Black Box Testing

Explore Page

#	Test Input	Expected Output	Actual Output
1.	Launch app with internet	Map loads with polygons	Map loads with polygons
2.	Launch app without internet	Map loads with without polygons	Map loads with without polygons
3.	Press polygon	Map zooms into the selected polygon	Map zooms into the selected polygon
4.	Press nearby polygon while specific polygon selected	Map centers onto selected polygon	Map centers onto selected polygon
5.	Press areas not in polygons (outside map)	System does not do anything	System does not do anything
6.	Pinch out/in	Map zooms in/out, clamped between 12 and 18	Map zooms in/out, clamped between 12 and 18
8.	Using pinch gestures to zoom in beyond limits	Map does not zoom in past limits	Map does not zoom in past limits
9.	Using pinch gestures to zoom out beyond limits	Map does not zoom out past limits	Map does not zoom out past limits
10.	Drag map	Map pans in that direction without lag	Map pans in that direction without lag
11.	Drag map outside bounds	Map does not pan out of bounds	Map does not pan out of bounds

Search Locations

#	Test Input	Expected Output	Actual Output
1.	Type valid location name into search bar	System list out matching results below search bar	System list out matching results below search bar
2.	Type invalid location name into search bar	System displays “No Locations found”	System displays “No Locations found”
3.	Nothing is typed into search bar	System list out all locations	System list out all locations
4.	Type valid location name with a filter that some location fulfil	System list out matching results below search bar	System list out matching results below search bar
5.	Type valid location name with multiple filters that some location fulfil	System list out matching results below search bar	System list out matching results below search bar
6.	Type valid location name with all filters selected	System list out matching results below search bar	System list out matching results below search bar
7.	Nothing is typed into search bar with a filter that some location fulfill	System list out matching results below search bar	System list out matching results below search bar
8.	Nothing is typed into search bar with multiple filters that some location fulfill	System list out matching results below search bar	System list out matching results below search bar
9.	Nothing is typed into search bar with all filters selected	System list out matching results below search bar	System list out matching results below search bar

Settings

#	Test Input	Expected Output	Actual Output
1.	Press ‘Import Data’ and select supported file type containing only valid data	System updates app with data System displays “Import successful”	System updates app with data System displays “Import successful”
2.	Press ‘Import Data’ and select supported file type containing relevant and irrelevant data	System updates app with relevant data System displays “Import successful”	System updates app with relevant data System displays “Import successful”
3.	Press ‘Import Data’ and select supported file type containing only irrelevant data	System does not update app System displays “Import successful”	System does not update app System displays “Import successful”
4.	Press ‘Import Data’ and select supported file type containing no data	System does not update app System displays “Import successful”	System does not update app System displays “Import successful”
5.	Press ‘Import Data’ and select unsupported file type	System displays “Failed to import data”	System displays “Failed to import data”
6.	Press ‘Export Data’, select file type and valid location is chosen	Data is saved into valid location	Data is saved into valid location
7.	Press ‘Export Data’, select file type but invalid location is chosen	System displays error message	System displays error message

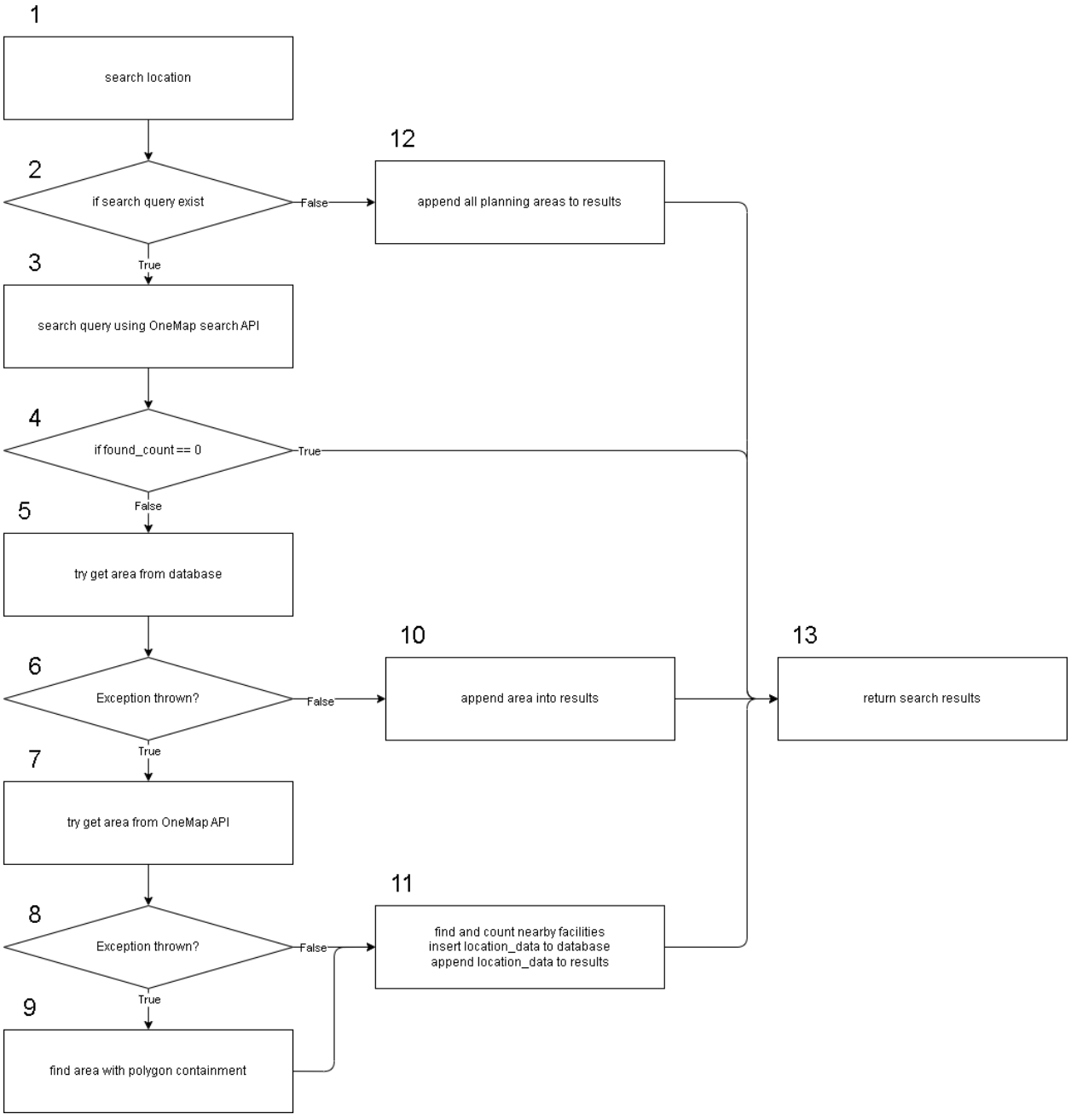
Compare

#	Test Input	Expected Output	Actual Output
1.	Unique locations selected in every slot	“Compare Locations” button is available	“Compare Locations” button is available
2.	Unique locations not	“Compare Locations”	“Compare Locations”

	selected in all slots	button is unavailable	button is unavailable
3.	Press “- Slots’’ while having more than 2 slots	System removes slot	System removes slot
4.	Press “+ Slots” while having less than 5 slots	System adds slots	System adds slot
5.	Press ‘- Slots’ while having 2 slots	‘- Slots’ button is greyed out when there are 2 slots.	‘- Slots’ button is greyed out when there are 2 slots.
6.	Press ‘+ Slots’ while having 5 slots	‘+ Slots’ button is greyed out when there are 5 slots.	‘+ Slots’ button is greyed out when there are 5 slots.

White Box Testing

Search Locations



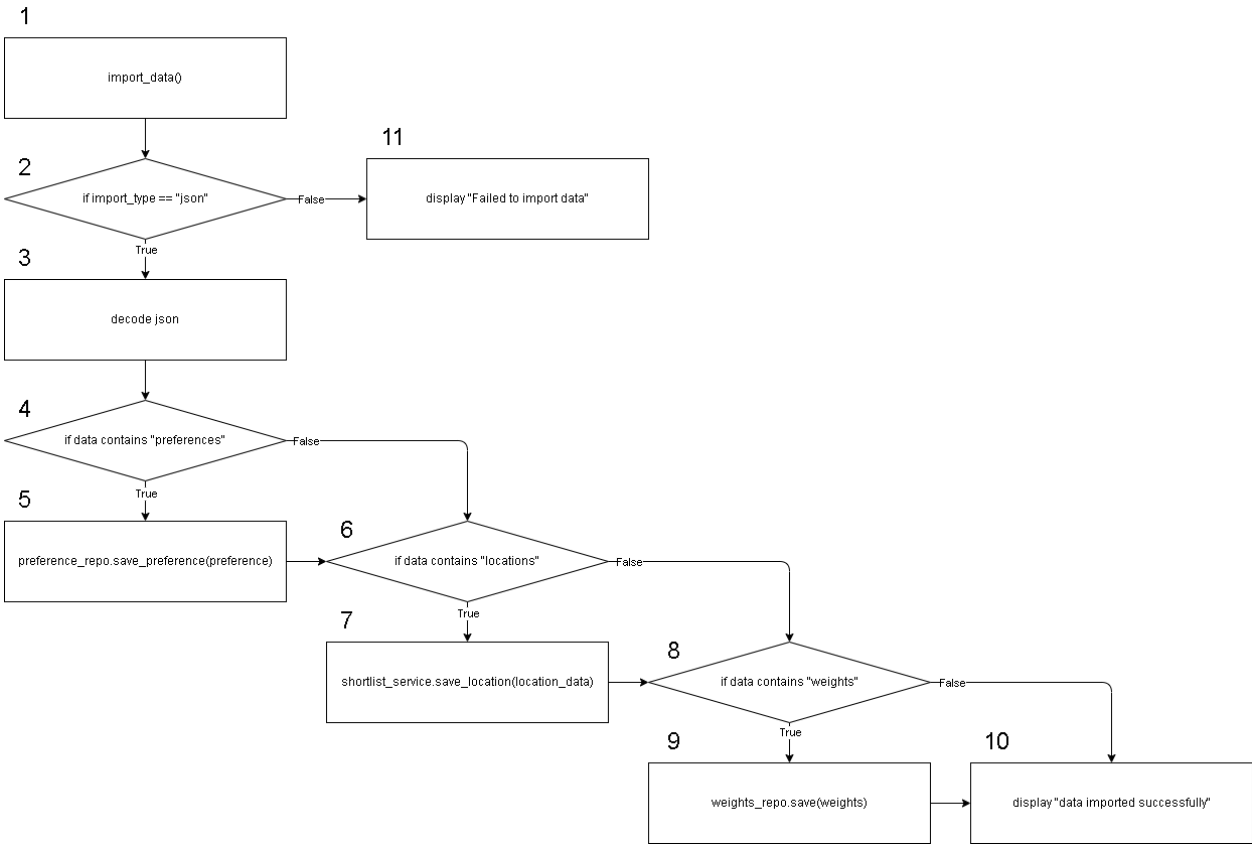
Cyclomatic Complexity = |decision points| + 1 = 4 + 1 = 5

Basic Paths

1. Basis path #1 (baseline): 1, 2, 3, 4, 5, 6, 10, 13
2. Basis path #2: 1, 2, 3, 4, 5, 6, 7, 8, 11, 13
3. Basis path #3: 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13
4. Basis path #4: 1, 2, 3, 4, 13
5. Basis path #5: 1, 2, 12, 13

#	Test Input	Expected Output	Actual Output
1.	Valid location name typed into search bar. Location is already in database	System list out matching results below search bar System prints(“Found n matching streets in street_geocode.db”)	System list out matching results below search bar System prints “Found 3 matching streets in street_geocode.db”,
2.	Valid location name typed into search bar. Location is not in database Planning area is available on OneMap	System list out matching results below search bar. System prints(“Found n unique streets from OneMap not in database”) System prints(“Adding m unique streets on OneMap”)	System list out matching results below search bar. System prints “Found 4 unique streets from OneMap not in database”, “Adding 4 unique streets on OneMap”
3.	Valid location name typed into search bar. Location is not in database Planning area not available on OneMap	System list out matching results below search bar. Database has location inserted	System list out matching results below search bar. Database has location inserted
4.	Invalid location name typed into search bar.	System displays ‘No Locations found’ System prints “No results found.”	System displays ‘No Locations found’ System prints “No results found.”
5.	Nothing is typed into search bar	System list out all locations	System list out all locations

Import Data



Cyclomatic Complexity = | decision points| + 1 = 4 + 1 = 5

Basic Paths

1. Basis path #1 (baseline): 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
2. Basis path #2: 1, 2, 3, 4, 5, 6, 7, 8, 10
3. Basis path #3: 1, 2, 3, 4, 5, 6, 8, 10
4. Basis path #4: 1, 2, 3, 4, 6, 8, 10
5. Basis path #5: 1, 2, 3, 4, 6, 8, 10
6. Basis path #5: 1, 2, 11

#	Test Input	Expected Output	Actual Output
1.	Import json file with only relevant data	System updates app with relevant data System displays “Import successful”	System updates app with relevant data System displays “Import successful”
2.	Import json file with 1 relevant data missing	System updates app with relevant data	System updates app with relevant data

		System displays “Import successful”	System displays “Import successful”
3.	Import json file with multiple relevant data missing	System updates app with relevant data System displays “Import successful”	System updates app with relevant data System displays “Import successful”
4.	Import json file with no relevant data	System does not update app System displays “Import successful”	System does not update app System displays “Import successful”
5.	Import json file with no data	System does not update app System displays “Import successful”	System does not update app System displays “Import successful”
6.	Import invalid file type	System displays “Failed to import data”	System displays “Failed to import data”

Speaker: Jet
Good afternoon, everyone!

Do you hate drowning in spreadsheets, conflicting advice from friends or always having the innate fear of choosing your future home? Finding the perfect place to live can feel like a high-stakes treasure hunt without a map.

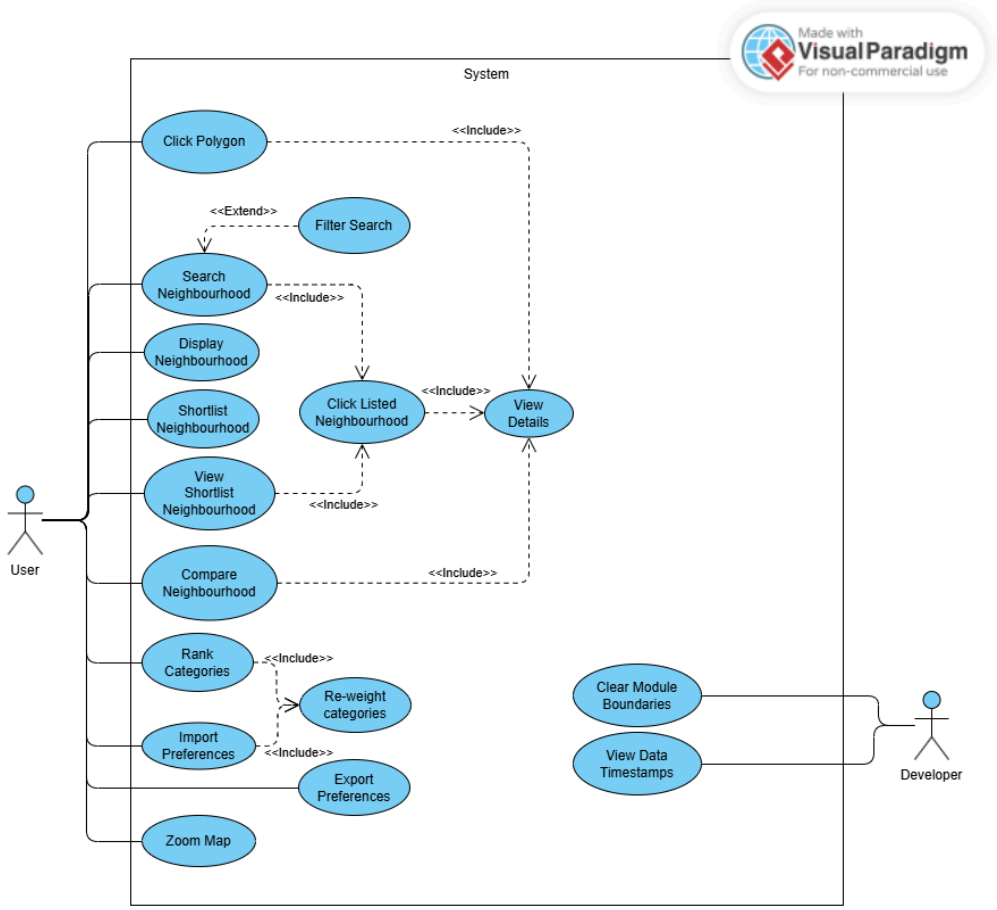
(Pause for effect)

But what if your phone is that treasure map? You're already trying to find a job in this economy post-graduation; why add the stress of finding a place to live? You don't have time for that! So today, we are proud to introduce LivaSG, the app that turns precious house-hunting time into job-hunting time!

Let us walk you through our app features with our use case diagrams, outlined and clearly demonstrated, followed by the overview of the application, UI, analytics, and functions!

Speaker: Xi Quan

1. **Explanation of Use Case Model**



Use Case	Function
Exploring Properties	
Click Polygon	1) Start with Explore Page: Enter Explore Page to view Singapore map 2) Click Areas: Select any polygon to see basic information and can click "View properties" to view more details
View Details	1) View Details: Click "Details" for comprehensive area analysis 2) View Ratings: Click "Ratings" for breakdown of categories scores
Display Neighbourhood	1) Display Neighborhood: The user can view a list of all available neighbourhoods.
Advanced Search	
Search Neighbourhood	1) Access Search Function: Use search bar at explore page 2) Apply Filters: Select preferred amenities 3) Review Results: Click on search results for detailed views
Personalised Recommendations	
Rank Categories	1) Set Preferences: Navigate to Preferred tab on navigation bar 2) Rank Categories: Order the 5 evaluation categories by importance
Re-weight Categories	1) Re-weight categories: Recalibrate scores with new weights
Comparison Features	
Compare Neighbourhood	1) Select Locations: Choose areas to compare in Compare tab on navigation bar 2) View Radar Charts: Visual comparison of category scores

	3) Analyse Differences: Overlay of price trend and radar chart
Data Management	
Shortlist Neighbourhood	Bookmark Locations: Save areas of interest for quick access
Import Preferences	Import Settings: Restore previous configurations from previous setup or device
Export Preferences	Export Data: Download your configurations onto your device

Speaker: Calvin

1. Launch Page

First impressions matter, and we want our app to work on any device. Part of our Non-Function Requirements is usability: Our app works on any screen size **(resize screen)**, and we have an accessible colour palette, alongside clear icons and user interface.

But before you even use the app, we want to make your experience as customised as possible. When you launch the app for the first time, you get to choose your priorities in our Preference page, the heart of our app.

Our solution is tailored according to your needs; beyond graduates looking for the cheapest place to live, whether you're a fitness fanatic who needs a convenient gym **(someone act like they are carrying weights)**, a foodie craving diverse options (**someone act like they are hungry and eating)**, a parent prioritising their children's education **(someone act like mom rocking a baby)** or a simple nature enthusiast, we have a just the recommendation for you.

With our five categories: Affordability, Accessibility, Amenities, Environment and Community, you get to rank your priorities. With 120 different permutations, our app can cater to your nicest desires!

If you change your priorities halfway, you can always return to this page under the "preference" tab! For now, hit Save when you're done. Your preferences will immediately be processed by our backend, and you'll be brought to the feature that makes our app so special: the Explore Page! .

2. Explore Page

Behold, a map you can call your personal real estate consultant! Our map is designed to be fast- taking less than 2.5 seconds to load the map when the app is launched!

(Gesture to the polygons)

These colourful shapes you aren't just a pretty face, they're live data from the OneMap API, dynamically rated just for you. The legend here tells you everything **(point to legend at bottom right of screen)**. Green areas are highly aligned to your preferences. Yellow is "maybe material" and red… Well, it's not completely disastrous, but more so less recommended relative to other locations based on your preference. We call this the "Relative Rating Scale", because one's paradise could be another's nightmare!

3. Clicking a Polygon

(Click on Bukit Panjang polygon)

Now, suppose this green zone catches your eye. A single click redirects you to a zoomed in map. With the polygon clearly highlighted, you'll know exactly where you're looking without being overwhelmed. Here you are presented with 2 powerful options: viewing the district's analytics, or viewing its facilities.

Let's start with the cold hard facts: the Analytics.

Speaker: Sam

4. View Analytics

(Click on "Analytics")

This is where we get smart. We collate all the analytics automatically from multiple APIs just for you, so you no longer have to bury your head in piles of research and you can use that time to doom scroll or catch up on your favourite shows.

The Radar Chart gives you a crystal-clear breakdown of how your selected location scores across all five key categories. And these scores are not randomly generated! The Affordability is calculated from a non linear curve of price data, Accessibility is calculated based on how well connected the location is with MRT,bus and carpark nodes, Amenities is calculated by just the raw numbers of schools,sports,hawkers ,healthcare facilities, Environment is calculated by

presence of parks and lastly Community is calculated by Community Centre presence. Yeah.. we are crunching a lot of data on your behalf!

Of course, we also have the one thing we believe everyone is concerned about: the price trend. You get to watch how the housing market flows! Control what you see by filtering the time range, and (hover/tap) on individual points to get exact prices. No more guessing if now is a good time to buy, you'll know instantly.

5. View Facilities

(Return to Specific View and click "Facilities")

Beyond numbers, sometimes you just want to know if there is an MRT within walking distance. The Facilities tab allows you to do just that.

We've zoomed in even further and plotted out every key amenity. You can filter to show just what you care about, from schools, to parks, to transport. Want to know more about each amenity? Clicking on the pinpoint allows you to see the name of that specific school or gym and its exact location on the map.

6. Saving Your Favourites and the Saved Tab

(Return to Specific View)

Found a winner? Or maybe just a "maybe"? You can hit "Save" on the top right corner of the screen. **(Click Bookmark)** This bookmark is waiting for you in the Saved tab. **(Click Saved Tab)** Here, you can search through all your bookmarks, filter them and view its details, and if you inevitably change your mind, because honestly, we know you are indecisive, you can simply remove it through a single tap or remove it from the Saved Tab. **(Run through the features)**

7. Preference Tab

But what if your priorities shift? What if you get a dog and suddenly parks are your new number one? Just pop back into Preferences and move Environment into the top spot.

(Click on Preference Tab and shift Environment to 1st place)

When you update them, the magic happens instantly. You'll literally see the colours shift as your new lifestyle kicks in. It's like watching your personality update in real-time.

Speaker: Loong Kiat

8. Search Page

But maybe you think districts are too big, and you want to narrow down your results. Our application allows you to search for specific locations. Just key in a specific address or utilise our search filters **(Click "Near MRT" and "Good Schools" in filters, search "Ang" and click "Ang Mo Kio Ave 1")** and you will be able to view the same details **(hover over the analytics and facilities buttons)**. Just like before, if you like it, you can save it. Just a small note, every school is a good school so don't be surprised if you see unfamiliar schools. Of course, you can always return and revert to clicking other polygons, and continue your exploration from there. **(Click other polygons)**

9. Compare Tab

Now, this is where decisions get made, where your top contenders go up against each other in a head to head battle.

Just add the locations you want to compare in each slot. If 2 locations aren't enough, you can always compare more, up to 5. Or remove it if it's too many- we aren't your parents! **(add until 5 and remove back to 2)** If you try to compare the same area, well you can't. **(Point to greyed out option)** We will stop your mistake before it happens. Once you're ready, click compare and behold! All comparisons are on the same view, you can see how their scores square up against each other or in our case radar against each other, and their price trends lining up together. **(Type "b" and click BAYSHORE ROAD" And "Bedok" in suggested)**

10. Settings Page

(Click on Explore Tab then Settings)

Finally, we believe you should own your data. In Settings, you can export all your preferences and saved locations as a clean JSON file. Back it up, send it to your partner for review or whatever you need.

And if you're switching phones or anything like that, you can import that JSON file to restore the app to the way you liked it, or you can choose to reset it all, for a fresh start. We're flexible, not clingy.

Speaker: Mun Kuan

And there, we present our working prototype.

1. Requirements Elicitation and Analysis

It all starts with requirements elicitation. We were meticulous. Every requirement was atomic, verifiable and unambiguous. No room for "might" ot "maybe". Our data dictionary and continual refinement killed uncertainties before a single line of code was written.

This clarity flowed directly into our requirement analysis, where we iteratively refined our Class Diagrams, Sequence Diagrams and Dialog Map. When there were issues, we clarified.

2. System Design

For the system's architecture, we chose a 4 Layered Structure.

The rules were simple. The upper layers can call lower layers, but never the other way around. This helped eliminate dependencies and reduced coupling. Hence, giving us reusability in the lower layers and providing upper layers extensibility, meaning we can build new features on top of this stable foundation without breaking whatever's below. **(Explain with Search and Time Range filter)** As a result this allowed for improved maintainability of out system, which clearly aligns with our non-functional requirement.

3. Implementation

With this blue print set, we moved to implementation, adopting the Scrum method. We detailed our product backlog, delegated them, and met after each sprint. We built the product incrementally, adapting to challenges that may arise along the way.

4. Testing

We ensured that testing was not an afterthought, considering it throughout the entire life cycle. From unit tests on individual components to integration tests ensuring both frontend and backend worked seamlessly together. Hence, allowing us to deliver such high quality and reliable experience you saw previously.

Speaker: Jet

5. Challenges

Finally, we faced a lot of technical challenges due to the nature of the data needed to be processed in this application. To interpret the coordinates for the polygon, we needed to implement the svy21 to wgs84 function, and had point-in-polygon checks, and polygon processing from GeoJSON. To optimize performance, we added different ways for the app to cache data, such as disk caching and api response caching. Multiple extensive error handling is needed throughout the codebase. A complex ranking algorithm was also needed to cater to the actual use case of the app. Lastly, the backend frontend integration was also complex given the amount of API response models needed.

Speaker: Mun Kuan

Conclusion

To conclude, what my group has created today is a powerful synergy between fantastic user experience coupled with rock-solid engineering foundation. We have built a platform that transforms the chaos of home hunting into confident decisions, and we're excited to share it with you. So, are you ready to take a leap and join us in our journey in finding a LIVA-ble home?

Thank you!