mobile Devices: Android and iOS devices*

Operating System:
The application will be fully compatible with the following operating systems for both development and deployment:

- *Development Environment: Windows 10/11, macOS, and Linux distributions (Ubuntu, Fedora, etc.)*
- *Server-side (for deployment): Linux-based systems (Ubuntu preferred) for backend deployment, as Flask and other Python-based services work well on these systems.*
- *Client-side (for use by end-users): Cross-platform support for Windows, macOS, and major browsers (Chrome, Firefox, Safari, Edge).*
-

Web Server & Deployment:

- *Web Server: The application will be hosted on a Flask-based backend, which can be deployed on a Linux server running Apache or Nginx.*
- *Database Server: The system will use MongoDB for storing non-relational data (e.g., user preferences, transaction history) and PostgreSQL or SQLite for relational data (e.g., pricing trends, flat data).*
-

Software Components:

- *Backend: Python 3.x (using Flask as the framework), along with libraries such as:*
- *PyMongo for MongoDB interaction*
- *SQLite3 for database management (local storage)*
- *Flask-Login for user authentication*
- *Flask-WTF for web forms and validation*

- *Frontend: HTML5, CSS3, and JavaScript (with frameworks/libraries like React or Vue.js if future dynamic interactions are needed).*

- *Database:*
- *MongoDB for storing unstructured data such as trends and user preferences.*
- *PostgreSQL for structured data such as pricing and flat information.*

- *API Integrations: The app will connect to public government data APIs (e.g., from data.gov.sg) for resale price trends and other relevant information*

Other Tools:

- *Version Control: GitHub for version control, collaboration, and deployment processes.*

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Agent Selection & Commission bidding

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

This feature allows sellers to select property agents based on both cost and quality by setting a maximum commission rate and enabling agents to compete for listings by offering lower commission rates. The system enhances transparency by displaying agent profiles, including their years of experience (YOE), past transactions, client reviews, qualifications, and historical commission trends for similar properties.

Priority: High - Essential for ensuring sellers receive professional assistance while maintaining competitive agent fees.

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

User Action: Seller inputs property details and sets a maximum commission rate.

System Response: Displays a list of agents with profiles, including experience, past property transactions reviews, and commission trends.

User Action: Multiple agents submit commission bids based on the seller's criteria.

System Response: The system updates and displays the best available offers in real time.

User Action: Seller reviews and shortlists agents based on cost, experience, and reviews.

System Response: Agents are notified when they are shortlisted and may submit a revised bid.

User Action: Seller selects the most suitable agent, and contact details are exchanged.

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

REQ-1: The system shall allow sellers to input property details and set a maximum commission rate.

REQ-2: The system shall enforce a minimum decrement for commission offers, preventing property agents from underbidding by negligible amounts (e.g., at least 0.01% lower than the previous offer).

REQ-3: The system shall allow multiple property agents to submit commission offers lower than the seller's maximum rate, with a minimum set

REQ-4: The system shall display agent profiles, including YOE, client reviews, qualifications, and past commission rates.

REQ-5: The system shall enable sellers to compare multiple commission bids in real time.

REQ-6: The system shall allow sellers to shortlist and compare agents based on price, experience, and reviews.

REQ-7: The system shall notify agents when they are shortlisted by a seller and allow them to submit revised bids.

REQ-8: The system shall allow the exchange of contact information between sellers and selected agents.

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 Agent Profile and Transparency

4.2.1 Description and Priority

This feature ensures transparency by providing comprehensive agent profiles, including their professional background, qualifications, and historical transaction data. Sellers can assess agents based on verifiable information before making a decision.

Priority: Medium - Important for ensuring seller confidence in agent selection.

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.2.2 Stimulus/Response Sequences

User Action: Seller browses agent profiles based on experience, past sales, and commission history.

System Response: Displays agent details, including transaction records and verified client reviews.

User Action: Seller filters and compares agents based on specific criteria.

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.2.3 Functional Requirements

REQ-1: The system shall maintain a database of agent profiles, including experience, qualifications, and sales history.

REQ-2: The system shall retrieve and display agent commission rates for past transactions.

REQ-3: The system shall allow sellers to filter agents based on commission trends, experience, and ratings.

REQ-4: The system shall enable verified clients to leave reviews and ratings for agents after sales is completed.

4.3 User Account & Preference Storage (which includes reviews and ratings)

4.3.1 Description and Priority

This feature allows users to create accounts, save preferences, and store selected agents for future reference. Sellers can track agent interactions, while agents can manage their commission offers and respond to potential clients.

Priority: High - Enhances user convenience and system usability.

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.3.2 Stimulus/Response Sequences

User Action: User registers for an account and selects their role as either a Seller, Property Agent, or Buyer.

System Response: Stores the selected role and customizes the user experience based on the role type.

If Seller:

User Action: Seller lists a property by providing relevant details such as location, price, and property type.

User Action: Seller sets a maximum commission rate for property agents.

System Response: The listed property becomes visible to property agents.

User Action: Seller views all the Property agent that bid for the property and selects the one that resonates with him best

If Property Agent:

User Action: Property agent views listed properties that are open for bidding.

User Action: Property agent submits a bid for a property based on commission rates.

System Response: The system updates the bid list in real time and notifies the seller of new bids.

If Buyer:

User Action: Buyer views listed properties and can filter based on preferences (e.g., price, location, flat type).

System Response: The system displays available properties matching the buyer's criteria.

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.3.3 Functional Requirements

REQ-1: The system shall allow users to create and manage accounts.
REQ-2: The system shall require users to select their role (Seller, Property Agent, or Buyer) during registration.
REQ-3: The system shall customize the user experience based on the selected role.

REQ-4: The system shall enable sellers to save preferred agents and track interactions.

REQ-5: The system must allow users to submit and update agent reviews and ratings

REQ-6: The system shall ensure that reviews and ratings are linked to verified transactions.

REQ-7: The system shall allow users to retrieve and modify their saved preferences at any time.

REQ-8: The system shall allow property agents to update their profiles, including experience, past transaction, qualifications, and commission rates.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

- *The system should retrieve and display property listings and agent profiles within 3 seconds to ensure an optimal user experience.*
- *The commission bidding leaderboard should update in real-time, with no more than a 2-second delay between bid submission and display.*
- *The system should process and store agent commission bids within 2 seconds of submission.*
- *Search and filtering operations for agents and properties should not exceed 4 seconds, even under peak load.*
- *Notifications to agents about being shortlisted should be delivered within 2 seconds to maintain competitive bidding dynamics.*

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

- *The system shall provide disclaimers informing users that commission offers are non-binding until a formal agreement is signed, preventing misunderstandings or disputes.*
- *The system shall alert users about potential risks of choosing agents solely based on the lowest commission rate, encouraging consideration of experience and reviews.*
- *Automated backups shall be performed daily to prevent data loss, with backups stored securely and retained for at least 30 days.*
- *The system shall implement data validation checks to prevent submission of invalid bids (e.g., negative commission rates or excessively low values).*
- *In case of system failure or downtime, the system shall ensure no data corruption or loss of ongoing bids or transactions.*
- *The system shall comply with relevant data protection regulations (e.g., PDPA in Singapore) to safeguard user information and prevent unauthorized access.*

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

- *For account creation, the system shall require user authentication through secure methods (e.g., email verification and two-factor authentication) for all users, including sellers, property agents, and buyers.*
- *For property agents, their account creation will involve an additional step where their credentials will be checked against the list of agents provided from ACEAS*
- *All user data, including login credentials and preferences, must be encrypted using industry-standard encryption methods.*
- *The system must follow Singapore's Personal Data Protection Act (PDPA) guidelines to ensure proper handling of user data.*

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

- *Usability: The system shall provide an intuitive and user-friendly interface, allowing new users to navigate and perform actions (e.g., placing a bid, shortlisting agents) within 3 clicks or taps.*
- *Availability: The system shall maintain an uptime of at least 99.9%, ensuring continuous access for users except during scheduled maintenance.*
- *Reliability: The system shall ensure accurate bid processing and data consistency, even in case of system failures or interruptions.*
- *Maintainability: The system should be modular to facilitate easy updates and enhancements.*
- *Interoperability: The system shall integrate seamlessly with third-party APIs (e.g., for property data) while maintaining data consistency.*
- *Scalability: The system should be designed to handle a growing number of users and an expanding database of resale flat transactions without significant degradation in performance.*

- *Testability: The system shall support automated testing to verify core functionalities before deployment.*
- *Portability: The system shall be compatible with all major web browsers (Chrome, Firefox, Safari, and Edge)*

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

- *Users must create an account and be authenticated before submitting reviews or ratings on flats.*
- *Only verified users can submit more than one review per flat to prevent spam or manipulation.*
- *Flat recommendations must be based on publicly available government data from data.gov.sg and should be updated biannually.*
- *The system should not allow recommendations to be influenced by paid promotions to maintain fairness and transparency.*
- *Users can compare up to 5 different flats at once in the price comparison feature.*
- *The system should adhere to all data privacy and security regulations set by Singaporean authorities.*

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Data Dictionary

SR NO	TERMS	DEFINITION
1.	Resale Flat	A previously owned Housing and Development Board (HDB) flat that is being sold by its current owner rather than directly from the HDB
2.	Homebuyers	Individuals or couples actively searching for resale flats.
3.	User Account Management	A system feature that allows users to manage their account credentials, housing preferences (budget, flat size, amenities, etc.), and saved flats.
4.	Remaining Lease	The number of years left on the flat's lease before it expires. The system accounts for this while

		recommending flats as it affects affordability and resale value.
5.	Commute Tolerance	The maximum travel time or distance the user is willing to commute to and from their workplace The system considers this while recommending flats.
6.	Amenities	Facilities and services available near the resale flat such as: schools, polyclinics, gyms, supermarkets.
7.	Resale Price Trends	The fluctuation of property prices over time which enables users to estimate whether a flat pricing is justifiable.
8.	Agent	An agent refers to a real estate professional who assists property buyers and sellers in transactions by helping sellers list properties, providing expert advice, and negotiating commission rates through a bidding system.
9.	Bid Processing	A system that ensures quick processing and updating of property bids in real-time.
10.	Verified Transactions	A method to provide credibility for a completed transaction by a user upon which the user will be able to submit a review.
11.	User Authentication	A process that ensures the accurate identity of a user and doesn't allow for impersonation. It is implemented by using login credentials and security verification methods.

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

Source: http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc