Quality Report

Member	UPI
Jimmy Wang	swan825
Dylan Xin	dxin779
Peter Zhong	tzho416

SOLID principle:

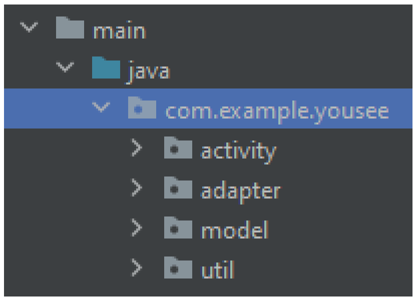
DataProvider and Model are not directly used by the activity classes, as we have used interfaces to achieve dependency injection. This is the same as what we proposed in the Design Doc, as we know that this is an important part of our design that increases the maintainability of our code.

Coding Practices:

For this project, we wanted to adapt good code base management and versioning to ensure the quality of our code. To achieve this, we used Github “branch on feature” and frequent pull requests which allowed us to simplify the process of providing and receiving feedback from other group members. For each pull request, the changes will be reviewed and approved before merging to the main branch; this was done to ensure the main branch is always in a valid working state.

To easily track the changes each of our members made, we followed the convention to add tags before our commit messages. These tags were usually [feat] and [fix] to indicate what type of change it was. Additionally, we also used a third-party application called ClickUp to set deadlines and track our progress throughout the project.

Throughout the project, we consistently used naming conventions such as pascal case for class names and camel case for methods and fields. We avoided using ambiguous and non-descriptive variable names to improve the readability of our code. For the package structure, we divided the classes into four categories being “activity”, “adaptor”, “model” and “util”. This helps us separate the functionality of our application and ultimately makes it easier to manage. An image of the package structure for our project can be seen below.



Plan Schedule Changes:

The implementation of the application was completed as scheduled. However, the quality report is started 6 days later than the scheduled date due to a large number of assignments all of us have in other courses.

Database Scheme Changes:

Instead of having three separate collections for each item's category, we are only using one collection to hold all the items. The document in this collection will have an extra field called itemType which represents which category it belongs to. This will be used during the conversion from document to java object (GPU/CPU/RAM) in the DataProvider to ensure a correct constructor is called to construct the object.

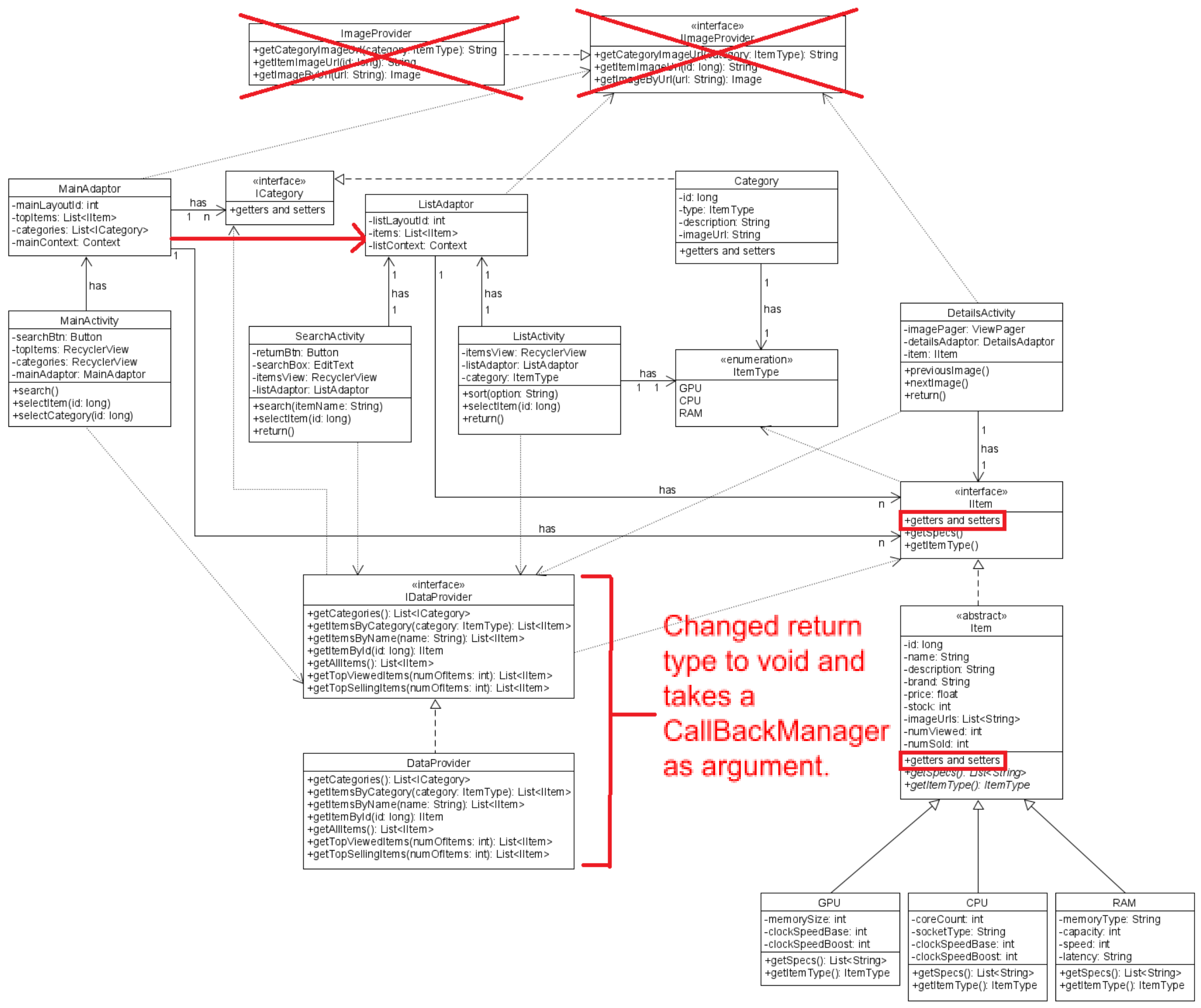
Class Diagram Changes:

Throughout development we have made a few changes and improvements to our class diagram. To ensure GUI responsiveness, we changed the Data provider to return a CallBackManager interface instead of a list containing the data. This allows the data to be fetched asynchronously and the calling activity to specify a callback action once the requested data have been received. For example, the Top Picked RecyclerView inside the MainActivity needs to be populated with data once the data has been retrieved, so the method populateTopRated will be a callback function that is passed to the DataProvider to be executed once the fetch is successful.

Another change is to the model methods, we added setters to all the fields in the item class which allows Firebase to use them for parsing between document and java objects.

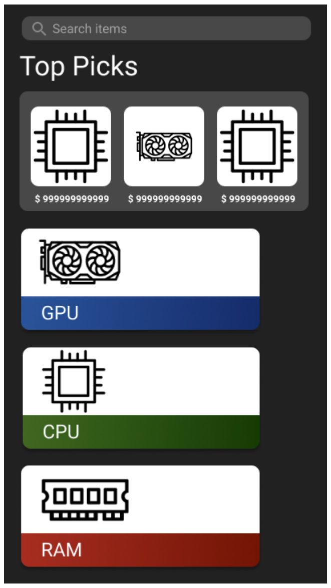
The Image provider was removed because we found that it was more convenient to store the images in the drawable file locally and retrieve the relevant images by matching the image ID stored in firebase.

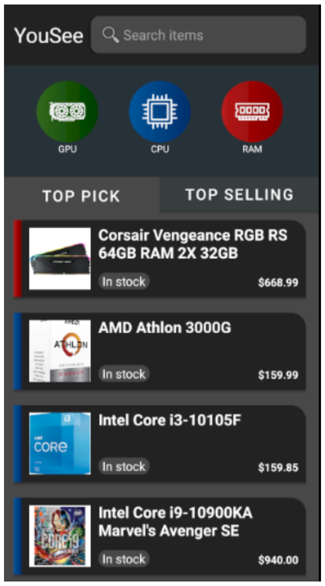
After seeing a lot of amazing demos from other groups, we decided to improve the layout of our MainActivity. This change resulted in the MainActivity reusing our ListAdaptor to populate the RecyclerView for our top selling and viewed items. This change also made the UI look more consistent across all the pages. A better representation of the changes made to our class diagram during development can be seen below.

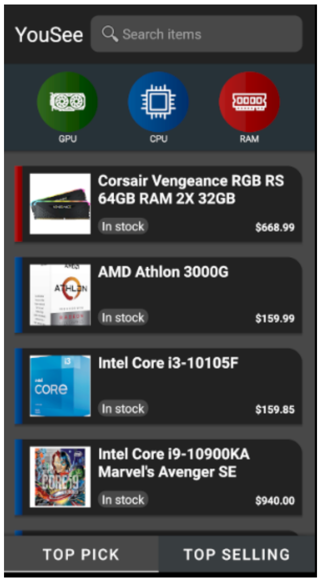


GUI changes:

MainActivity







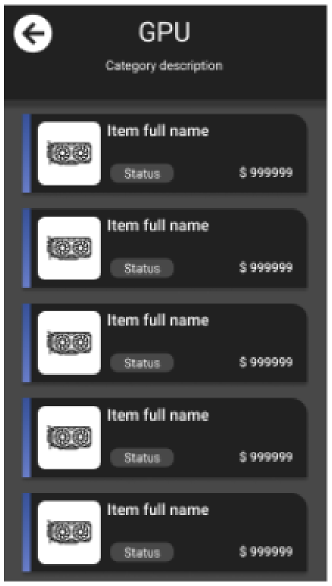
Version 1 Version 2 Version 3

Version 1 shows our initial design of the MainActivity. The problem with this design is that the cardviews of categories are too big, especially on smaller devices. The cardviews are not only aesthetically unpleasant but also do not fit in the screen without scrolling.

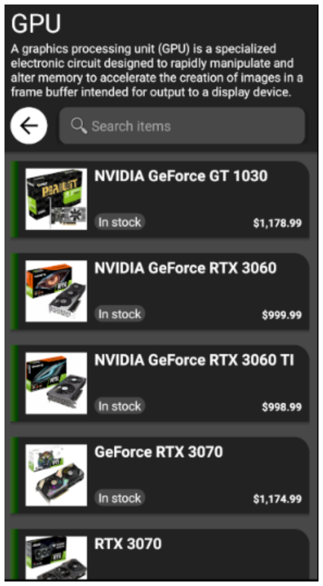
We improved our design in the second version of the MainActivity. The app name is now included in the app bar which is necessary but missing in the first version. We now have three icon buttons with labels in a horizontal recycler view to display our categories. This is much cleaner than the gigantic cardviews in the first version. The colors of the CPU and GPU are swapped which is simply a design change.

The feedback from Reza (the “top pick” and “top selling” tabs are confusing) after our demo leads to the final version of the MainActivity. The main change is that we moved the tabs of “top pick” and “top selling” to the bottom of the screen as Reza suggested. The app bar will be collapsed when the user scrolls down and the categories list will stick on the top of the screen. There are also some minor improvements like paddings and a white line at the bottom of the selected tab to indicate that it is currently selected.

ListActivity



Version 1

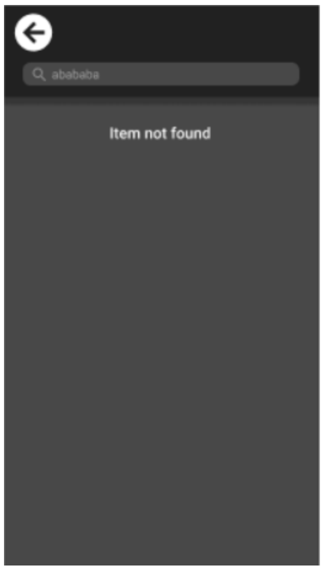


A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.

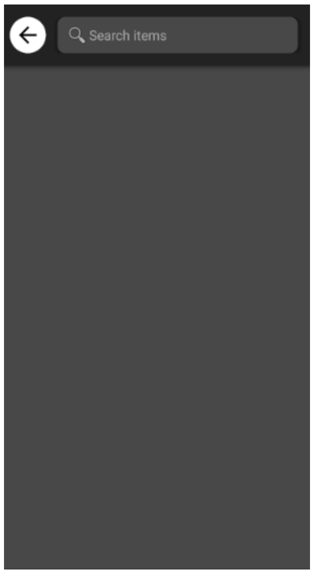
Version 2

In the first version of the ListActivity, a long category description would make the app bar very large and cause the app to be unusable in landscape mode. We had a workaround which is to set the maxLines attribute of the textview to 2 lines only. However, it is obvious that this is not a good solution, as the user will not be able to see the full description. We wrapped the category name and description in a CollapsingToolbarLayout so they can be hidden as the user scrolls down to view the items. It would be better to wrap the description in a ScrollView and set the maximum height of the toolbar. However, we decided that the description should not be too long anyway. Therefore, we neglected this potential issue of the app bar becoming too large as the app would still be usable because it can collapse. The search field for searching items within the category is a new feature added. The horizontal layout that contains the return button and the search field has a consistent design as in the SearchActivity and they will stick on top of the screen when the app bar is collapsed.

SearchActivity


ababab a

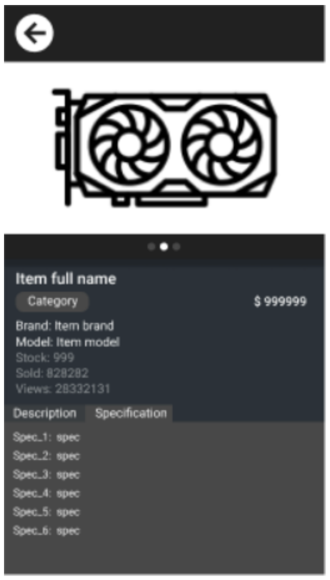
Item not found



Version 1 Version 2

We aligned the back button and search field horizontally. This removed the unnecessary blank space on the top of the app bar.DetailActivity

SearchActivity

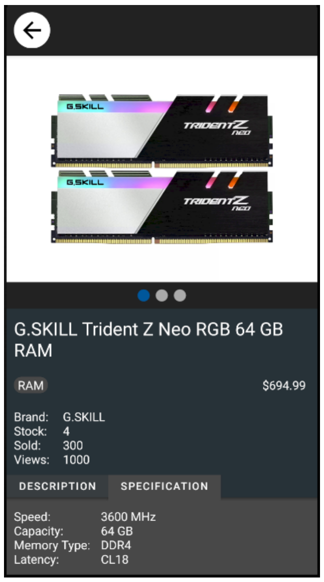

Item full name

Category $ 999999

Brand: Item brand
Model: Item model
Stock: 999
Sold: 828282
Views: 28332131

Description Specification

Spec_1: spec
Spec_2: spec
Spec_3: spec
Spec_4: spec
Spec_5: spec
Spec_6: spec

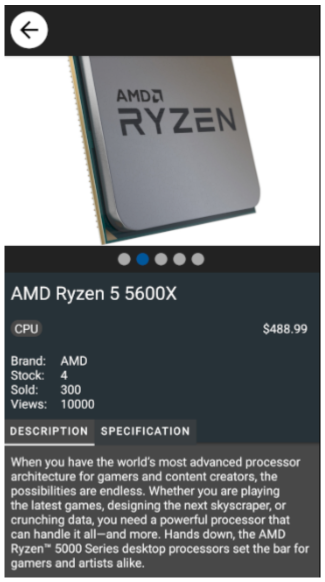

G.SKILL Trident Z Neo RGB 64 GB RAM

RAM $694.99

Brand: G.SKILL
Stock: 4
Sold: 300
Views: 1000

DESCRIPTION SPECIFICATION

Speed: 3600 MHz
Capacity: 64 GB
Memory Type: DDR4
Latency: CL18


AMD Ryzen 5 5600X

CPU $488.99

Brand: AMD
Stock: 4
Sold: 300
Views: 10000

DESCRIPTION SPECIFICATION

When you have the world's most advanced processor architecture for gamers and content creators, the possibilities are endless. Whether you are playing the latest games, designing the next skyscraper, or crunching data, you need a powerful processor that can handle it all—and more. Hands down, the AMD Ryzen™ 5000 Series desktop processors set the bar for gamers and artists alike.

Version 1 Version 2 Version 3

The second version of the DetailsActivity has some small changes to the first version. The alignment of the textviews of the specifications is improved. We also changed the text color to be all white as the grey color used in the first version is a little unreadable. There are no major changes to the Details Activity after the demo, the only change was to make the tab's background consistent with the tabs in the MainActivity, with a white line in the bottom to indicate it has been selected.