Software Requirements Specification

for

<House Buying Guide>

Version 1.0 approved

Prepared by <Rui Yang>

<Meatballs>

<28/08/2020>

Table of Contents

Table of Contents ii
Revision History ii
1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 1
1.5 References 1
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics 2
2.4 Operating Environment 2
2.5 Design and Implementation Constraints 2
2.6 User Documentation 2
2.7 Assumptions and Dependencies 3
3. External Interface Requirements 3
3.1 User Interfaces 3
3.2 Hardware Interfaces 3
3.3 Software Interfaces 3
3.4 Communications Interfaces 3
4. System Features 4
4.1 System Feature 1 4
4.2 System Feature 2 (and so on) 4
5. Other Nonfunctional Requirements 4
5.1 Performance Requirements 4
5.2 Safety Requirements 5
5.3 Security Requirements 5
5.4 Software Quality Attributes 5
5.5 Business Rules 5
6. Other Requirements 5
Appendix A: Glossary 5
Appendix B: Analysis Models 5
Appendix C: To Be Determined List 6

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

<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

<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.>

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.>

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.>

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.>

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.>

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

4.1 Search historical data

4.1.1 Description and Priority

Retrieve resale/private sale data according to user-defined search criteria (residential type, location, type of housing, floor level, lease information, square footage, price range (total), price range ($psf), sale type, project name). This use case is of highest priority.

4.1.2 Stimulus/Response Sequences

1. User selects option to *search historical data*, and may select to search as a buyer or seller
2. If buyer is selected, System displays parameters of sale type, residential type, location, type of housing, floor level, lease information, square footage, price range (total), and price range ($psf)
3. If seller is selected, System displays parameters of project name, residential type, location, type of housing, floor level, lease information, square footage, price range (total), and price range ($psf)
4. If neither is selected, System displays parameters of residential type, location, type of housing, floor level, lease information, square footage, price range (total), and price range ($psf)
5. System requests the input of criteria to filter historical data
6. **User** enters one, several or all criteria to filter historical data
7. System searches through the **Server database** for historical data matching input criteria
8. If matching historical data is retrieved, System displays the data to **User**
9. If **User** is logged in, System runs *store search/request history*

4.1.3 Functional Requirements

REQ-1 The user must be able to input one, several or all criteria into the query system.

REQ-2 By default, search history must be displayed in

REQ-2.1 From top row to bottom row: In descending order of date

REQ-2.2 Apart from this, the system must include an option to allow the user to change display order by date, category and input parameters.

REQ-3 The system must display dates in dd/mm/yyyy hh:mm:ss format.

REQ-4 If the system cannot find data matching search criteria selected in database, the system must display an error message “No matching results”, and prompt the user to enter another set of criteria

REQ-4 If the server database is inaccessible or empty, the system must display an error message “Server offline” and return the user to the homepage.

4.2 Request financing guide

4.2.1 Description and Priority

Suggests a financing plan based on user input parameters of price housing, current available funds, desired monthly repayment and desired repayment duration. This use case is of medium priority.

4.2.2 Stimulus/Response Sequences

1. User selects option to *request financing guide*
2. System requests the input of at least 3 out of 4 parameters
3. User enters at least 3 out of 4 parameters
4. System passes parameters to Server Database
5. Server Database returns financing guide (loan amount, unentered parameter) to system
6. System displays financing guide to User
7. System runs *store search/request history* if User is logged in.

4.2.3 Functional Requirements

REQ-1 The system must display an error message “Insufficient information for calculation” if the user enters less than three out of four of the parameters.

REQ-2 If the server database is inaccessible or empty, the system must display an error message “server offline” and return the user to the homepage.

REQ-3 The system must display an error message “Login credentials incorrect” if the username and password are not verified.

4.3 Request price prediction

4.3.1 Description and Priority

Returns a predicted price for a housing unit in the future. This use case is of high priority.

4.3.2 Stimulus/Response Sequences

1. User makes a query to the system
2. System verifies that an appropriate query has been made by User
3. System returns a price prediction
4. If User is logged in, System runs *store search/request history*

4.3.3 Functional Requirements

REQ-1 The system must only accept user queries that contain at least one parameter among location, housing type, square area, floor level and lease information.

REQ-1.1 The system must display an error message “Invalid query. Please try again.” if invalid query made by the user.

REQ-2 The system must:

REQ-2.1 Display a predicted price only if the server returns a confidence score of at least 80%.

REQ-2.2 Display an error message “Unable to predict price” as long as the server returns a confidence score lower than 80%.

REQ-3 The system must display an error message “Unable to record query.” if the server database is unable to record price prediction.

REQ-4 The system must display an error message “Server offline” and return the user to the homepage if the server database is inaccessible or empty.

4.4 View search history

4.4.1 Description and Priority

Displays history of all price prediction queries, request of financial planning and historical data search request. This use case is of low priority.

4.4.2 Stimulus/Response Sequences

1. User selects option to view search history
2. System pulls user search history data from Server database
3. System displays search result to User

4.4.3 Functional Requirements

REQ-1 Search history must store up to 100 queries and requests per user.

REQ-2 By default, search history must be displayed in

REQ-2.1 From top row to bottom row: In descending order of date and ascending order of row number.

REQ-2.2 From left most column to right column: row number, date, category, input parameters.

REQ-2.3 Apart from this, the system must include an option to allow the user to change display order by date, category and input parameters.

REQ-3 The system must display dates in dd/mm/yyyy hh:mm:ss format and input parameters separated by a comma.

REQ-3.1 "-" is displayed if input parameters are not applicable.

REQ-4 The system must display most recent searches in a drop down menu such that the user is able to search again using the same parameters if needed.

REQ-5 User must be able to filter results based on categories of request price prediction, search historical data and request a financial plan.

REQ-6 The system must display an error message "Function not available. Please log in to continue" and return to the login page if the user is not logged in.

REQ-7 The system must display an error message "No search history for <username> found" and return to the homepage if user has no search history.

REQ-8 The system must display an error message "Server offline" and return the user to the homepage if the server database is inaccessible or empty.

4.5 Store search/request history

4.5.1 Description and Priority

Stores the search/request history of the user such that they may repeat the same request in the future without reentering input parameters. This use case is of low priority.

4.5.2 Stimulus/Response Sequences

1. User makes a *search historical data, request price prediction,* or *request financing guide* request to the System.
2. System passes search/request parameters to Server Database
3. Server Database stores search/request parameters, and associates the parameters with the user

4.5.3 Functional Requirements

REQ-1 The system must verify that the user is logged in every time the *store search/request history* function is run.

REQ-2 Search history must store up to 100 queries and requests per user.

REQ-3 Store function must be done automatically whenever a successful request price prediction, request financing guide or search historical data is made by the user.

REQ-4 Stored data must be located in a .csv file.

REQ-5 System must be able to store the exact same input parameters in the exact same format with the same search done.

REQ-6 Data stored must be specific to the search or request made by the user.

REQ-6.1 If *request price prediction* is made, an entry containing the date the query was made, price predicted and the parameters input must be stored in the database.

REQ-6.2 If *request financing guide* is made, an entry containing the date the query was made, the parameters input, and the financing plan output values must be stored in the database.

REQ-6.3 If *search historical data* is made, an entry containing the date the query was made and the parameters input must be stored in the database.

REQ-6.4 All date entries must be recorded in dd/mm/yyyy hh:mm:ss format.

REQ-6 If the server database is inaccessible or empty, the system must display an error message "server offline" and return user to the homepage

4.6 Log in

4.6.1 Description and Priority

Enables the user to log into the system. This use case is of medium priority.

4.6.2 Stimulus/Response Sequences

1. User selects option to *log in*
2. System requests the input of an email and password
3. User enters email and password
4. System verifies that email and password are in the Server Database
5. Once email and password are verified, System logs in the user and displays a success message

4.6.3 Functional Requirements

REQ-1 The system must include an option for the user to reset password during login.

REQ-2 The system must include an option for the user to sign up for an account during login.

REQ-3 The system must display an error message "Login credentials incorrect" if the username and password are not verified.

REQ-3.1 The system must accept a maximum of 5 attempts at inputting username and password, after which the user must be given a 5-minute timeout.

REQ-4 If the server database is inaccessible or empty, the system must display an error message "server offline" and return the user to the homepage.

4.7 Signup

4.7.1 Description and Priority

Allows User to sign up for an account using a username and password. This use case is of low priority.

4.7.2 Stimulus/Response Sequences

1. User selects option to *signup*
2. System requests the input of an email, a name and a password
3. User enters a valid email, a name and a password
4. System passes the email and password to the Server Database
5. If the email is not in use, Server Database creates a new user with the input email and password
6. System displays a success message “Account created! Please login from the homepage”
7. System returns User to the homepage

4.7.3 Functional Requirements

REQ-1 System must only accept:

REQ-1.1 Unique emails (not already present in server database).

REQ-1.2 Passwords that contain at least one of each of the following: a lowercase alphabet, an uppercase alphabet, a number and a symbol.

REQ-1.3 Passwords that contain between 8-21 characters.

REQ-2 If the email entered already exists in the server database, the system must display an error message “Email already in use” and request user to input another email

REQ-3 The system must verify that:

REQ-2.1 The user enters a password twice;

REQ-2.2 The two passwords entered are matching.

REQ-4 If server database is inaccessible or empty, system must display an error message “Server offline” and return user to the homepage

4.8 Reset password

4.8.1 Description and Priority

Allows User to reset password for their account

4.8.2 Stimulus/Response Sequences

1. User selects option to reset password
2. System requests the input of an email
3. User enters email
4. System verifies that email is in the Server Database
5. If the email is valid, System displays a success message “We have sent an email to your provided email address with password reset information.”
6. System sends an email to provided email address with a link to reset user password

7. User opens email and clicks on provided link to reset password
8. System requests the input of a new password
9. User enters a new password
10. System passes password to Server Database
11. Server Database stores new password associated with the email
12. System displays a success message “password reset! please login from the homepage”
13. System returns User to the homepage

4.8.3 Functional Requirements

REQ-1 System must only accept:

REQ-1.1 Passwords that contain at least one of each of the following: a lowercase alphabet, an uppercase alphabet, a number and a symbol.

REQ-1.2 Passwords that contain between 8-21 characters.

REQ-2 The system must verify that:

REQ-2.1 The user enters a password twice;

REQ-2.2 The two passwords entered are matching.

REQ-2 If the entered email is not in the server database, the system must display an error message “Email not found” and request for another input of an email.

REQ-3 If the server database is unable to verify username or record new password, system must display “server offline” and return user to the homepage.

REQ-4 If server database is inaccessible or empty, system must display an error message “Server offline” and return user to the homepage

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 The system must respond within 5 seconds once the username and password are entered into the login portal.

5.1.2 All accounts created must be ready for access within 10 seconds upon successful registration.

5.1.3 All query results for price must be returned within 5 seconds.

5.2 Safety Requirements

5.2.1 Application does not retrieve any crucial information that will compromise user's privacy such as NRIC, Address, Birthdate, etc.

5.2.2 Passwords will not be encrypted (TBD).

5.3 Security Requirements

5.3.1 The user must input their email and password in order to register for an account.

5.3.2 All emails must be verified for validity based on whether they contain '@' and '.' in its domain.

5.3.3 The system must accept a maximum of 5 attempts at inputting username and password, after which the user must be given a 5-minute timeout.

5.3.4 When resetting passwords, the user must enter a valid username within 8 tries. The account must be locked for 24 hours upon failing all 8 tries.

5.4 Software Quality Attributes

5.4.1 The server database must be refreshed on 15th of every month by batch drop.

5.4.1.1 *"Unable to update server database"* must be displayed upon failing to successfully retrieve new dataset and update database.
5.4.1.2 *"No updates"* must be displayed if there is no new dataset found.
5.4.1.3 *"Server offline"* must be displayed upon querying all empty database

5.4.2 The system must return the same output provided the same input parameters every time unless there is an update to the server database after refresh (see 5.4.1).

5.4.3 Full system functionality must be restored within 5 minutes of a system reboot.

5.4.4 All messages and information displayed must be in English.

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.>

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.>

Appendix A: Glossary (Data dictionary)

Term	**Definition**
Actors	
User	A buyer or seller who utilizes the System
Server Database	The back-end system that stores and manages data
Search Parameters:	
Residential Type	Private/public housing
Location	Address, postal district or township
Type of Housing	Number of rooms in resale flat, type of private housing (Strata Detached/Strata Semi-detached/Strata Terrace/Detached/Semi-detached/Terrace/Apartment/Condominium/Executive Condominium)
Floor Level	Storey at which resale flat is located, number of storeys in private housing
Lease Information	Tenure left on property, to nearest month
Square Footage	Land area of property in square feet
Price Range (Total)	Price of property in dollars
Price Range ($psf)	Price of property in dollars per square foot

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

1. User log in and sign up can be done directly using gmail instead of an email input
2. Address, postal code, and postal district are acceptable parameters for a location based search parameter

<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