

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

School of Computer Science and Engineering

Advanced Software Engineering

System Requirement Specifications (SRS)

Website Portal: HouseHunt

Supervisor: Althea Liang

13 September 2021

By Team 5:

Tang Kai Wen, Alvin

Yong Wen Shiuan

Bankata Mishra Spriha

Gupta Suhana

Heng Chor Chen, Sabrina

Heng Fuwei Esmond

Table of contents

Table of contents	2
1. Problem Statement	5
2. Overview	5
2.1 Background	5
2.2 Overall Description	6
3. Investigation and Analysis Methodology	7
3.1 System Investigation	7
3.2 Analysis Methodology	7
3.2.1 Feasibility study and requirements elicitation	7
3.2.2 System analysis and requirements specification	8
3.2.2.1 Perform an analysis of the problem using UML	8
3.2.2.2 Scope and Limitations	8
Business Analysis	8
Business Rules	8
Business System Interface	9
Associated Budget Requirements	9
Requirement Analysis	10
User Requirements Definition	10
Security Requirements	10
Data Analysis	11
Data Collection Process	11
Data Validation	11
Data Storage	11
Data Manipulation & Retrieval	11
Process Analysis	12
Application Architecture	16
Usability, User Interface Design & Interaction	16
Application Implementation	16
Information Structure	17
3.2.3 Object-Oriented Design using UML	18
3.2.3.1 Use Case Model	19
3.2.4 Prototyping	19
4. Constraints	20
4.1 Scalability and Costs	20

4.2 Data and Function Mapping	20
4.3 Proprietary Hardware and Software	20
4.4 Batch Updates vs (close) Real-Time Updates	20
4.5 Project Schedule	20
5. Operational Requirements	21
5.1 Help Desk Support	21
5.2 Application Services and Technical support	21
5.3 Administration Features	21
5.4 System hardware fail over and routine back up	21
5.5 Audit Trail	22
6. Functional Requirements	23
6.1 Searching for a flat	23
6.2 Estimate Selling Price	24
6.3 View on Map	25
6.4 View Data Visualizations	25
6.5 Calculate Maximum Affordability	26
7. Input Requirements	27
7.1 User to Select Input Fields to Filter HDB Resale Flat Transactions	27
7.2 User to Click on Google Maps API	27
7.3 User to Calculate Affordability of Flat	27
7.4 User wants to View Data Visualization	27
8. Process Requirements	28
8.1 Compatibility	28
8.2 data privacy	28
8.3 data validation	28
8.4 Performance	28
8.5 Security	28
9. Output Requirements	29
9.1 after user inputs the necessary details to search a flat	29
9.2 users wants to view the flat location on map	29
9.3 user wants to calculate affordability of flat	29
9.4 user wants to view data visualisation	29
10. Hardware Requirements	30
10.1 Network	30
10.2 Client Computers & Smart Devices	30

11. Software Requirements	30
11.1 Client Operating Systems	30
11.2 Client Application	30
11.3 Network system	31
12. Deployment Requirements	32
Appendix A	33

1 Problem Statement

With the advent of the COVID-19 pandemic, the Singaporean real estate market has undergone some major changes. Its current state is extremely volatile and HDBs are susceptible to price fluctuations. The current market is confusing to first-time buyers and sellers due to the vast amount of data available. Moreover, it is difficult to stay up to date with the real estate market due to its erratic nature and the effects of external unforeseen circumstances such as multiple lockdowns to control the proliferation of COVID-19.

Hence, we propose a web based flat resale tool, 'HouseHunt,' that estimates the value of real estate sellers' flats based on factors such as flat type and model, locality, remaining lease, and floor area. HouseHunt also caters to potential buyers searching for their own HDB accommodation.

2 Overview

2.1 Background

The Singapore housing market is ever-changing. Buyers and sellers need a one-stop destination to solve the problems they face when entering the housing market. Singapore's real estate prices are highly dependent on a multitude of factors, which can be overwhelming to novice users. Moreover, the recent pandemic has caused a wave of cancellation of Build-to-Order. As a result, even experienced users may face confusion with regards to the impact it has on the resale market.

Young Singapore Adults factor several key factors when they have to purchase their first HDB. The most deciding factor for them is the location. More specifically, how close their HDB is with respect to public transport, schools, and shopping centers. Furthermore, as previously mentioned, the recent pandemic had made many young home buyers recalibrate their choices as prices of such flats had ballooned beyond their budgets. Weighing all these factors could pose an issue when all the data are presented by their property agent as they may be bombarded by information, they do not know what to compare with. This could lead to decision fatigue and ending up purchasing a HDB flat that they are not satisfied with.

Given the volatile state of the Singaporean real estate market, there is a need for a comprehensive and cogent solution that collates currently available real estate information and presents it based on users' priorities. HouseHunt aims to be the one-stop solution to the average Singaporean's real estate needs by focusing on ease of access and reliability.

2.2 Overall Description

In essence the HouseHunt system provides the interface to help find the Ideal Resale HDB system. As the recent pandemic had made many young home buyers recalibrate their choices as prices of such flats had ballooned beyond their budgets.

With HouseHunt, it aims to be the one-stop solution to the average Singaporean's real estate needs.

HouseHunt is a website application built with Django framework, designed to assist Singaporean users on resale flat related information. Information provided includes flat finding based on certain criteria, estimations of resale flats prices based on recent data, house affordability calculations using user's income information, and data visualisations of house prices against certain parameters such that users can understand the pricing trends in Singapore

3 Investigation & Analysis Methodology

3.1 System Investigation

The HouseHunt system provides information including flat finding based on certain criteria. Users can either search for a resale flat, calculate their house affordability, estimate the flat price of a resale flat, or view data visualisations of resale flat prices against towns, flat types, or year of sale.

HouseHunt estimates the resale flats prices based on recent data, allowing users to estimate the selling price of the flat. The system allows the user to input the location, flat type, flat model, remaining lease of the flat and floor area of the flat which are mandatory. Upon submission, the system will then return a selling price estimate of the flat based on these inputs. If the system is unable to provide an accurate estimate, the system must notify the user. The house affordability calculation is calculated by the user's income information as the system will allow the user to input their down payment, monthly income, debts, and interest rate of the loan they are planning to take. Upon submission, the system will then return the house affordability of the user.

Furthermore, HouseHunt provides the users the ability to search for a flat with the returned house affordability as their maximum price. Data visualisations of house prices against certain parameters such as house prices against town, flat types, or year of sale will then help users better understand the pricing trends in Singapore and provide them with a graph depicting their choice.

3.2 Analysis Methodology

3.2.1 Feasibility study and requirements elicitation

Organize a development and implementation team composed of people knowledgeable about the current resale prices of housing information with which regular meetings will be held. A series of interviews with the managers and the developers of the current HouseHunt application will be arranged. Interview and feedback from the personnel and staff working directly with the HouseHunt system is needed to define the current environment and future system requirements. A Feasibility and Risk Assessment study will be conducted to determine which solution(s) are most appropriate based upon the results of the interviews.

3.2.2 System analysis and requirements specification

3.2.2.1 Perform an analysis of the problem using object-oriented techniques

An external view of the enterprise model of the information provided including location, flat type, flat model, remaining lease of the flat and floor area of the flat will be developed using Unified Modeling Language (UML). This System Requirement Specifications documents will form part of the documentation for the project. Some desired features of the new system include:

- The ability to search flat based on criteria
- The ability to search/view estimation of flat prices
- Provide House affordability calculation
- Data visualisation to better understand the pricing trends

3.2.2.2 Scope and Limitations

Analysis methodology will involve business analysis, requirement analysis, data analysis, process analysis, (web) and application architecture:

Business Analysis

HouseHunt is a non-profit web application that runs on modern browsers that support JavaScript. Hence it can be used on mobile devices running iOS or Android operating systems, as well as computers.

Business Rules

The following rules shall be taken into consideration by the stakeholders, developers, and owners of HouseHunt application when decisions need to be made:

1. HouseHunt shall remain free for all users.
2. HouseHunt shall not display advertisements
3. HouseHunt shall not store user input data.

Business System Interface

Collaborative business includes the interactions between the different departments within an organization, hence, interfaces are a crucial element in a collaborative business process. Thus, Business Interoperability Interface (BII) is implemented. It enables business interoperability among the various departments, in other words, exchange of information and making the best use of it. Exchange of information among the different departments will be done through the MediaWiki site, emails, text messages and Google Drive for essential document creations, updates, and revisions.

Associated Budget Requirements

Item	Supplier	Quantity	Unit Price	Total
Front-End Developer	-	1	\$5,000	\$5,000
Back-End Developer	-	2	\$5,000	\$10,000
Quality Assurance Manager	-	1	\$4,000	\$4,000
Project Manager	-	1	\$4,000	\$4,000
Release Manager	-	1	\$4,000	\$4,000
Computers	DELL	6	\$1,000	\$6,000
Office Rental	NTU	1	\$6,000	\$6,000
Router	TP-Link	1	\$35	\$35
Main Power Supply	PUB	1	\$500	\$500
			Total	\$39,535

Table 1: Associated Budget Requirements

Requirement Analysis

The requirement analysis outlines user requirements definition, as well as the security requirements of HouseHunt. The system I/O descriptions will be explained in the Input and Output Requirements section, while functional requirements will be explained under the Functional Requirements section.

User Requirements Definition

The system user's requirements definition dictates what the system has to provide the user to fulfil the relevant use cases provided by the use case model.

1. The user interface must be user friendly and not too confusing.
2. The user must be able to input flat type, remaining lease years, maximum resale price, town, minimum floor area, and flat model to search for a list of flats.
3. The user must be able to view the flat location on Google Maps after pressing the Enter button.
4. The user must be able to request the system for a visualisation of resale flat prices.
5. The user must be able to input their monthly fixed income, monthly debt, monthly payments/loans interest rate, and cash towards down payment to calculate maximum affordable price

Security Requirements

System Security is paramount in any software development project, this prevents the system from being hacked or compromised by third parties. The following lists the security measures HouseHunt aims to uphold:

- User input values for monthly income, monthly debt, and down payment must not be stored, in case of hacking incidents where such data may be leaked and made public.
- Access to the deployed website must involve forcing HTTPS, whereby anyone who accesses the site using the insecure http URL will be redirected to the secure HTTPS one.

Data Analysis

Data Collection Process

Resale flat transaction data used in HouseHunt was obtained from gov.sg. The URL is as follows: <https://data.gov.sg/dataset/resale-flat-prices>.

Data Validation

In the HouseHunt system, user input validation is done via the following mechanisms:

1. Dropdown boxes: For input fields with limited options, e.g., Town and flat type, there are dropdown menus to restrict user inputs
2. Compulsory fields
3. Restricted numerical range of inputs, e.g., no negative numbers for certain fields

Data Storage

The resale flat data used in the HouseHunt app will be stored in a SQLite3 database.

Data Manipulation & Retrieval

Of the Create, Retrieve, Update, Delete functions, only Retrieval of data can currently be done through the website's user interface.

Process Analysis

The Dialog Map below indicates the different possible states and transitions if one is in the process of using the HouseHunt website:

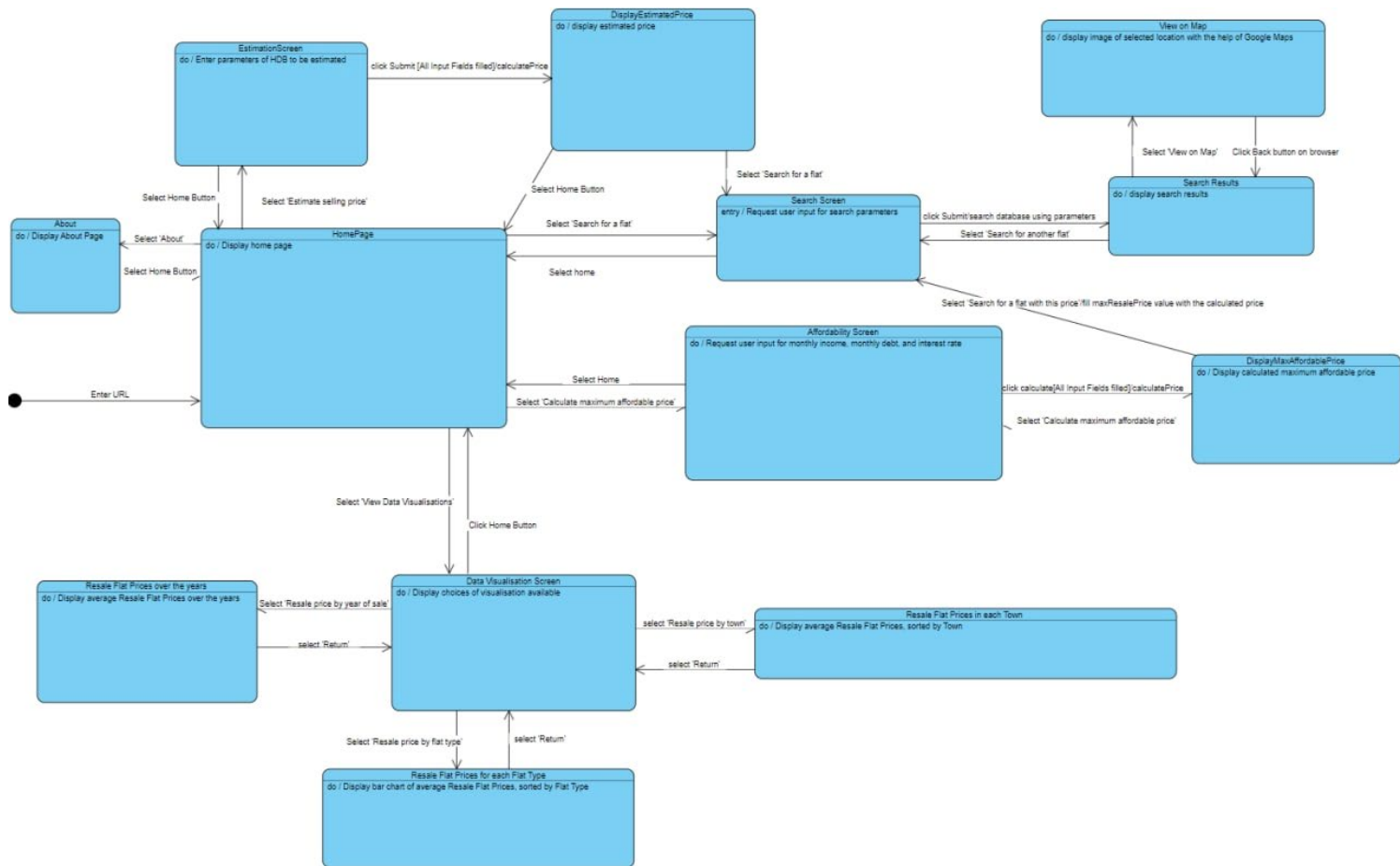


Figure 1: Dialog Map of typical HouseHunt Usage

Moreover, attached below are the Sequence Diagrams for each of the use cases for HouseHunt.

1. Searching For a Flat

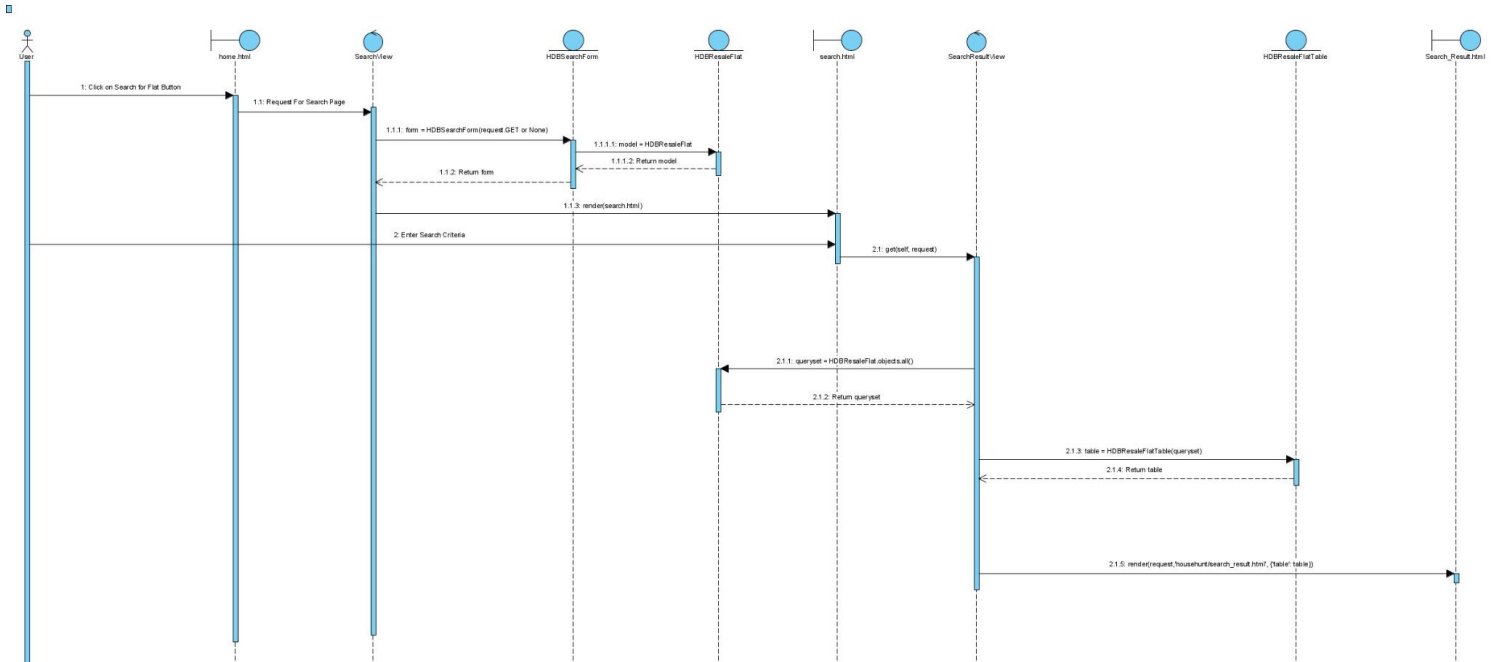


Figure 2: Sequence diagram of a typical Searching For a Flat Process

2. Estimate Selling Price

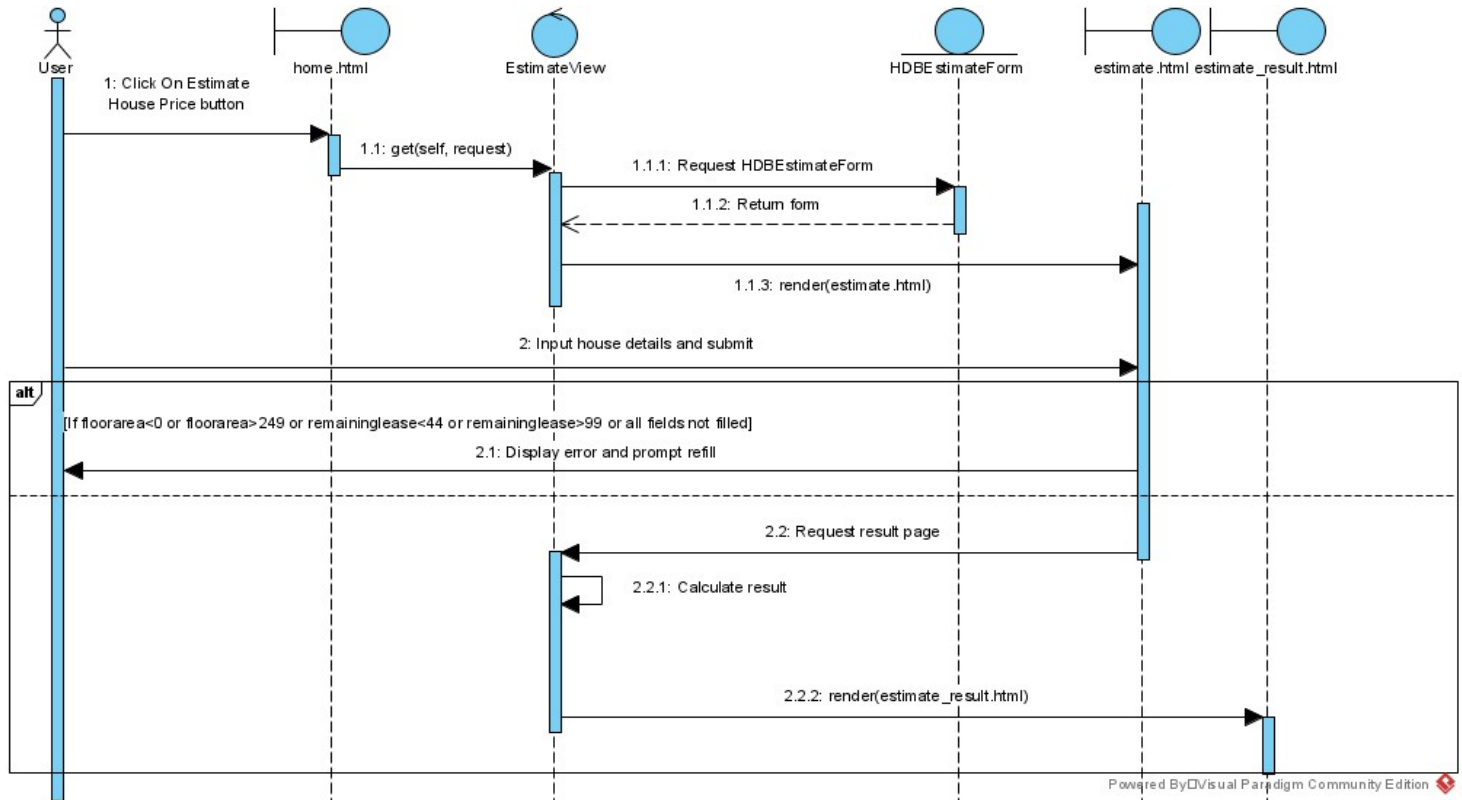


Figure 3: Sequence diagram of a typical Estimate Selling Price Process

3. Calculate Maximum Affordable Price

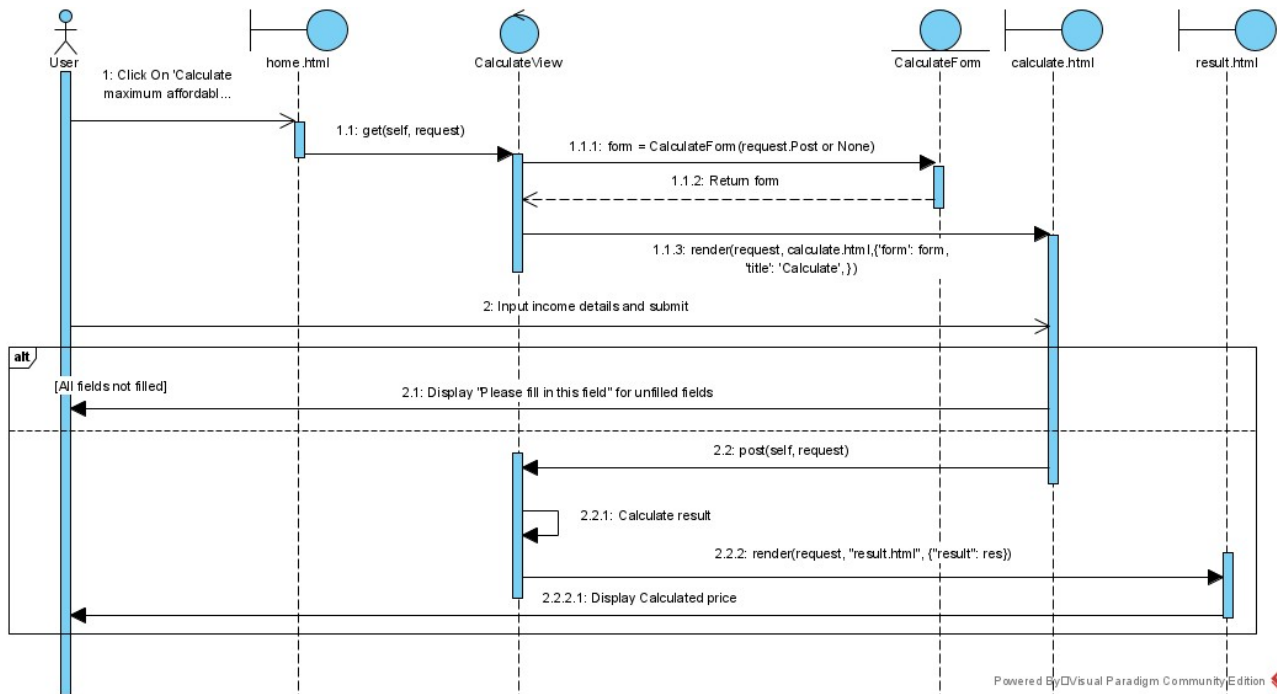
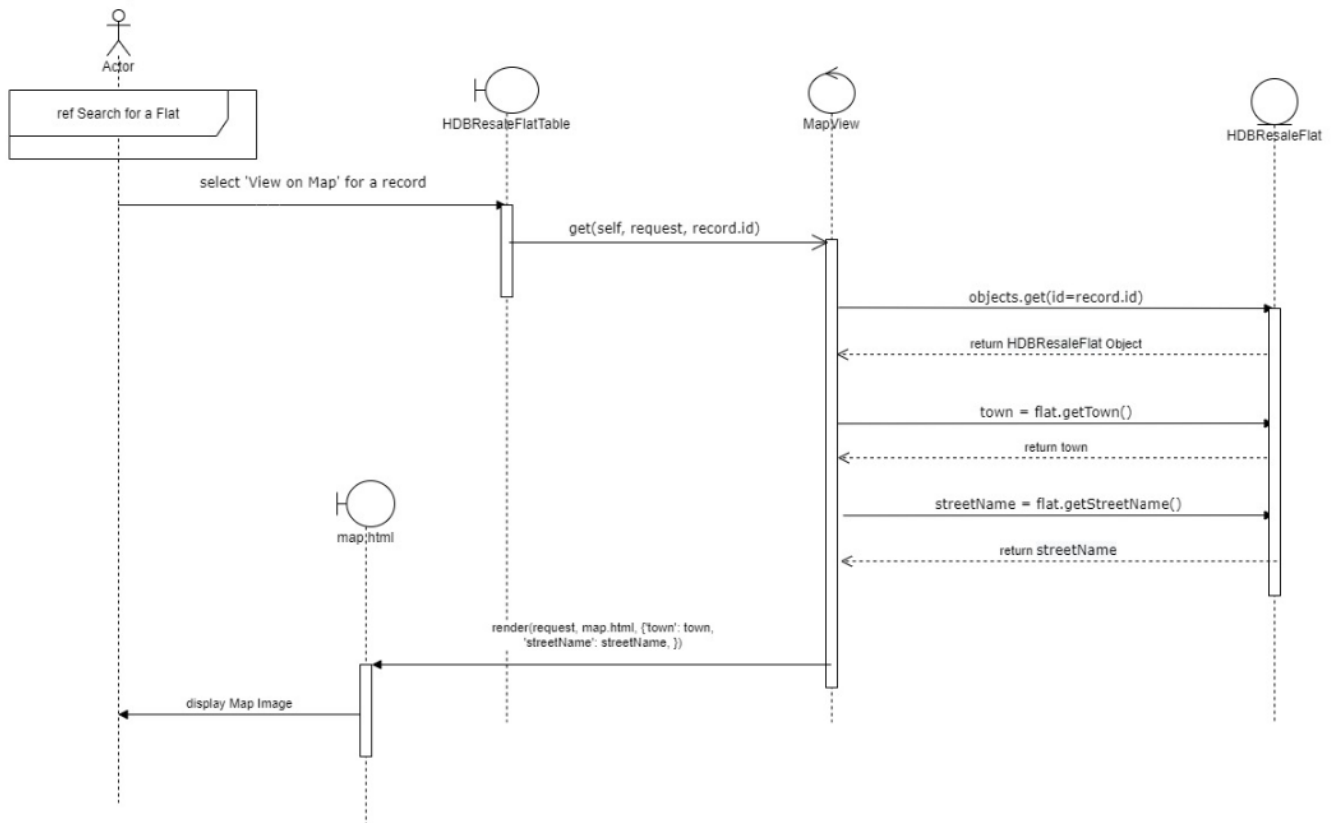


Figure 4: Sequence diagram of a typical Calculation of Maximum Affordable Price Process

4. View On Map

Figure 5: Sequence diagram of a typical Viewing on Map Process



5. View Data Visualizations

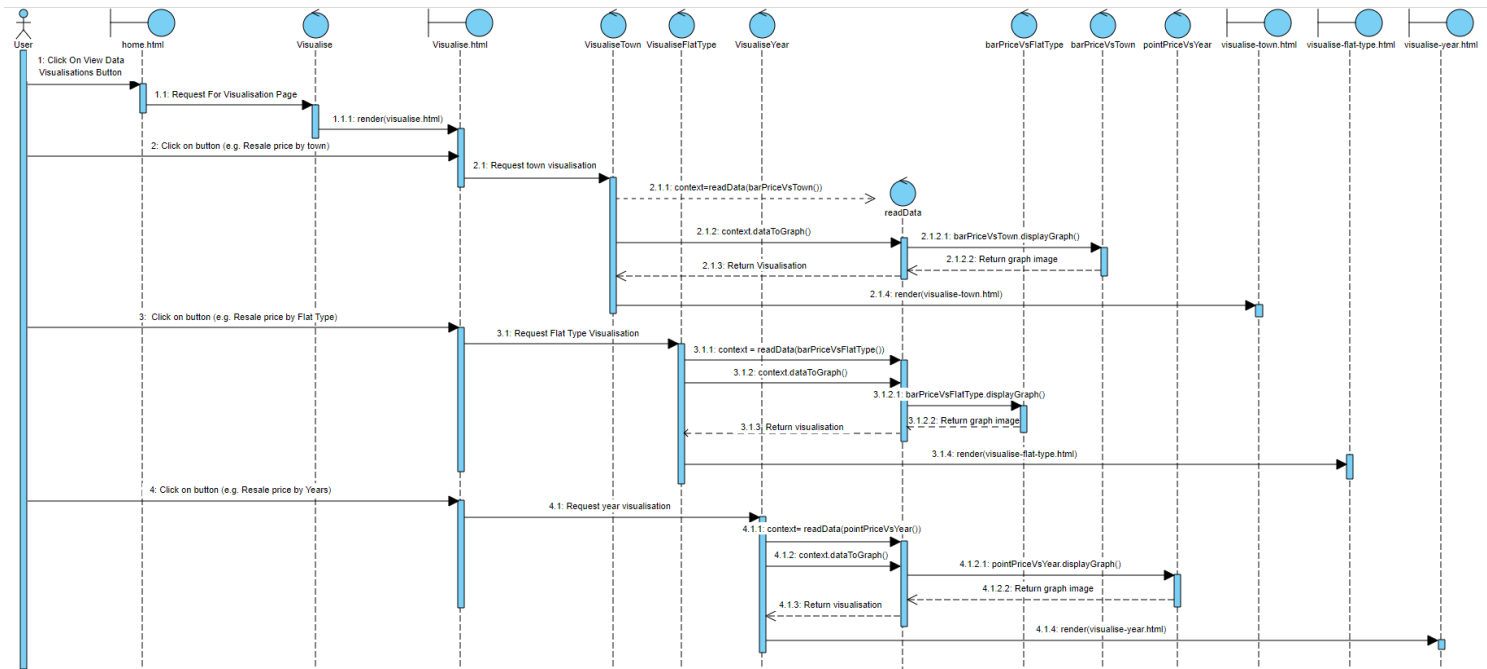


Figure 6: Sequence diagram of a typical Viewing of Data Visualizations Process

Application Architecture

Usability, User Interface Design & Interaction

During the specification analysis phase, the team will ensure usability and maintain transparency on the desired deliverable by coming up with a User Interface (UI) design (included in Appendix A). This will serve as a good reference documentation as well as the initial phase of the proof of concept for the team and stakeholders to validate the product's scope and limitations following the ISO 9241-220:2019 standard which addresses ergonomics of Human-System Interaction (Also referred as Human-Computer Interaction).

The UI design that is hand drawn is provided to the stakeholders with steps to access it in Appendix A. The UI design provides detailed comments that explains in a quick and easy way to translate high-level design concepts into tangible and testable artifacts. For example, from one single UI design file, not only will the front-end team be clear about their deliverables but the backend team will also understand what data will need to flow through this page. Aside from the development team, other stakeholders like the Project Manager, Quality Assurance team, or users will also know what to expect from the product.

Once the UI design is cross-checked and agreed upon by the team, the development team will subsequently enter the design of the software architecture for the prototype according to the accepted design and constraints.

Application Implementation

With the software blueprints on hand, the development team will henceforth proceed onto the next step on the roadmap which is to rapidly prototype the most viable prototype as per requirements using the Bootstrap framework and Django.

The frontend involves HTML and CSS. It uses the Bootstrap framework as well for items such as buttons.

The backend server uses the Django framework for web development. As Django is a 'batteries-included' framework, the development team can spend less time reinventing the wheel and more time building the core features of the web portal. Django is ideal for development of large-scale, database-driven, interactive web applications. It is also highly customizable and scalable with extensive community and documentation to allow the development team to efficiently match any

new changes requested by the stakeholders without any delays in the makings of the HouseHunt web portal. The backend also interfaces with the SQLite3 database using Django's in-built Object-Relational-Mapper (ORM).

Therefore, using these frameworks to develop the HouseHunt application, the development team is confident to meet the scope of the requirements defined in the proposal. However, there are still certain constraints with these technologies which will be clarified in the Constraint section of this report.

Information Structure

The team will be using the SQLite3 database to store the HDB resale flat data. The interfacing between the database and the backend is achieved through Django's Object-Relational Mapper (ORM). The development team will draft a typical UML Diagram to fully illustrate the associations between schemas.

Following the journal, the team will first draft a Entity Class Diagram and then further finalize the details under the Full Class Diagram. The class diagram is used to describe the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. The figure 7 below hence illustrates the Entity Class Diagram followed by figure 8 which illustrates the Full Class Diagram for HouseHunt.

Entity Class Diagram

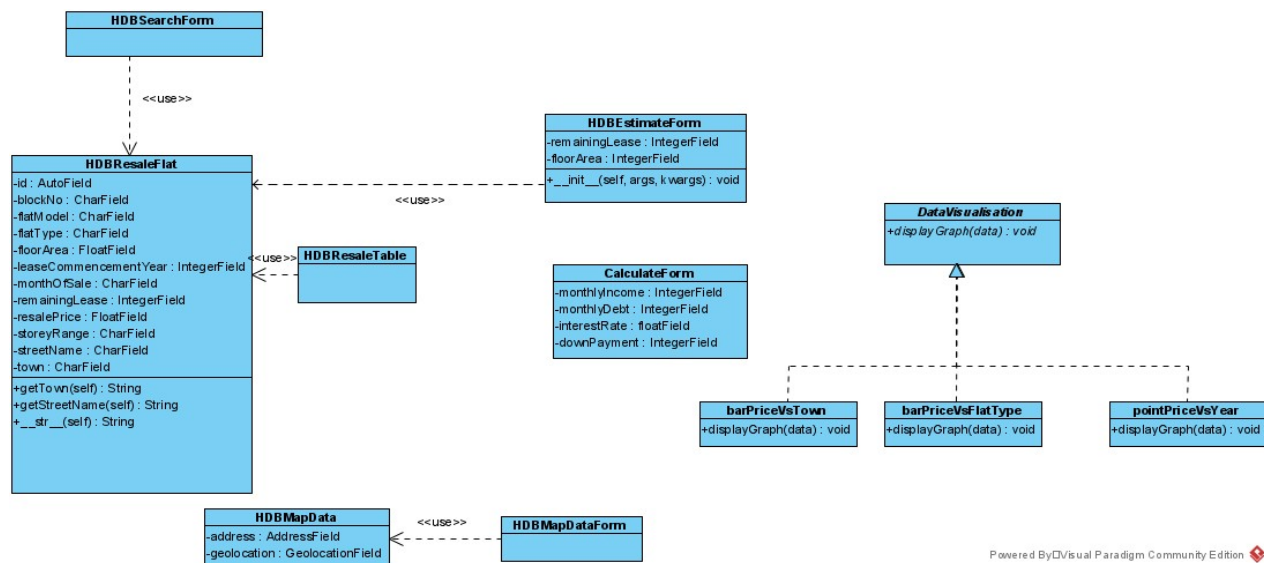


Figure 7: HouseHunt's Entity Class Diagram

Full Class Diagram

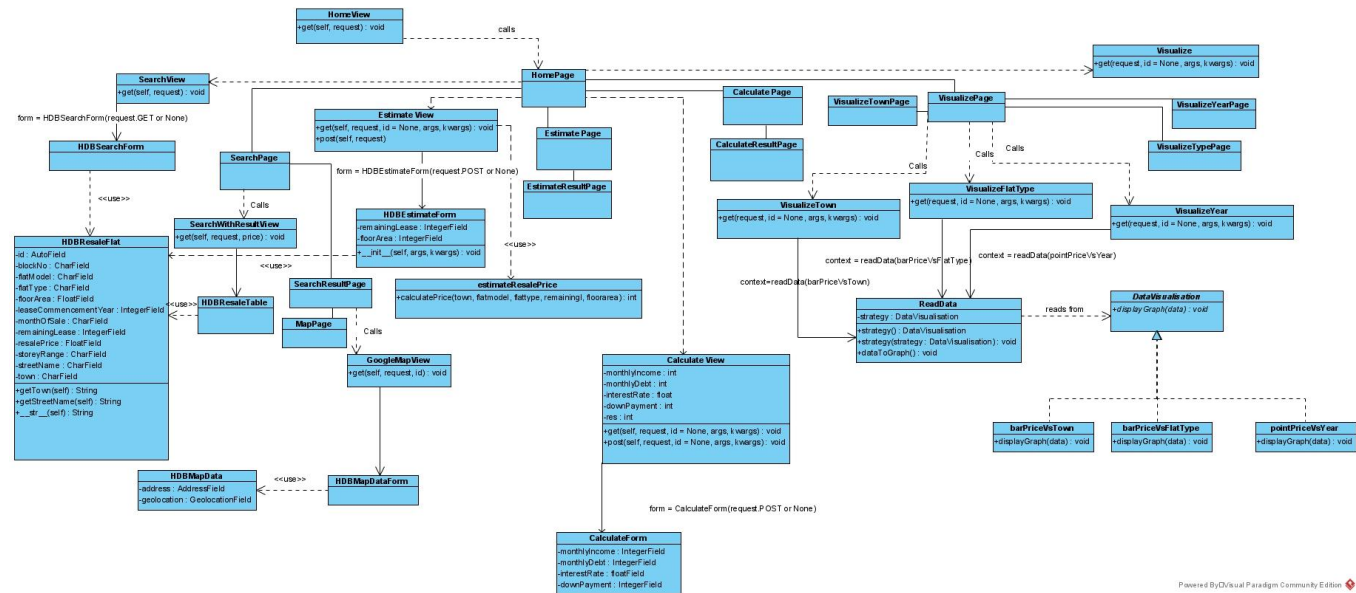
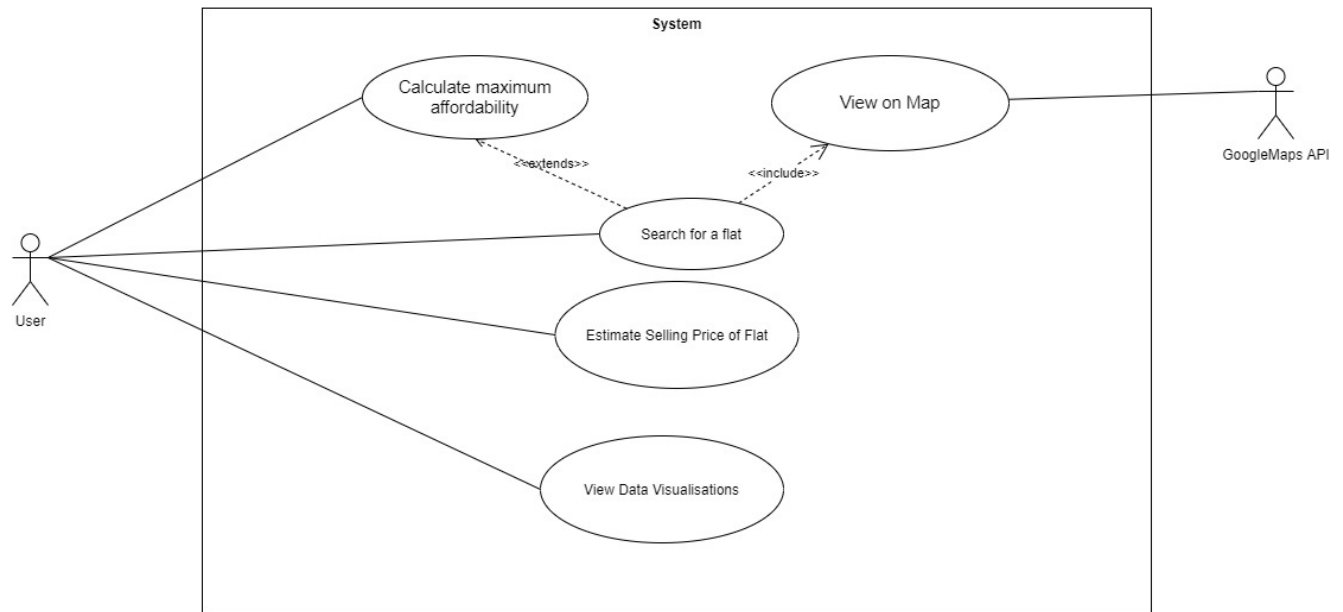


Figure 8: HouseHunt's Full Class Diagram

3.2.3 Object-Oriented Design using UML

A detailed object-oriented design for the registration system will be developed. UML will be used again for the graphical representation and documentation of the design. The system will primarily concern itself with the registration process. At its core, a user will fill out or answer a web-based form that will be processed in near real time by the host back- end system. In addition, the system will allow users to check housing areas that are suitable for their needs and provide feedback regarding current HouseHunt applications.

3.2.3.1 Use Case Model



3.2.4 Prototyping

The Object-Oriented Rapid Prototyping (OORP) method will be used to implement a limited and functional prototype for the registration system. The prototype will be a working example of part of the system for demonstration and proof of concept purposes only. It will include web-based forms as an end-user interface with the database. The prototype will be presented to the implementation team.

4 Constraints

4.1 Scalability and Costs

The backend server uses the Django framework for web development (specifically, version 3.1.7). It interfaces with the SQLite3 database using Django's in-built Object-Relational-Mapper (ORM). Django is an extremely scalable framework. It allows for horizontal scaling to support hundreds of millions of requests. Generally, as the user base expands, websites start slowing down. Django apps can be scaled reliably as web traffic increases.

Meanwhile, SQLite tends to work well with websites that generate low to medium traffic. As HouseHunt relies mostly on read access rather than write access, thus limiting its interactions with the database, there should be no issues with scalability. SQLite has been documented to work well with websites that generate up to 100K hits/day.

4.2 Data and Function Mapping

A new function added to the system cannot be readily mapped to the HouseHunt website. For example, adding a new flat to the SQLite3 database will require changes in source code and recompilation of the website.

4.3 Proprietary hardware and software

The Web Application is designed for devices that support web browsers with Javascript capabilities, including computers, phones and tablets. The application also requires an Internet connection.

4.4 Batch updates vs. (close) Real-time updates

SQLite3 does not offer real time updates. Instead, batch updates are used.

4.5 Project Schedule

There is a seven-week timeframe to implement a fully functional HouseHunt web application in time for a final presentation to the clients.

5 Operational Requirements

5.1 Help Desk Support

System users have a 24x7 access to telephone assistance for questions that are technical in nature, such as, slow, or sluggish system response time, incompatible browser features, application errors, system downtime inquiries, account lock-out assistance, etc.

5.2 Application Services and Technical support

Programmers and application developers will have access to source code to address bugs or system enhancements as deemed necessary. Network Administrator and DBA support is also required to maintain a 24x7 system uptime.

A help desk support will be made available to the users, in situations where they encounter technical issues such as system downtime. This support will be open from 8am to 5pm on the working days to aid the users.

5.3 Administration Features

System security and access levels are provided in the online system. There are varying levels of system access and functional authority. Each student's access is limited to his/her own registration records. Only authorized system administrator(s) have access to all student registration records.

Backup will be scheduled by the development team on a fixed date of every month unless new features are being implemented into the system or a bug fix is required.

5.4 System Interface independent of VRU

The VRU system will remain operational, and its functionality will be complementary but independent from the online registration system. At any one time, students may use either the VRU system or the online system only, but not both. The online system will be operational even if the VRU system is offline and vice-versa.

5.5 Audit Trail

PythonAnywhere stores an access log, error log and server log for the website. Each entry in the access log indicates the time and IP address of access. These can be used in auditing for verification.

6 Functional Requirements

The HouseHunt website attempts to be a 'one-stop' portal for potential buyers and sellers of resale HDB flats.

6.1 Searching for a flat

6.1.1 The system must allow the user to input flat type, remaining lease years, maximum resale price, town, minimum floor area, and flat model to search for a list of flats.

6.1.1.1 The system must allow the user to select a flat type from a dropdown box with options: 1-Room, 2-Room, 3-Room, 4-Room, 5-Room, Executive and Multi-generation.

6.1.1.1.1 If left blank, the system must put a default value of "Any" for the flat type.

6.1.1.2 The system must allow the user to key in the remaining lease years of the flat.

6.1.1.2.1 The system must allow the user to key in only numerical values.

6.1.1.3 The system must allow the user to key in the maximum resale price of the flat.

6.1.1.3.1 The system must allow the user to key in only numerical values.

6.1.1.4 The system must allow the user to select the desired town from a dropdown box of different towns in Singapore. (For e.g. Any, Ang Mo Kio, Bishan, etc.)

6.1.1.4.1 The system must put a default value of 'Any' for the town selection.

6.1.1.5 The system must allow the user to select a minimum floor area.

6.1.1.5.1 The system must allow the user to key in only numerical values.

6.1.1.6 The system must allow the user to select a flat model from a dropdown box with options: '2-room', 'Adjoined flat', 'Apartment', 'DBSS', 'Improved', 'Improved-Maisonette', 'Maisonette', 'Model A', 'Model A-Maisonette', 'Model A2', 'Multi Generation', 'New Generation', 'Premium Apartment', 'Premium Apartment Loft', 'Premium Maisonette', 'Simplified', 'Standard', 'Terrace', 'Type S1', 'Type S2'.

6.1.1.6.1 If left blank, the system must put a default value of "Any" for the flat model.

6.1.2 The system must save the data mentioned in 6.1.1 to the database upon successful user request.

6.1.3 Upon successful user request, the system must retrieve a list of flats that matches the user's inputs from the data.gov.sg API.

6.1.3.1 The system must save the list of flats to a database.

6.1.3.2 The system must query the Google Maps API for each flat retrieved in 6.1.3 for a street view using the Street attribute of the flat.

6.1.3.2.1 The system must save the street view of the flat to the database.

6.1.4 Upon entering valid input parameters, the system must display a list of flats that matches the user's input.

6.1.4.1 The system must display the list of flats starting from the earliest resale date transaction to the latest resale date transaction

6.1.4.1.1 The system must allow the user to view more information about each flat in the list.

6.1.4.1.1.1 Information about a specific flat must include the price of the flat.

6.1.4.1.1.2 Information about a specific flat must include the street of the flat.

6.1.4.1.1.3 Information about a specific flat must include the block number of the flat.

6.1.4.1.1.4 Information about a specific flat must include the town of the flat.

6.1.5.1.2 The system must be able to display the GPS location of the flat.

6.2 Estimate Selling Price

6.2.1 The system must allow the user to input flat type, town, flat model, remaining lease in years and floor area to search for a list of flats.

6.2.1.1 The system must allow the user to select a flat type from a dropdown box with options: 1-Room, 2-Room, 3-Room, 4-Room, 5-Room, Executive and Multi-generation.

6.2.1.2 The system must allow the user to select the desired town from a dropdown box of different districts in Singapore. (For e.g. Ang Mo Kio, Bishan, etc.)

6.2.1.3 The system must allow the user to select a flat model from a dropdown box with options: '2-room', 'Adjoined flat', 'Apartment', 'DBSS', 'Improved', 'Improved-Maisonette', 'Maisonette', 'Model A', 'Model A-Maisonette', 'Model A2', 'Multi Generation', 'New Generation', 'Premium Apartment', 'Premium Apartment Loft', 'Premium Maisonette', 'Simplified', 'Standard', 'Terrace', 'Type S1', 'Type S2'.

6.2.1.4 The system must allow the user to key in the remaining lease years of the flat.

6.2.1.4.1 The system must allow the user to key in only numerical values.

6.2.1.4.2 The system must allow the user to key in only values within the range 77~99 inclusive.

6.2.1.5 The system must allow the user to key in the floor area of the flat.

6.2.1.5.1 The system must allow the user to key in only numerical values.

6.2.1.5.2 The system must allow the user to key in only values within the range of 0~279 inclusive.

6.2.2 Upon user request for an estimate, the system must validate that all fields listed in 6.2.1 are filled.

6.2.2.1 The system must display an error message if either 6.2.2, 6.2.1.4.2 or 6.2.1.5.2 is not fulfilled.

6.2.3 Upon successful request, the system must return a price estimate of the flat.

6.3 View on Map

6.3.1 The system must allow the user to retrieve the address of the flat from 6.1.3.

6.3.2 The system must display the retrieved address in 6.3.1 for the user to confirm whether it is the correct address.

6.3.3. The system must allow the user to view the location on Google Maps after pressing the Enter button.

6.3.3.1 The system must allow the user to view the location on a dynamic map view, satellite view and street view.

6.3.3.2 The system must allow the user to toggle around the map to view nearby amenities and the surroundings.

6.3.3.3. The system must return the geolocation coordinates.

6.4 View Data Visualizations

6.4.1 The user must be able to request the system for a visualisation of resale flat prices.

6.4.1.1 Upon initial user request for visualisation of resale flat prices, the system must first display the list of graphs that the user can select to view.

6.4.1.2 Upon user request, the system must display a bar chart showing the average resale prices of flats categorised by town.

6.4.1.3 Upon user request, the system must display a bar chart showing the average resale prices of flats categorised by flat type.

6.4.1.4 Upon user request, the system must display a line graph showing the average yearly

resale prices of flats over time.

6.4.1.5 The system must be able to switch between the three visualisations in 6.4.1.2, 6.4.1.3 and 6.4.1.7 upon user request.

6.5 Calculate Maximum Affordability

6.5.1 The system must allow the user to input their monthly fixed income, monthly debt, monthly payments/loans interest rate, and cash towards down payment.

6.5.1.1 The system must allow the user to key in their monthly fixed income.

6.5.1.1.1 The system must allow the user to key in only numerical values.

6.5.1.2 The system must allow the user to key in their monthly debt.

6.5.1.2.1 The system must allow the user to key in only numerical values.

6.5.1.3 The system must allow the user to key in their loan interest rate.

6.5.1.3.1 The system must allow the user to key in only numerical values.

6.5.1.4 The system must allow the user to key in their cash towards down payment.

6.5.1.4.1 The system must allow the user to key in only numerical values.

6.5.2 Upon user request for a financing guide, the system must validate that all fields listed in 6.5.1 are filled.

6.5.2.1 The system must display an error message if 6.5.2 is not fulfilled.

6.5.3 Upon successful request, the system must return an estimate of the maximum amount the user should spend on a flat

7 Input Requirements

7.1 User to select input fields to filter HDB resale flat transactions

Each user has the option to input fields for town, flat type, flat model, maximum price, floor area, and remaining lease years. Of the fields, town, flat type, flat models have dropdown menus, limiting user input options for those fields. User fills in the desired filters and submits it. System queries the data.gov.sg resale flat price dataset using the filter to get a filtered dataset.

7.2 User to click on Google Maps API

User clicks on Google Maps button or View Map button. The system requests the Google Maps API for either a street view or a dynamic map view.

7.3 User to calculate affordability of flat

User to input fields for monthly income, debts, down payment, and loan interest rate to estimate the maximum amount the user should spend on a flat.

7.4 User wants to view data visualisation

User clicks on the View Data Visualisations button and can choose between viewing data visualisations of flat price against town, flat type or year or sale.

8 Process Requirements

The following are among the inherent requirements that the HouseHunt system must be able to handle.

8.1 Compatibility

The website must be accessible on different kinds of devices. Specifically, the website must be accessible on phones running Android 6.0 and above, as well as phones running iOS 13.0 and above. Moreover, the application must be compatible with computers running Windows 7 and above as well as computers running Mac 11 and above.

8.2 Data privacy

User input values for monthly income, monthly debt, and down payment must not be stored, in case of hacking incidents where such data may be leaked and possibly be sold for a profit.

8.3 Data validation

There will be data validation routines as part of the HouseHunt website, in case of input errors from the user's end.

8.4 Performance

For each of the use cases (i.e., estimating sale price, calculating affordability, searching for a flat, viewing data visualisation, etc.), the system must return a response in 20 seconds or less.

8.5 Security

Access to the deployed website must involve forcing HTTPS, whereby anyone who accesses the site using the insecure http URL will be redirected to the secure HTTPS one.

9 Output Requirements

9.1 After user inputs the necessary details to search a flat

System queries the data.gov.sg resale flat price dataset using the filter to get a filtered dataset, which is then displayed to the user. This includes the available flats in Singapore that are filtered according to the town area, flat type, flat model, maximum price, floor area remaining lease years and their available selling price.

9.2 User wants to view the flat location on map

The system requests the Google Maps API for either a street view or a dynamic map view and retrieves the image. User is then redirected to a page with the map view displayed.

9.3 User wants to calculate affordability of flat

System calculates the house affordability and displays it with relevant information of users. This includes monthly income, debts, down payment, and loan interest rate.

9.4 User wants to view data visualisation

Once user chooses a visualisation, system will display a graph corresponding to its description.

10 Hardware Requirements

10.1 Network

Internet access through any network

10.2 Client Computers

- Mac, Unix, and Windows client computers
- Phones running Android 9.0 or above / iOS 11 or above

10.3 Production support systems

The website is deployed on PythonAnywhere, which writes disk data to two separate disks, one primary one and one mirror.

11 Software Requirements

11.1 Client Operating Systems

- UNIX (any flavor)
- MAC
- Windows
- iOS
- Android

11.2 Client Application

Any JavaScript compatible browsers such as:

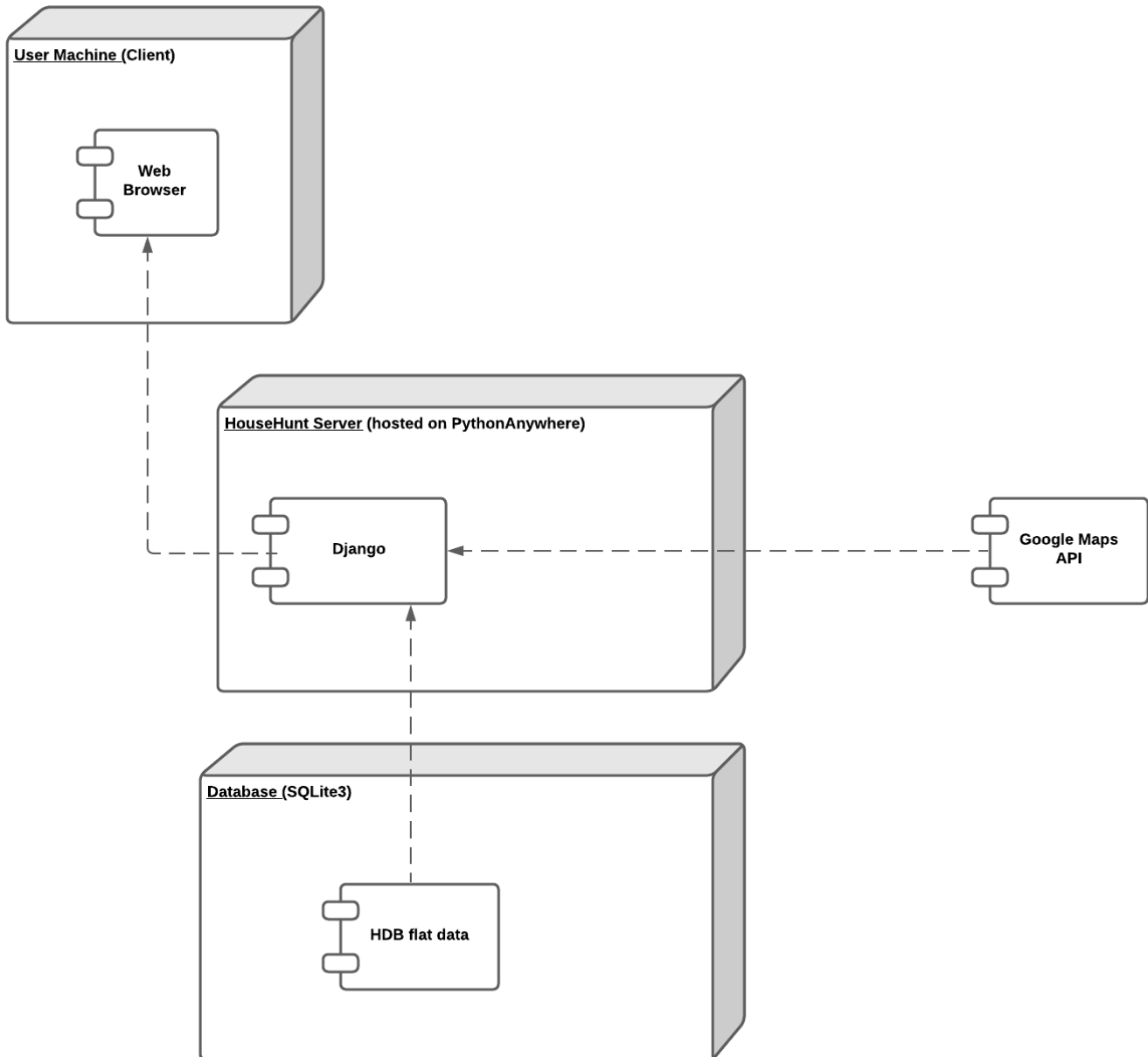
- Chrome
- Firefox
- Safari

11.3 Network system

Network software and protocols for systems to communicate:

- TCP/IP
- HTTPS

12 Deployment Requirements



The website is built using the Django framework. Google Maps APIs such as Geocoding API and Street View Static API may be called by the website. The resale flat data provided by data.gov.sg is stored in the SQLite3 database, which interfaces with the backend server using Django's in-built Object-Relational-Mapper.

Appendix A

UI Mockup File Name

Refer to the file named “HouseHunt Mockup UI.pdf” for clearer view of the Mockup of HouseHunt

UI Mockup Sample:

Recommend Me A Flat Page

HOUSEHUNT About Contact Us

RECOMMEND ME A FLAT

Monthly Fixed Household Income:

Credit Card Minimum Payment:

Car Loan Payment:

Housing Grants:

Family Size:

SUBMIT

Allow users to fill in their **personal** information so that the system can recommend to users a lot of flats

- Allow users to key in the input - mandatory field to fill in.

Please ensure you fill in monthly household income!

An error pop up if the monthly fixed household income text field is not filled.

Clicking on "submit" will bring user to Recommendation page where our system suggests users the various flats that is best suitable for them

Figure A.1: UI MockUp of HouseHunt's interface for Recommend Me A Flat page