Use Cases

for

HDB Resale PriceVisualiser

Version 1.1 approved

Prepared by Chaewon Kim (U2323002D)

Team 5

24 February 2025

Revision History

Name	Date	Reason For Changes	Version
Chaewon Kim	24/2/2025	First version, all cases	1.1

Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1. Created By

Supply the name of the person who initially documented this use case.

1.3.2. Date Created

Enter the date on which the use case was initially documented.

1.3.3. Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.3.4. Date Last Updated

Enter the date on which the use case was most recently updated.

2. Use Case Definition

2.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

2.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

2.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

2.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

2.5. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

2.6. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

2.8. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course". Example: X.Y.AC.1.

2.9. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by "EX" to indicate "Exception". Example: X.Y.EX.1.

2.10. Includes

List any other use cases that are included ("called") by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

2.11. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

2.12. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

2.13. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

3. Use Cases

3.1. Login/Signup

Use Case ID:	UC-ACCOUNT-01		
Use Case Name:	Login/Signup		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	User (Primary), Application (Secondary)
Description:	This use case allows the user to login to an existing account or signup to create a new account.
Preconditions:	1. The user must have already opened the website. 2. The user must have an established connection to the website's homepage, but not logged in.
Postconditions:	1. The user must be logged into their account
Priority:	1
Frequency of Use:	1 time per use of website
Flow of Events:	1. The user opens the website. 2. The user clicks on the "Log In" button on the right of the navbar. 3. The user enters their username/email and password. 4. The website authenticates the user. 5. Once authentication is successful, the website logs the user into their account. 6. The website goes back to the homepage, with the user now logged in.

Alternative Flows:	AF001.1: At Step 2, if the user does not click the "Log In" button. 1. The website displays the homepage. 2. When the user clicks on *any* button, the website automatically displays the login popup page. AF001.2: At Step 3, if the user clicks on "Continue as guest" instead of entering in their login details. 1. The website closes the login popup. 2. The website directs the user back to the homepage, where they can navigate the website as a guest. AF001.3: At Step 3, if the user has not created an account before. 1. The user clicks on the "Signup" button. 2. The website displays the "Signup" page, prompting the user to enter necessary details. 3. The application saves said details within the user database. 4. The website directs the user back to to the homepage, with the user now logged in. AF0001.4: At Step 3, if the user enters in their details incorrectly (i.e. incomplete email). 1. The website prompts the user to enter information correctly. 2. Goes to step 3. AF0001.4: At Step 4, if authentication fails. 1. The website will display an error message displaying the reason for failed authentication. 2. The user can either reattempt login or continue as guest.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.2. Forgot Password

Use Case ID:	UC-ACCOUNT-02		
Use Case Name:	Forgot Password		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	User (Primary), Application (Secondary)
Description:	This use case allows a user to reset their password in case they forget it.
Preconditions:	1. The user must have already opened the website. 2. The user must have access to the login popup page but cannot log in due to a forgotten password.
Postconditions:	1. A password reset email is sent to the user. 2. If the user follows the reset link and enters a new password, their account is updated with the new credentials.
Priority:	1
Frequency of Use:	1 time per use of application, as needed by users who forget their passwords
Flow of Events:	1. The user opens the website. 2. The user clicks on the “Log in” button on the right of the navbar. 3. The user selects “Forgot Password?” Below the login form. 4. The website prompts the user to enter their registered email. 5. The user enters their email and clicks “Send Reset Link.” 6. The website verifies whether the email exists in the user database. 7. If the email is found, the website sends a password reset link to the user’s email. 8. The user checks their email and clicks on the reset link. 9. The website loads the Reset Password Page, prompting the user to enter a new password. 10. The use entered a new password and confirms it. 11. The system validates the new password and updates the database. 12. The user is redirected to the login page with a confirmation message: “Your password has been successfully reset.”

Alternative Flows:	AF002.1: At Step 4, if the user enters an email not registered in the system: 1. The website displays an error message: "This email is not associated with any account." 2. The user is prompted to enter another email or sign up for a new account. AF002.2: At Step 10, if the user enters an invalid or weak password (eg. Less than 8 characters): 1. The website displays a message: "Password must be at least 8 characters long and contain a mix of letters and numbers" 2. The user re-enters a stronger password and clicks "Submit" again.
Exceptions:	• If the email server fails to send the password reset email, the website displays an error message: "Unable to send password reset email. Please try again later."
Includes:	• UC-ACCOUNT-01 (Login/Signup)
Special Requirements:	• The reset password link must expire after 30 minutes for security reasons. • The system should implement CAPTCHA verification to prevent bot abuse.
Assumptions:	• The user has access to the email account used for registration.
Notes and Issues:	• Consider implementing 2FA verification for an additional layer of security.

3.3 Change Account Details

Use Case ID:	UC-ACCOUNT-03		
Use Case Name:	Change Account Details		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Authenticated User (Primary), Application (Secondary)
Description:	This use case allows a user to update their username, email, or password while logged into their account.
Preconditions:	1. The user must have an authenticated account and be logged in. 2. The user must navigate to the account settings page.
Postconditions:	1. The user's updated details are saved in the database. 2. The system logs out the user if their password is changed, requiring them to log in again.
Priority:	1
Frequency of Use:	As needed by users who wish to update their details.
Flow of Events:	1. The user opens the website and logs in. 2. The user clicks on their profile icon in the navbar. 3. The website displays the account settings page. 4. The user selects the field to update (username, email, or password). 5. The user enters their new details and clicks "Save Changes." 6. The system validates the input (eg. Unique username, valid email format, strong password). 7. If validation passes, the system updates the user details in the database. 8. The system displays a confirmation message: "Your account details have been updated successfully." 9. If the user changed their password, the system logs them out, requiring them to log in again.

Alternative Flows:	AF003.1: At Step 6, if the user enters an email/username already registered to another account: 1. The website displays an error message: "This email/ username is already in use. Please choose another." 2. The user must enter a different email/username. AF003.2: At Step 6, if the user enters a weak password: 1. The system rejects the password and prompts for a stronger one.
Exceptions:	• If the server fails to update details, the system displays an error message: "Unable to update account details at this time. Please try again later."
Includes:	• UC-ACCOUNT-01 (Login/Signup)
Special Requirements:	• The system should send an email notification if the user changes their email or password.
Assumptions:	• Users will not need administrator approval to change their details
Notes and Issues:	N/A

3.4 Logout

Use Case ID:	UC-ACCOUNT-04		
Use Case Name:	Logout		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Authenticated User (Primary), Application (Secondary)
Description:	This use case allows the user to log out of their account and return to the homepage.
Preconditions:	1. The user must be logged into their account.
Postconditions:	1. The user is logged out and redirected to the homepage. 2. All session data is cleared for security purposes.
Priority:	1
Frequency of Use:	1 time per session.
Flow of Events:	1. The user clicks on their profile icon in the navbar. 2. The system displays a dropdown menu with the “Logout” button. 3. The user clicks “Logout”. 4. The system terminates the user session. 5. The user is redirected to the homepage.
Alternative Flows:	AF004.1: If the user attempts to access a restricted page (i.e. saved price comparisons) after logout, they are redirected to the login page.
Exceptions:	• If server issues prevent session termination, the user sees: “Logout failed. Please try again.”
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	• Consider implementing auto-logout after inactivity.

3.5 Search for HDB Prices

Use Case ID:	UC-SEARCH-01		
Use Case Name:	Search for HDB Prices		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows users to search for HDB resale prices based on different criteria such as location, estate, or flat type.
Preconditions:	1. The user must have already opened the website. 2. The user must have an active internet connection.
Postconditions:	1. The search results are displayed dynamically based on the user's query.
Priority:	1
Frequency of Use:	Multiple times per session.
Flow of Events:	1. The user enters a location, estate, or flat type into the search bar. 2. The user clicks the "Search" button. 3. The system queries the HDB Price Database for matching results. 4. The system displays the results dynamically in the TransactionsTable.js. 5. The user can further refine the search by applying filters (See UC-SEARCH-02)
Alternative Flows:	AF005.1: At Step 3, if no matching results are found: 1. The system displays: "No results found for your search. Try adjusting your criteria."
Exceptions:	• If the database is unavailable, the system displays an error message: "Service temporarily unavailable. Please try again later."
Includes:	• UC-SEARCH-02 (Filter Search Results) • UC-SEARCH-03 (Sort Listings)
Special Requirements:	• Autocomplete suggestions should be provided based on common search terms.
Assumptions:	• The search algorithm ranks the most relevant results at the top.
Notes and Issues:	• Consider implementing fuzzy search to handle minor typos.

3.6 Filter Search Results

Use Case ID:	UC-SEARCH-02		
Use Case Name:	Filter Search Results		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows users to refine their search results using filters such as price range, flat type, lease remaining, and estate.
Preconditions:	1. The user must have already searched for listings using the search bar. 2. The system must have retrieved matching results from the database.
Postconditions:	1. The search results update dynamically based on the selected filters.
Priority:	1
Frequency of Use:	Frequently used after performing a search.
Flow of Events:	1. The user selects filter options from the search results page. 2. The system applies the filters and updates the displayed listings. 3. The user can modify filters, and the results dynamically adjust. 4. If needed, the user can reset all filters to the default search view.
Alternative Flows:	AF006.1: At Step 2, if no listings match the filters: 1. The system displays: “No results found. Try adjusting your filters.”
Exceptions:	• If the filter system fails, the system displays: “Filters cannot be applied at this time.”
Includes:	UC-SEARCH-01 (Search for HDB Prices)
Special Requirements:	• Filters should be interactive and apply instantly without requiring a page refresh.
Assumptions:	• Users will prefer filtering by multiple criteria simultaneously.
Notes and Issues:	• Consider adding a slider for price range filtering instead of manual input.

3.7 Sort Listings

Use Case ID:	UC-SEARCH-03		
Use Case Name:	Sort Listings		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows users to sort search results based on criteria like lowest to highest price, lease remaining, and floor area.
Preconditions:	1. The user must have searched for HDB listings. 2. The system must have displayed results in a table format.
Postconditions:	1. The search results are sorted in real-time based on the user's selection.
Priority:	1
Frequency of Use:	Regularly used when browsing listings.
Flow of Events:	1. The user clicks the "Sort By" dropdown menu. 2. The user selects a sorting option (eg. Lowest Price First). 3. The system reorders the listings accordingly. 4. The user can change the sorting order or reset to default sorting.
Alternative Flows:	AF007.1: At Step 3, If there are no listings to sort: 1. The system displays: "No results available for sorting."
Exceptions:	• If the sorting system fails, the system displays: "Sorting is currently unavailable."
Includes:	UC-SEARCH-01 (Search for HDB Prices)
Special Requirements:	• Sorting should be instantaneous with no page reload.
Assumptions:	• Users expect ascending and descending sorting for all fields.
Notes and Issues:	N/A

3.8 View Transaction Details

Use Case ID:	UC-SEARCH-04		
Use Case Name:	View Transaction Details		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows users to view detailed information about a specific HDB resale transaction, including price history, lease details, and transaction date.
Preconditions:	1. The user must have searched for HDB listings. 2. The user must click on a specific transaction from the search results.
Postconditions:	1. The system displays a detailed breakdown of the transaction.
Priority:	1
Frequency of Use:	Multiple times per session.
Flow of Events:	1. The user selects a listing from the search results table. 2. The system retrieves the full transaction details from the HDB Price Database. 3. The system displays: • Estate name • Flat type • Transaction price • Date of transaction • Lease remaining • Historical price changes
Alternative Flows:	AF008.1: If transaction data is incomplete: 1. The system displays: “Some details may be missing for this transaction.”
Exceptions:	• If database retrieval fails, the system shows: “Unable to load transaction details. Please try again later.”
Includes:	• UC-SEARCH-01 (Search for HDB Prices)
Special Requirements:	• The transaction details page should allow easy navigation back to the search results.
Assumptions:	• All transactions in the database have a unique transaction ID.
Notes and Issues:	N/A

3.9 Compare HDB Prices

Use Case ID:	UC-SEARCH-05		
Use Case Name:	Compare HDB Prices		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows users to compare HDB resale prices based on different factors like district, town, and year.
Preconditions:	1. This user must have already opened the website. 2. The user must have searched for listings or accessed the Price Comparison page.
Postconditions:	1. A comparative analysis of HDB prices is displayed visually.
Priority:	1
Frequency of Use:	Multiple times per session.
Flow of Events:	1. The user clicks “Compare Prices” form the homepage or navbar. 2. The system displays a comparison interface. 3. The user selects a comparison variable (eg. District, town, year). 4. The system fetches historical price trends from the HDB Price Database. 5. The system generates a chart comparing selected variables. 6. The user can toggle between different metrics.
Alternative Flows:	AF006.1: At Step 4, if no data is available for a selected variable: 1. The system displays: “Insufficient data to generate a comparison for this selection.”
Exceptions:	• If data fetching fails, the system displays: “Unable to retrieve comparison data. Please try again later.”
Includes:	• UC-TRENDS-01 (View Recent Transactions) • UC-TRENDS-02 (Analyse Historical Price Trends)
Special Requirements:	• Interactive charts should be implemented for better visualisation.
Assumptions:	• The comparison model uses real-time and historical data.
Notes and Issues:	• Consider adding AI-driven insights for deeper analysis.

3.10 Interactive Map

Use Case ID:	UC-MAP-01		
Use Case Name:	View HDB Listings on Map		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary), Google Maps API (External)
Description:	This use case allows users to view HDB resale listings as map pins on an interactive map.
Preconditions:	1. The user must have searched for HDB listings. 2. The user must have enabled location services if they want to view flats nearby.
Postconditions:	1. The map interface displays HDB listings visually.
Priority:	1
Frequency of Use:	Regularly used by users browsing location-based data.
Flow of Events:	1. The user clicks “Interactive Map” on the homepage or navbar. 2. The system loads MapBoundary.js and fetches map data. 3. The system retrieves HDB listing coordinates from the database. 4. The system plots the listings as map pins. 5. The user can hover over a pin to view price details. 6. The user can click a pin to open detailed property information.
Alternative Flows:	AF010.1: At Step 4, if user has location services disabled: 1. The system prompts: “Enable location services for nearby listings.”
Exceptions:	• If the Google Maps API fails, the system displays: “Map data unavailable. Please try again later.”
Includes:	• UC-SEARCH-01 (Search for HDB Prices) • UC-SEARCH-04 (View Transaction Details)
Special Requirements:	• The map should include filter options for easier navigation.
Assumptions:	• Google Maps API will be used for geolocation services.
Notes and Issues:	• Consider implementing heat maps for better price visualisation.

3.11 Hover Over Map for Info

Use Case ID:	UC-MAP-02		
Use Case Name:	Hover Over Map for Info		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary), Google Maps API (External)
Description:	This use case allows users to hover over properties on the interactive map to view quick price information.
Preconditions:	1. The user must have opened the interactive map. 2. The system must have plotted HDB resale listings on the map.
Postconditions:	1. The system displays tooltip popups with property details.
Priority:	1
Frequency of Use:	Regularly used by users browsing location-based data.
Flow of Events:	1. The user moves their cursor over a map pin. 2. The system retrieves basic transaction details. 3. The system displays a tooltip with: • Flat type • Price • Lease remaining
Alternative Flows:	AF011.1: If the database does not contain price details for a specific listing: 1. The system displays: “No price data available.”
Exceptions:	• If the map API fails, the system displays: “Map data unavailable. Please try again later.”
Includes:	• UC-MAP-01 (View HDB Listings on Map)
Special Requirements:	• The tooltip should appear instantly with no delay.
Assumptions:	• The Google Maps API supports interactive hover effects.
Notes and Issues:	N/A

3.12 Click on Map Pin for Details

Use Case ID:	UC-MAP-03		
Use Case Name:	Click on Map Pin for Details		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Authenticated User (Primary), Application (Secondary)
Description:	This use case allows users to click on a map pin to view full details of a property listing.
Preconditions:	1. The user must have opened the interactive map. 2. The system must have plotted HDB resale listings on the map.
Postconditions:	1. The system redirects the user to the full transaction details page.
Priority:	1
Frequency of Use:	Frequently used by users who want more details on specific properties.
Flow of Events:	1. The user clicks on a map pin. 2. The system retrieves full transactions data. 3. The system opens a new page displaying: • Estate name • Flat type • Transaction price • Date of transaction • Historical price trends
Alternative Flows:	AF012.1: At Step 3, if transaction data cannot be retrieved: 1. The system displays: “Transaction details are currently unavailable.”
Exceptions:	• If the map interaction fails, the system displays: “Unable to load listing details.”
Includes:	• UC-MAP-01 (View HDB Listings on Map) • UC-SEARCH-04 (View Transaction Details)
Special Requirements:	• The new page should open in a modal or a separate page for better user experience.
Assumptions:	• Each listing has a unique transaction ID stored in the database.
Notes and Issues:	N/A

3.13 Highlight Cheapest Listing

Use Case ID:	UC-MAP-04		
Use Case Name:	Highlight Cheapest Listing		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows the system to automatically highlight the cheapest listing on the interactive map.
Preconditions:	1. The user must have opened the interactive map. 2. The system must have multiple HDB resale listings plotted.
Postconditions:	1. The cheapest listing is visually distinct (eg. Coloured differently).
Priority:	1
Frequency of Use:	Frequently used by users looking for the best deals.
Flow of Events:	1. The system retrieves all transaction prices in the search results. 2. The system identifies the lowest-priced listing. 3. The system highlights the listing. 4. The user can click the highlighted listing for more details.
Alternative Flows:	AF013.1: At Step 3, if multiple listings have the same lowest price: 1. The system highlights all matching listings.
Exceptions:	• If the database query fails, the system does not highlight any listing.
Includes:	• UC-MAP-01 (View HDB Listings on Map)
Special Requirements:	• The highlight colour should be distinct but accessible to colourblind users.
Assumptions:	• Users will use this feature to find affordable options quickly, or to pinpoint the lowest price based on time period, flat type, and/or estate.
Notes and Issues:	N/A

3.14 View Recent Transactions

Use Case ID:	UC-TRENDS-01		
Use Case Name:	View Recent Transactions		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows users to view a list of the latest HDB resale transactions.
Preconditions:	1. The website must be fully loaded. 2. The user must navigate towards the homepage.
Postconditions:	1. The most recent resale transactions are displayed in a table.
Priority:	2
Frequency of Use:	Passively viewed by users.
Flow of Events:	1. The system retrieves recent HDB resale transactions from the database, which is regularly updated. 2. The system displays the estate, flat type, price, and date.
Alternative Flows:	AF014.1: At Step 1, if transaction data is not available: 1. The system displays: “No recent transactions found.”
Exceptions:	• If database is offline, the system shows: “Transaction history unavailable.”
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	• No user interaction is required—purely informational.

3.15 Analyse Historical Price Trends

Use Case ID:	UC-TRENDS-02		
Use Case Name:	Analyse Historical Price Trends		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Public User (Primary), Application (Secondary)
Description:	This use case allows users to view historical trends of HDB resale prices over time.
Preconditions:	1. The user must have accessed the “Insights & Trends” section from the homepage.
Postconditions:	1. A trend graph is displayed based on historical data.
Priority:	1
Frequency of Use:	Occasionally used for research purposes.
Flow of Events:	1. The user navigates to the “Insights & Trends” page. 2. The user selects a time range (eg. last 5 years). 3. The system retrieves historical resale prices from the database. 4. The system generates a graph displaying price trends. 5. The user can then filter by location, features, rooms, etc.
Alternative Flows:	AF015.1: At Step 4, if no historical data is available: 1. The system displays: “No historical data found for this selection.”
Exceptions:	• If data fetching fails, the system shows: “Error retrieving price trends.”
Includes:	N/A
Special Requirements:	• Graphs should be interactive, allowing zooming and filtering.
Assumptions:	N/A
Notes and Issues:	• Consider adding machine learning-based predictions for price forecasting.

3.16 AI-Powered Insights

Use Case ID:	UC-TRENDS-03		
Use Case Name:	AI-Powered Insights		
Created By:	Chaewon Kim	Last Updated By:	Chaewon Kim
Date Created:	24-02-2025	Date Last Updated:	24-02-2025

Actor:	Authenticated User (Primary), Application (Secondary), AI Model (External)
Description:	This use case provides users with AI-driven price predictions based on market trends.
Preconditions:	1. The user must have searched for HDB prices or accessed the AI Insights page.
Postconditions:	1. The system generates a predicted price range for selected listings for future years.
Priority:	2
Frequency of Use:	Occasionally used for trend analysis.
Flow of Events:	1. The user clicks “AI-Powered Insights” on the homepage. 2. The system loads an AI Model API for trend prediction. 3. The system retrieves historical transaction data. 4. The AI model predicts future price trends. 5. The user views the forecasted prices.
Alternative Flows:	AF016.1: At Step 4, if the AI model fails to generate insights: 1. The system displays: “AI insights are currently unavailable.”
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	• Consider integrating machine learning models for accuracy.