Software Requirements Specification

for

SimplyStay

Version 1.0 approved

Prepared by JENTS

SC2006 Software Engineering

22/02/2024

Table of Contents

[1. Introduction 1](#_Toc159535185)

[1.1 Purpose 1](#_Toc159535186)

[1.2 Document Conventions 1](#_Toc159535187)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc159535188)

[1.4 Product Scope 2](#_Toc159535189)

[1.5 References 2](#_Toc159535190)

[2. Overall Description 3](#_Toc159535191)

[2.1 Product Perspective 3](#_Toc159535192)

[2.2 Product Functions 3](#_Toc159535193)

[2.2.1 Login Page 3](#_Toc159535194)

[2.2.2 Search Page 3](#_Toc159535195)

[2.2.3 Calculator Page 3](#_Toc159535196)

[2.2.4 Favourites Page 4](#_Toc159535197)

[2.2.5 Comparison Page 4](#_Toc159535198)

[2.3 User Classes and Characteristics 4](#_Toc159535199)

[2.3.1 Users eligible to purchase a BTO flat 4](#_Toc159535200)

[2.3.2 Users ineligible to purchase a BTO flat 5](#_Toc159535201)

[2.4 Operating Environment 6](#_Toc159535202)

[2.4.1 Production Environment 6](#_Toc159535203)

[2.4.2 Development Environment 6](#_Toc159535204)

[2.5 Design and Implementation Constraints 6](#_Toc159535205)

[2.6 User Documentation 7](#_Toc159535206)

[2.7 Assumptions and Dependencies 7](#_Toc159535207)

[3. External Interface Requirements 7](#_Toc159535208)

[3.1 User Interfaces 7](#_Toc159535209)

[3.1.1 Main Page 7](#_Toc159535210)

[3.1.2 Login Page 8](#_Toc159535211)

[3.1.3 Search Page 8](#_Toc159535212)

[3.1.4 Calculator Page 9](#_Toc159535213)

[3.1.5 Comparison Page 9](#_Toc159535214)

[3.1.6 Favourites Page 10](#_Toc159535215)

[3.2 Hardware Interfaces 10](#_Toc159535216)

[3.2.1 Client-side Requirements 10](#_Toc159535217)

[3.2.2 Server-side Requirements 11](#_Toc159535218)

[3.3 Software Interfaces 11](#_Toc159535219)

[3.4 Communications Interfaces 11](#_Toc159535220)

[4. System Features 11](#_Toc159535221)

[4.1 Registration 11](#_Toc159535222)

[4.1.1 Description and Priority 11](#_Toc159535223)

[4.1.2 Stimulus/Response Sequences 12](#_Toc159535224)

[4.1.3 Functional Requirements 16](#_Toc159535225)

[4.2 Login 17](#_Toc159535226)

[4.2.1 Description and Priority 17](#_Toc159535227)

[4.2.2 Stimulus/Response Sequences 17](#_Toc159535228)

[4.2.3 Functional Requirements 20](#_Toc159535229)

[4.3 Searching 20](#_Toc159535230)

[4.3.1 Description and Priority 20](#_Toc159535231)

[4.3.2 Stimulus/Response Sequences 20](#_Toc159535232)

[4.3.3 Functional Requirements 23](#_Toc159535233)

[4.4 Favourite Houses 24](#_Toc159535234)

[4.4.1 Description and Priority 24](#_Toc159535235)

[4.4.2 Stimulus/Response Sequences 24](#_Toc159535236)

[4.4.3 Functional Requirements 26](#_Toc159535237)

[4.5 Calculator 27](#_Toc159535238)

[4.5.1 Description and Priority 27](#_Toc159535239)

[4.5.2 Stimulus/Response Sequences 27](#_Toc159535240)

[4.5.3 Functional Requirements 29](#_Toc159535241)

[4.6 Comparison 29](#_Toc159535242)

[4.6.1 Description and Priority 29](#_Toc159535243)

[4.6.2 Stimulus/Response Sequences 29](#_Toc159535244)

[4.6.3 Functional Requirements 31](#_Toc159535245)

[4.7 Maps 32](#_Toc159535246)

[4.7.1 Description and Priority 32](#_Toc159535247)

[4.7.2 Stimulus/Response Sequences 32](#_Toc159535248)

[4.7.3 Functional Requirements 33](#_Toc159535249)

[4.8 Logout 34](#_Toc159535250)

[4.8.1 Description and Priority 34](#_Toc159535251)

[4.8.2 Stimulus/Response Sequences 34](#_Toc159535252)

[4.8.3 Functional Requirements 36](#_Toc159535253)

[5. Other Nonfunctional Requirements 36](#_Toc159535254)

[5.1 Performance Requirements 36](#_Toc159535255)

[5.2 Security Requirements 36](#_Toc159535256)

[5.3 Software Quality Attributes 37](#_Toc159535257)

[5.3.1 Maintainability 37](#_Toc159535258)

[5.3.2 Scalability 37](#_Toc159535259)

[5.3.3 Usability 37](#_Toc159535260)

[5.3.4 Availability 37](#_Toc159535261)

[5.3.5 Reliability 37](#_Toc159535262)

[6. Other Requirements 37](#_Toc159535263)

[Appendix A: Data Dictionary 38](#_Toc159535264)

[Appendix B: Analysis Models 39](#_Toc159535265)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements specifications of the SimplyStay web application system. It will include the functional and non-functional requirements, its user interfaces, use case descriptions, as well as its constraints and dependencies.

## Document Conventions

|  |  |
| --- | --- |
| Font | Arial |
| Line Spacing | 1.5 |
| Text Alignment | Full Justification |
| Heading 1 | Bold, size 18 |
| Heading 2 | Bold, size 16 |
| Heading 3 | Bold, size 14 |
| Content | Size 12 |
| Technical Standards | IEEE 830-1998 |

Please refer to Appendix A (Data Dictionary) for a list of definitions and special terms used in this document.

## Intended Audience and Reading Suggestions

This document is intended to be read by the SimplyStay development team and SimplyStay users. However, external audiences interested in the SimplyStay website are welcome to read it as well.

This document starts with **Section 1: Introduction**, outlining the purpose and scope of the website. Next, **Section 2: Overall Description** gives an overall description of the website, explaining its functions and constraints, as well as its assumptions and dependencies. After that, **Section 3: External Interface Requirements** elucidates the user interfaces of the website. This is followed by **Section 4: System Features** and **Section 5: Other Nonfunctional Requirements** highlighting the functional and non-functional requirements and its corresponding use case descriptions. The document concludes with **Appendix A and B** detailing the Data Dictionary and Analysis Models respectively.

All audiences are encouraged to read Section 1 to gain a brief overview of the website and the SRS document, as well as Appendix A to famillarise themselves with the terminology used in the document. The SimplyStay development team are highly encouraged to read the entirety of this document in sequential order. SimplyStay users may read Section 3 to gain a better understanding of the website’s user interface.

## Product Scope

In Singapore, buying public housing for the first time can be a daunting task for young adults. As such, we have elected to create a convenient and intuitive website, called SimplyStay, to help young Singaporeans ease the process of finding their dream home.

Our target demographic are young Singaporeans looking to purchase public housing in Singapore. While we are specifically targeting age groups of 20-35, Singaporeans of any age are welcome to use the website as well.

## References

* "IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984, vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205.
* Software Requirements by Karl Wiegers and Joy Beatty

# Overall Description

## Product Perspective

The SimplyStay website is a new self-contained product. It is inspired by other websites that aggregate car prices for Singaporeans, and we aspire to not only achieve that functionality for BTO flats but add value by aggregating other types of information pertinent to housing, such as the distance to public transport infrastructure.

<INSERT SYSTEM ARCHITECTURE DIAGRAM HERE>

## Product Functions

The functions are summarised with respect to the website’s different pages.

### Login Page

* Users can register for an account using their email address and a password
* Users can login using their login credentials

### Search Page

* Users can search for houses by different filters such as price, location, housing type, and floor area
* The page will return all the houses that satisfy the user’s chosen filters

### Calculator Page

* Users can input their CPF, savings, and disposable income into the calculator
* The page will recommend payment plans and their duration based on the user’s input

### Favourites Page

* Logged in users can bookmark and remove their favourite houses
* Guest users will not be able to access this page, and will be notified by the website as such

### Comparison Page

* Users can select two houses from the list of houses available
* The page will display the statistics of the selected houses and highlight superior and inferior statistics in green and red respectively

## User Classes and Characteristics

The user classes are ranked in order of importance.

### Users eligible to purchase a BTO flat

|  |  |
| --- | --- |
| Aspect | Description |
| Frequency of Use | High |
| Subset of Product Functions Used | All |
| Technical Expertise | High |
| Characteristics | These will be people who are aged 18 to 35 and are married, or unmarried people aged 35 and above.  As the primary target group, they will use the website regularly and register an account on the website.  Their technical expertise is expected to be high, that they can comfortably navigate modern websites on their desktops and know what details to look out for in purchasing a BTO flat. |

### Users ineligible to purchase a BTO flat

|  |  |
| --- | --- |
| Aspect | Description |
| Frequency of Use | Low |
| Subset of Product Functions Used | Search Page  Comparison Page |
| Technical Expertise | Low |
| Characteristics | These will be people who do not fit the requirements to purchase a BTO flat or are current homeowners curious about the housing market.  They may visit the website occasionally from time to time and will likely be guest users.  Their technical expertise may vary, although it is expected that they can comfortably navigate modern websites on their desktops. |

## Operating Environment

### Production Environment

|  |  |
| --- | --- |
| Operating System | Microsoft Windows 7.0 and above  macOS Sierra and above |
| Web Browsers | Google Chrome, Mozilla Firefox, Microsoft Edge, browsers that support HTML5, CSS3, and Javascript |

### Development Environment

|  |  |
| --- | --- |
| Development environment | Description |
| Front-end: React.js | TBD |
| Back-end: TBD | TBD |
| Database: Firebase | TBD |
| Hosting service: TBD | TBD |

## 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).>

## User Documentation

For the SimplyStay development team or any developers interested in maintaining and continuing the project, there will be README.md files to guide them in setting up the development environment for the website.

<INSERT README FILES LINK HERE>

## 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).>

# External Interface Requirements

## User Interfaces

### Main Page

|  |  |
| --- | --- |
| Image | Description |
| A person and person sitting on a couch looking at a computer  Description automatically generated | This will be the page that greets users when entering the website for the first time. They will be able to navigate to the various pages on the website. |

### Login Page

|  |  |
| --- | --- |
| Image | Description |
| A screenshot of a login page  Description automatically generated | Users can log in using their email or password. If they do not have an account, they can choose to register for one. |

### Search Page

|  |  |
| --- | --- |
| Image | Description |
| A person and person sitting on a couch looking at a computer  Description automatically generated | Users can search for BTO flats using the following filters: Price, Location, Housing Type, and Floor Area. |

### Calculator Page

|  |  |
| --- | --- |
| Image | Description |
| A screenshot of a calculator  Description automatically generated | Users can input their CPF, Savings, and Disposable Income to calculate the type and duration of payment plans they can use. |

### Comparison Page

|  |  |
| --- | --- |
| Image | Description |
| A screenshot of a phone  Description automatically generated | Users can select two houses and compare their statistics. |

### Favourites Page

|  |  |
| --- | --- |
| Image | Description |
| A screenshot of a social media post  Description automatically generated | Logged in users can view their bookmarked houses. |

## Hardware Interfaces

### Client-side Requirements

All devices supporting browsers specified in **Section 2.4.1: Production Environment** are compatible with the website.

### Server-side Requirements

<TBD>

## 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.>

## 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.>

# System Features

## Registration

### Description and Priority

|  |  |
| --- | --- |
| Description | First time users can register for an account |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 001 | | |
| Use Case Name: | Registration | | |
| Created By: | Nigel Paliath | Last Updated By: | Nigel Paliath |
| Date Created: | 5 February 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Database |
| Preconditions: | 1. The user must be connected to the Internet. 2. The user does not have an account prior to registration. 3. The user has navigated to the login interface |
| Postconditions: | 1. The user successfully registered an account with their email and password and their account is added into the system database.   OR   1. The user is notified of the reason(s) why the registration was unsuccessful. |
| Frequency of Use: | Low |
| Flow of Events: | 1. At the homepage of the website, the user can click onto the “Login” button and the system will redirect the user to the login page.  2. Since the user is a new user, he would need to register for a new account.  3. The user can click on “Sign Up” and he will be redirected to the registration page.  4. The user would then need to input a valid email address, username, birth date, a valid mobile phone and all other respective fields.  5.The user will need to input a valid password that contains at least 8 characters which includes an upper-case, lower-case letter, numerical digits and a special character.  6. The user will also need to input the password once more to confirm his password.  7. At the end of the registration form, the user would need to check the checkbox of “I agree to the Terms of Use and Privacy Policy”.  8. The user will click on the “Sign up” button to register their account.  9. The system validates if there is an identical email existing in the system  10. The system will verify if the password satisfies all requirements  11. Upon verification, the system will store all information in the database securely.  12. Once registration is successful, the system will automatically help the user to log into his account. |
| Alternative Flows: | AF-S5: The user inputs a password that does not satisfy all the requirements set.  1. The system displays the message “Password does not meet all requirements, please try again!” under the password field.  2. The system returns to Step 5 and waits for inputs from the user.  AF-S6: The user inputs a mismatched password.  1. The system displays the message “Passwords do not match, please try again!” below the password field.  2. The system returns to Step 5 and waits for inputs from the user.  AF-S7: The user did not check the checkbox of “I agree to the Terms of Use and Privacy Policy.”  1. Upon clicking on the “Sign up” button, the system will display the message “Please check the checkbox for acknowledging the Terms of Use and Privacy Policy!” at the top of the registration form.  2. The system will return to Step 4 and wait for input from the user.  AF-S8: The user did not complete all of the input fields.  1. Upon clicking on the “Sign up” button, the system will display the message “Please check that all fields are filled up!” at the top of the registration form  2. The system will return to Step 4 and wait for user to complete all inputs.  AF-S9: The user inputs an email address that had already been registered.  1. The system displays the message “Email address has already been registered, please input another email address!” under the email address field.  2. The system returns to Step 4 and waits for input from the user. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. User must be able to register for an account in the system

1.1. The system must provide two input fields for the user

1.1.1. One of the input fields must be the user’s email address

1.1.2. One of the input fields must be the user's password

1.2. The user must fill in all the input fields before registering

1.3. The system must verify that the inputted information is correct

1.3.1. The system must verify that the email address is valid

1.3.2. The system must verify that the email address has not been registered

1.3.3. The password must contain at least 8 characters

1.3.4. The password must contain at least an uppercase letter

1.3.5. The password must contain at least a lowercase letter

1.3.6. The password must contain at least a number

1.3.7. The password must contain at least a special character

1.3.8. The system must notify the user if any of the above is not met

1.4. The system must create an account for the user once verification is completed

1.5. The system must direct the user to the main page upon completion

## Login

### Description and Priority

|  |  |
| --- | --- |
| Description | The user can login to his/her account with their correct credentials that they inputted |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 002 | | |
| Use Case Name: | Login | | |
| Created By: | Leow Zheng Jie | Last Updated By: | Leow Zheng Jie |
| Date Created: | 22 Februrary 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), SimplyStay Website, Database |
| Description: | The user can login to his/her account with their correct credentials that they inputted |
| Preconditions: | 1. The user must be connected to the Internet.  2. The user has a registered account. |
| Postconditions: | 1. The User has successfully logged into his/her own account.  OR  2. The user is notified of the reason(s) why he is unable to login into his account |
| Frequency of Use: | High |
| Flow of Events: | 1. At the main page, the user can click onto the “Login” button and the system will redirect the user to the login page.  2. At the login page, the system requests the input of both the email address and password.  3. The user inputs their registered email address and his password.  4. The user clicks on the “Login” button  5. The system verifies the credentials provided with the database.  6. If the Email and Password are correct and verified, the user will be directed into his account. |
| Alternative Flows: | AF-S4: The user left the email or password input field blank.  1. Upon clicking onto the “Login” button, the system displays the following message “Please ensure that you entered an email address or password!”.  2. The system will prompt the user to fill up all the fields.  3. The system returns to Step 3 and waits for the user to fill in all the fields.  AF-S5: The user inputs an incorrect email address or password.  1. Upon clicking onto the “Login” button, the system displays the following message: “Invalid email address and/or password!” at the top of the login page   2. The system returns to Step 3 and waits for the user to fill in the fields again. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. The user must be able to login using their credentials

1.1. The system must provide two input fields for the user

1.1.1. One of the input fields must be the user’s email address

1.1.2. One of the input fields must be the user's password

1.2. The user must fill in all the input fields before logging in

1.3. The system must verify that the inputted information is correct

1.3.1. The system must verify that the email address is valid

1.3.3. The password must be tied to the email address

1.3.8. The system must notify the user if any of the above is not met

1.4. The system must log in the user once verification is completed

1.5. The system must direct the user to the main page upon completion

## Searching

### Description and Priority

|  |  |
| --- | --- |
| Description | Users can search for flats within a specified requirement. Requirements include price range, geographic area, housing type and floor area. Users should also be able to filter flats based on the fields mentioned. |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 003 | | |
| Use Case Name: | Searching | | |
| Created By: | Enric Tan | Last Updated By: | Enric Tan |
| Date Created: | 6 February 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Database |
| Preconditions: | 1. The user must be connected to the Internet. 2. The user has navigated to the search page |
| Postconditions: | 1. The user successfully receives details on the list of available flats within the specified requirements.   OR   1. No flats meet the specified requirements of the user and therefore result in an empty search result. |
| Frequency of Use: | High |
| Flow of Events: | 1. At the Search page of the website, the user will be shown a map of Singapore. On the right hand side of that map, multiple filters will be displayed. Filters include price range, geographic area, housing type and floor area.  2. The user can select the filters they want to apply on the results of their search for potential flats that they are interested in.  3. Once the user is satisfied with the filters applied, the user will press on the search button.  4. Results of the filtered search will be displayed to the user.  5. Display method will be markers of where the flats are on the map displayed.  6. Concise information of the flats will pop up when the user hovers over the markers on the map. Information includes district name, price, housing type and floor area. There will also be a favourite button to bookmark the selected flat.  7. The user can click into the markers on the map to obtain more information of the flats displayed.  8. Upon clicking on the markers, the user will be brought to another page where detailed information of the selected flat is displayed. The user will also be able to bookmark the selected flat here using the favourite button at the top of this page.  9. The user will be able to navigate back to the search page by clicking on a back button available at the bottom of the detailed information page.  10. The user will be able to reset the filters applied once back in the search page.  11. The user will also be able to look into other available flats from the result of their search. |
| Alternative Flows: | AF-S4: The user applied a filter where no available flats satisfy.  1. The message “No available flats!” will be displayed above the map display, the map will also not contain any markers that signifies available flats.  2. The user returns to Step 2 and reapply filters. |
| Exceptions: |  |
| Includes: | Favourite Houses |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. User must be able to search for houses with the filters provided

1.1. The system must provide four filters for the user

1.1.1. One of the filters must be by price in SGD

1.1.2. One of the filters must be by location

1.1.3. One of the filters must be by housing type

1.1.4. One of the filters must be by floor area in square meters

1.2. The user must be able to select any of the filters

1.3. The system must display all houses that satisfy the user’s chosen filters

1.3.1 The system must display the information in text form

1.4. The user must be able to bookmark the filtered houses

## Favourite Houses

### Description and Priority

|  |  |
| --- | --- |
| Description | Users can add flats that they have searched for into a bookmarked folder called Favourite Houses. The user will also be able to view the list of Favourite Houses bookmarked |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 004 | | |
| Use Case Name: | Favourite Houses | | |
| Created By: | Enric Tan | Last Updated By: | Enric Tan |
| Date Created: | 6 February 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Database |
| Preconditions: | 1. The user must be connected to the Internet. 2. The user has a valid account. |
| Postconditions: | 1. The system successfully saves the bookmarked flats into the Favorite Houses folder. |
| Frequency of Use: | High |
| Flow of Events: | 1. At the Favourite Houses page of the website, the user will be able to see all the flats that he has bookmarked under his account.  2. If there are currently no bookmarked flats, the page will contain an empty list.  3. The user will be able to click into the bookmarked flats to view more information on the house.  4. Upon clicking on the bookmarked flats, the user will be brought to a page where detailed information of the house is displayed.  5. The user will be able to navigate back to the Favorite Houses page by clicking on a back button available at the bottom of the detailed information page.  6. The user is able to remove flats from the bookmarked folder by clicking on the favourite button again corresponding to the selected flat.  7. To add flats to the bookmarked folder, the user will follow the steps as described in use case description 002 Searching, under steps 6 and 8. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. Logged in user must be able to view their bookmarked houses

1.1. The system must display all the bookmarked houses of the user

1.2. Logged in user must be able to remove their favourite houses

1.2. Guest users must not be able to view the favourite houses page

1.2.1. The system must notify the guest user of the restriction

## Calculator

### Description and Priority

|  |  |
| --- | --- |
| Description | User will be able to calculate how they would finance his/her target house based on their CPF, Cash savings, and disposable income |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 005 | | |
| Use Case Name: | Calculator | | |
| Created By: | Nigel Paliath | Last Updated By: | Nigel Paliath |
| Date Created: | 6 February 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Database |
| Preconditions: | 1. User has valid internet connection 2. User has a valid income range |
| Postconditions: | 1. The webpage will display information on how the user would finance the target house (eg. downpayment, mortgage) |
| Frequency of Use: | High |
| Flow of Events: | 1. User will click on the Calculator tab at the top of the webpage. 2. User will select the type of home as his/her target home. 3. User will key in his/her CPF, cash savings, and disposable income details in their respective fields. 4. User will click on the “Calculate” button. 5. Webpage will display financing information after calculations are done. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. The user must be able to check the payment plans of their BTO

1.1. The system must provide three input fields for the user

1.1.1. One of the input fields must be the user’s CPF in SGD

1.1.2. One of the input fields must be the user’s savings in SGD

1.1.3. One of the input fields must be the user's disposable income in SGD

1.1.4. The user must enter all three input fields

1.1.5. The system must notify the user if any of the inputs are unsuccessful

1.2. The system must display the buying power of the user in SGD

1.3. The system must display the down payment required in SGD

1.4. The system must display the grants available to the user

1.5. The system must display the payment plans in years and months

## Comparison

### Description and Priority

|  |  |
| --- | --- |
| Description | User will be able to compare 2 different houses based on their respective specifications |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 006 | | |
| Use Case Name: | Comparison | | |
| Created By: | Nigel Paliath | Last Updated By: | Nigel Paliath |
| Date Created: | 6 February 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Database |
| Preconditions: | 1. User has valid internet connection |
| Postconditions: | 1. The webpage will display a table of comparisons between the two target houses’ specifications |
| Frequency of Use: | High |
| Flow of Events: | 1. User will click on the Compare tab at the top of the webpage. 2. User will need to select 2 different houses as his/her target houses for comparison from a drop down list. 3. User will click on the “Compare” button. 4. Webpage will display a table comparing the 2 houses’ price, location, housing type and floor area. 5. User will be able to make a quick comparison from a glance as the more favourable statistic will be highlighted in green (eg. cheaper price) while the less favourable statistic (eg. smaller floor area) will be highlighted in red. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. User must be able to compare two houses of their own selection

1.1. The system must be able to display all the houses available

1.2. The user must be able to select two houses by name before comparing

1.3. The system must inform the user to select two houses before comparing

1.4. The system must display the two houses’ information side by side

1.5. The system must indicate the superior and inferior statistics by colour

1.5.1 The system must display the superior statistics in green

1.5.2. The system must display the inferior statistics in red

## Maps

### Description and Priority

|  |  |
| --- | --- |
| Description | Google Maps API will use the address of the users’ target house and the API would return the time taken to the nearest MRT and nearby amenities (eg. shopping malls, schools, childcare centres) on the Map |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 007 | | |
| Use Case Name: | Maps | | |
| Created By: | Nigel Paliath | Last Updated By: | Nigel Paliath |
| Date Created: | 17 February 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Google Maps |
| Preconditions: | 1. User has valid internet connection 2. Google Maps API needs to be set up |
| Postconditions: | The webpage will display the time taken to travel by foot or public transport to the nearest MRT and nearby amenities on the Map |
| Frequency of Use: | High |
| Flow of Events: | 1. After the user selects the target house, the user will click a “Show on Map” button. 2. A new tab will open showing the map with a pin on the location of the houses and other pins for the nearby amenities |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. The system must display nearby amenities and public transport to the selected house

1.1. The user must select a house

1.2. The system must use the Google Maps API on the selected house

1.3. The system must display nearby points of interest

1.3.1 The system must display nearby amenities

1.3.2. The system must display nearby public transport infrastructure

## Logout

### Description and Priority

|  |  |
| --- | --- |
| Description | Users can log out of their accounts |
| Priority | High |

### Stimulus/Response Sequences

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 008 | | |
| Use Case Name: | Logout | | |
| Created By: | Enric Tan | Last Updated By: | Enric Tan |
| Date Created: | 17 February 2024 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), SimplyStay Account System |
| Preconditions: | 1. User has an account. 2. User is already logged in to his account. |
| Postconditions: | 1. The user will not be logged in to any account on the website |
| Frequency of Use: | Low |
| Flow of Events: | 1. At the top right hand corner of every page on the website, there will be a Profile icon that users can click on. 2. Upon clicking on the Profile icon, the user’s profile will be expanded. 3. In the expanded Profile menu, there will be a “Logout” button. 4. The user will be able to click on the “Logout” button to log out of their account. 5. The website will prompt the user if he would like to confirm the log out. 6. The user will be returned to the website’s main page upon logging out, there will be no active logged in account at this point. |
| Alternative Flows: | AF-S5: The user does not confirm the log out prompt.   1. The user will be able to click on “Cancel” on the confirm logout prompt. 2. The user will be returned to the previous page that he was on upon clicking on cancel. |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

### Functional Requirements

1. The user must be able to log out of their account

1.1. The system be able to save the user’s bookmarked houses

1.2. The user must be able to log back in later

# Other Nonfunctional Requirements

## Performance Requirements

* The system must be able to return the results within 10 seconds of querying.
* The website must be fully functional within 3 seconds of opening it.
* The user must be able to successfully register an account within 2 minutes of entering the website.
* The user must be able to login within 3 seconds of inputting their login credentials

## Security Requirements

* Users must not be able to view other users’ email addresses and passwords on the website
* Users must not be able to change other users’ passwords on the website
* The system must not reveal the user’s password when the user inputs it

## Software Quality Attributes

### Maintainability

* The code must be modular and reusable where applicable
* Dependencies must be minimized to prevent coupling
* Documentation must be provided where necessary

### Scalability

* The website must support more than 20 concurrent users

### Usability

* The user interface must adhere to standard UI/UX principles

### Availability

* The downtime of the website must not exceed 12 hours per year.

### Reliability

* The system must display the correct information to the user 99.9% of the time

# 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: Data Dictionary

|  |  |
| --- | --- |
| Created by | Jared and Nigel |
| Date created | 30/1/2024 |
| API | Application Programming Interface, which is a type of software interface used to facilitate to communication between two or more computer programs |
| User Account | A means by which users can access our website, and also save previous information and data within their account. Users will have to input a unique username as unique identification, and their registered email address. |
| Guest | A means of which users can make use of the website without having to sign up for an account. Take note that by using the website as a guest, users' data inputs and information will not be saved. |
| HDBs | Housing Development Board and is colloquially used as short-hand to describe any government housing in Singapore. |
| BTO | Built-to-Order. It is one of the modes of sale of new HDB flats in Singapore. An HDB BTO sales launch happens every three months and typically offers a range of HDB flat types across a few estates. |
| Housing Type | As defined by HDB: 2-room Flexi, 3-,4-,5-room, to 3Gen flats |
| Floor area | The area designed for tenant occupancy, measured in terms of square metres |
| Area | General housing regions in Singapore which users can inquire about. They are namely: North, South, East, West, Central |
| Price Range | Users input their price ranges for their housing units. |
| Comparison | A page on the website where users will have the ability to compare two housing units with regards to different factors, such as affordability, and location. |
| Database | A spreadsheet containing all users’ accounts, and their details, such as username, password, email/phone number, along with all other saved data with regards to their housing requirements, etc |
| Amenities and Infrastructure | Refers to nearby facilities and infrastructure (MRTs, Schools, Shopping Malls, etc) |

# Appendix B: Analysis Models

Sequence diagrams, dialog maps, class diagrams, whatever