
Use Cases

for

PropertyIQ

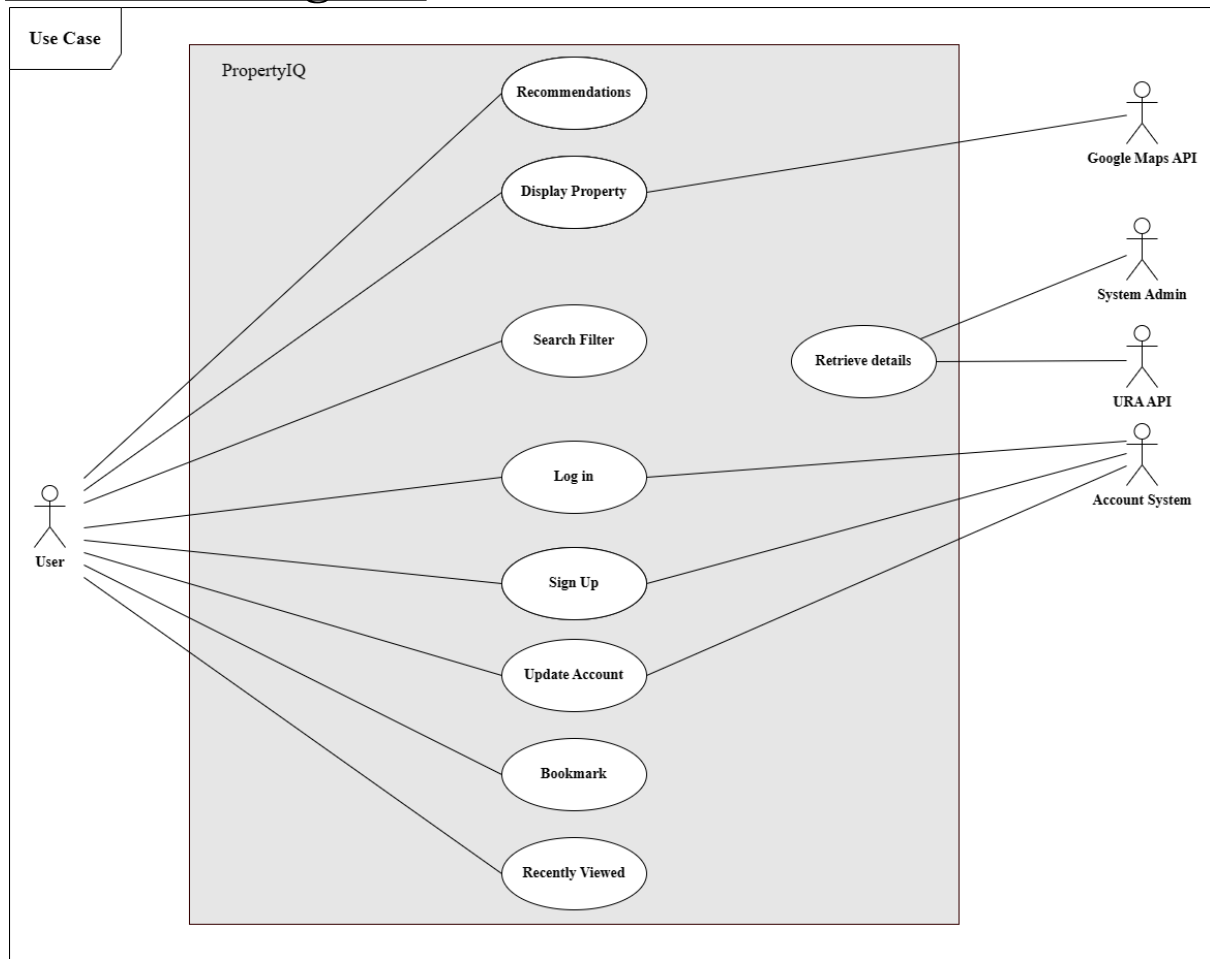
Version 1.0.1 approved

Prepared by Team 3.0

Nanyang Technological University

11/04/2023

Use Case Diagram



Use Case Descriptions

Legend:

Priority: Low, Medium, High

Frequency: Ranges from 1-5, 1 = Seldom used, 5 = Very Frequently used

| | | | |
|----------------|------------|--------------------|------------|
| Use Case ID: | UC_1 | | |
| Use Case Name: | Sign Up | | |
| Created By: | Wei Kang | Last Updated By: | Wei Kang |
| Date Created: | 03.02.2023 | Date Last Updated: | 11.04.2023 |

| | |
|--------------------|--|
| Actor: | User (Initiating Actor), Account System |
| Description: | Sign Up allows new guest users to create an account in our system database. Users must input their First Name, Last Name, Username, Email, and Password to create an account. |
| Preconditions: | 1. User must be on the sign up page. |
| Postconditions: | 1. System creates an account and stores account details in the database. 2. System displays the login page. |
| Priority: | High |
| Frequency of Use: | 3 |
| Flow of Events: | 1. User inputs required information: First Name, Last Name, Username, Email, Password. 2. Account System validates the information. 3. System creates the account. 4. System stores account information in the database. 5. System directs the user to the login page. |
| Alternative Flows: | <p>AF-S1-a. The email already exists: Email already taken.</p> <ol style="list-style-type: none">1. System displays "Email is taken!" error prompt.2. Continue from main flow step 1. <p>AF-S1-b. The user account already exists: Username already taken</p> <ol style="list-style-type: none">1. System displays "Username is taken!" error prompt.2. Continue from main flow step 1. <p>AF-S1-c. Password do not meet requirements.</p> <ol style="list-style-type: none">1. If the password inputted is not at least 8 characters long, system displays an error message prompt: "Invalid Password! Password must at least be 8 characters long."2. If the password inputted does not have a digit, system displays an error message prompt: "Invalid Password! Password must at least have 1 digit." |

| | |
|-----------------------|--|
| | <ol style="list-style-type: none"> 3. If the password inputted does not have a uppercase character, system displays an error message prompt: “Invalid Password! Password must at least have 1 uppercase letter.” 4. If the password inputted does not have a lowercase character, system displays an error message prompt: “Invalid Password! Password must at least have 1 lowercase letter.” 5. If the password inputted does not have at least 1 special character, system displays an error message prompt: “Invalid Password! Password must have at least 1 symbol (!@#\$%^&*_)”. 6. Continue from main flow step 1. <p>AF-S1-d. Email does not meet requirements.</p> <ol style="list-style-type: none"> 1. System displays “Please enter a valid email!” 2. Continue from main flow step 1. |
| Exceptions: | |
| Includes: | |
| Extends: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

| | | | |
|----------------|-------------------|--------------------|-------------------|
| Use Case ID: | UC_2 | | |
| Use Case Name: | Login | | |
| Created By: | Wei Kang | Last Updated By: | Wei Kang |
| Date Created: | 03.02.2023 | Date Last Updated: | 11.04.2023 |

| | |
|--------------------|---|
| Actor: | User (Initiating Actor), Account System |
| Description: | Login allows users to enter PropertyIQ to use features catered for registered users of the app such as bookmarking properties and viewing their recently viewed properties. The user must input their unique username and the corresponding registered password to log in to the system. |
| Preconditions: | 1. User must have an existing account in the database. |
| Postconditions: | 1. System displays the users' dashboard page. |
| Priority: | High |
| Frequency of Use: | 3 |
| Flow of Events: | <ol style="list-style-type: none"> 1. User enters username and password on the login page. 2. User submits details. 3. The Account System validates the user's username and password. 4. System displays the account dashboard page. |
| Alternative Flows: | <p>AF-S2-a. Invalid Username</p> <ol style="list-style-type: none"> 1. System cannot find the username in the database. 2. System displays an error message ("Unregistered Account! Register with PropertyIQ instead? "). 3. System redirects user to continue from Sign Up. (UC_1) <p>AF-S2-b. Invalid Password</p> <ol style="list-style-type: none"> 1. Password does not match for unique username. 2. System displays an error message ("Invalid credentials! Try again!"). 3. System decreases the number of attempts the user has left by 1. 4. System prompts user to input username and password again. 5. Continue from main flow step 1. <p>AF-S2-c. Missing Username/ Password</p> <ol style="list-style-type: none"> 1. System displays an error message ("Username/Password must be filled in"). 2. System prompts user to input username and password again. 3. Continue from main flow step 1. |
| Exceptions: | <p>EX-S2-a. Maximum 3 attempts exceeded</p> <ol style="list-style-type: none"> 1. System displays "Maximum attempts exceeded, try again in 1 hour" message. 2. System locks the user account for 1 hour. 3. System displays the locked out page. |

| | |
|--------------------------|--|
| Includes: | |
| Extends: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

| | | | |
|----------------|-----------------------|--------------------|-------------------|
| Use Case ID: | UC_3 | | |
| Use Case Name: | Update Account | | |
| Created By: | Wei Kang | Last Updated By: | Wei Kang |
| Date Created: | 03.02.2023 | Date Last Updated: | 10.04.2023 |

| | |
|--------------------|--|
| Actor: | User (Initiating Actor), Account System |
| Description: | Update Account allows the user to edit their account information such as username, first name, last name, email, password. |
| Preconditions: | 1. User must have an existing account. |
| Postconditions: | 1. New user information will be updated in the database. 2. System displays the account dashboard page. |
| Priority: | Medium |
| Frequency of Use: | 3 |
| Flow of Events: | 1. User inputs new input on the desired field that he/she wishes to update. 2. Account System validates the user input. 3. System updates information in the database. 4. System informs the user that the information has been updated. 5. System displays the account dashboard page. |
| Alternative Flows: | <p>AF-S3-a. The email already exists: Email already taken.</p> <ol style="list-style-type: none"> 1. System displays “Email is taken!” error prompt. 2. Continue from the main flow step 1. <p>AF-S3-b. The user account already exists: Username already taken.</p> <ol style="list-style-type: none"> 1. System displays “Username is taken!” error prompt. 2. Continue from the main flow step 1. <p>AF-S3-c. Password do not meet requirements.</p> <ol style="list-style-type: none"> 1. If the password inputted is not at least 8 characters long, system displays an error message prompt: “Invalid Password! Password must at least be 8 characters long.” 2. If the password inputted does not have a digit, system displays an error message prompt: “Invalid Password! Password must at least have 1 digit.” 3. If the password inputted does not have an uppercase character, system displays an error message prompt: “Invalid Password! Password must at least have 1 uppercase letter.” 4. If the password inputted does not have a lowercase character, system displays an error message |

| | |
|-----------------------|---|
| | <p>prompt: “Invalid Password! Password must at least have 1 lowercase letter.”</p> <p>5. If the password inputted does not have at least 1 special character, system displays an error message prompt: “Invalid Password! Password must have at least 1 symbol (!@#\$%^&*_)”.</p> <p>6. Continue from main flow step 1.</p> <p>AF-S1-d. Email does not meet requirements.</p> <p>1. System displays “Please enter a valid email!”</p> <p>2. Continue from the main flow step 1.</p> |
| Exceptions: | |
| Includes: | |
| Extends: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

| | | | |
|----------------|---------------------|--------------------|-------------------|
| Use Case ID: | UC_4 | | |
| Use Case Name: | Bookmark | | |
| Created By: | Nicholas Lim | Last Updated By: | Wei Kang |
| Date Created: | 02.02.2023 | Date Last Updated: | 10.04.2023 |

| | |
|-----------------------|---|
| Actor: | User |
| Description: | Bookmark allows the user to view, add, remove property that the user selected to bookmark. |
| Preconditions: | 1. User must currently be logged in |
| Postconditions: | 1. System displays the selected bookmarked property. |
| Priority: | Medium |
| Frequency of Use: | 3 |
| Flow of Events: | F-S4-a. User views bookmarks <ol style="list-style-type: none"> 1. User accesses the user dashboard page. 2. System retrieves bookmarked property from the database. 3. System displays all bookmarked property on the users' dashboard. 4. User selects a bookmarked property to view. 5. System redirects the user to the selected property listing. |
| Alternative Flows: | AF-S4-a. When user tries to view bookmark but has no bookmarked property <ol style="list-style-type: none"> 1. User accesses the user dashboard page. 2. System displays "You have not bookmarked any properties before!" on the users' dashboard. AF-S4-b. User adds a bookmark <ol style="list-style-type: none"> 1. User is viewing a property. 2. User opts to bookmark the property. 3. System saves the bookmarked property to the database. AF-S4-c. User deletes a bookmark <ol style="list-style-type: none"> 1. User is viewing a property. 2. User opts to remove the property from the bookmarked properties. 3. System removes the bookmarked property from the database. |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

| | | | |
|----------------|------------------------|--------------------|-------------------|
| Use Case ID: | UC_5 | | |
| Use Case Name: | Recently Viewed | | |
| Created By: | Nicholas Lim | Last Updated By: | Wei Kang |
| Date Created: | 02.02.2023 | Date Last Updated: | 10.04.2023 |

| | |
|-----------------------|---|
| Actor: | User |
| Description: | Recently Viewed allows the user to view all past viewed property. It also allows the user to delete all of the past viewed properties. |
| Preconditions: | 1. Users must currently be logged in |
| Postconditions: | 1. System displays the selected recently viewed property. |
| Priority: | Low |
| Frequency of Use: | 2 |
| Flow of Events: | <ol style="list-style-type: none"> 1. User accesses the Recently Viewed Properties page. 2. System retrieves all of the user's past viewed property from the database. 3. System displays all of the user's recently viewed properties. 4. User selects a property to view. 5. System redirects user to the selected property listing. |
| Alternative Flows: | <p>AF-S5-a. When the user has not viewed any property before</p> <ol style="list-style-type: none"> 1. User accesses the Recently Viewed Properties page. 2. System displays "You have not viewed any properties before!" on the page. <p>AF-S5-b. User deletes all recently viewed property</p> <ol style="list-style-type: none"> 1. User accesses the Recently Viewed Properties page. 2. User opts to delete all recently viewed properties. 3. System deletes all recently viewed properties from the database. 4. System displays "History successfully cleared!" prompt. |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

| | | | |
|----------------|------------------------|--------------------|-------------------|
| Use Case ID: | UC_6 | | |
| Use Case Name: | Recommendations | | |
| Created By: | Don Lim | Last Updated By: | Don Lim |
| Date Created: | 02.02.2023 | Date Last Updated: | 10.04.2023 |

| | |
|-----------------------|---|
| Actor: | User |
| Description: | Recommendations allows the user to view a selection of properties that are similar to the one they are currently viewing, based on the location, property type and rental price of the properties. |
| Preconditions: | 1. User has selected to view a property |
| Postconditions: | 1. System displays a list of recommended properties. |
| Priority: | High |
| Frequency of Use: | 5 |
| Flow of Events: | 1. User accesses the individual property listing page. 2. System computes a list of properties that are ideal alternatives to the property the user is currently viewing. 3. System displays recommended properties to the user. |
| Alternative Flows: | AF-S6-a. No suitable properties to recommend 1. System displays “No Recommendations Available” prompt. |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | 1. System prioritises recommendations based on certain criteria such as proximity to the selected property, property type and rent. |

| | | | |
|----------------|---------------------------------|--------------------|-----------------|
| Use Case ID: | UC_7 | | |
| Use Case Name: | Display Property | | |
| Created By: | Aditya Kumar Pugalia | Last Updated By: | Wei Kang |
| Date Created: | 03.02.23 | Date Last Updated: | 10.04.23 |

| | |
|-----------------------|---|
| Actor: | User (Initiating Actor), Google Maps API |
| Description: | Display Property provides property details from the database and displays it to the user. |
| Preconditions: | 1. User has selected to view a property |
| Postconditions: | 1. System displays the details of the selected property. |
| Priority: | High |
| Frequency of Use: | 5 |
| Flow of Events: | <ol style="list-style-type: none"> 1. System retrieves all property details of the property saved in the database. 2. System displays all property details on the property listing. 3. System computes recommended properties (UC_6). 4. System displays recommended properties to the user. 5. System authenticates with and accesses Google Maps API. 6. System displays property details of the selected property and recommended properties on Google Maps. 7. System calculates and forecasts rental flat prices for the project of the chosen flat and for the nearby rental housing. 8. System displays the Nearby Trend and Historical Trend on the property listing. |
| Alternative Flows: | |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

| | | | |
|----------------|---------------------------------|--------------------|-----------------|
| Use Case ID: | UC 8 | | |
| Use Case Name: | Retrieve Details | | |
| Created By: | Aditya Kumar Pugalia | Last Updated By: | Wei Kang |
| Date Created: | 03.02.23 | Date Last Updated: | 10.04.23 |

| | |
|-----------------------|--|
| Actor: | System Admin (Initiating Actor), URA API |
| Description: | Retrieve Details allows the PropertyIQ system admin to continually retrieve new rental housing data from the URA API and to update the information in the database. |
| Preconditions: | 1. System Admin is logged in with super user privilege and on the dashboard page. |
| Postconditions: | 1. The database and website is populated with new rental market data. |
| Priority: | High |
| Frequency of Use: | 1 |
| Flow of Events: | 1. System admin opts to update the properties. 2. System authenticates with the URA API using access key and secret key. 3. System retrieves data from the API. 4. System stores the property data in the system database. 5. System notifies all users through email that new properties have been populated on the website. 6. System redirects admin to the home page. |
| Alternative Flows: | |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

| | | | |
|----------------|--|--------------------|-------------------|
| Use Case ID: | UC_9 | | |
| Use Case Name: | Search Filter | | |
| Created By: | Andrada Angel John Bernardino | Last Updated By: | Wei Kang |
| Date Created: | 03.02.2023 | Date Last Updated: | 10.04.2023 |

| | |
|-----------------------|--|
| Actor: | User |
| Description: | Search Filter enables the User to search for properties by providing keywords and/or filtering by PROPERTY TYPE, BEDROOMS, RENTAL PRICE, and AREA. |
| Preconditions: | 1. User is currently on the main landing page. |
| Postconditions: | 1. System displays relevant Properties based on search query and filters applied. |
| Priority: | High |
| Frequency of Use: | 5 |
| Flow of Events: | <ol style="list-style-type: none"> 1. User enters search query in the search bar. 2. If filters are applied, system enhances the search query with the selected filters. <ol style="list-style-type: none"> a. If PROPERTY TYPE is used, properties corresponding to the selected PROPERTY TYPE are shown. b. If BEDROOMS is used, properties less than or equal to the number of BEDROOMS selected will be shown. c. If RENTAL PRICE is used, properties corresponding to the RENTAL PRICE range selected will be shown. d. If AREA is used, properties corresponding to the AREA range selected will be shown. 3. System will match the search query and the applied filters with the properties saved in the database to find corresponding suitable properties. 4. System displays suitable properties to the user. |
| Alternative Flows: | AF-S9-a. There are no properties matching the keywords of the search query or filter chosen in the database. <ol style="list-style-type: none"> 1. System displays “No Listings Available” prompt to the user. |
| Exceptions: | |
| Includes: | |
| Extends: | |
| Special Requirements: | |
| Assumptions: | |

| | |
|-------------------|--|
| Notes and Issues: | |
|-------------------|--|