

Description

Intended User

Features

User Interface Mocks

Screen 1 – Main Screen – My wish list

Screen 2 – Menu in the main screen

Screen 3 – Item Search (Phone)

Screen 4 – Search a store (Phone)

Screen 5 – Item search results

Screen 6 – Item search results – Add quantity to shopping cart

Screen 7 – Add item to wish list

Screen 8 – Shopping Cart

Screen 9 – Checkout summary

Screen 10 – Shopping History

Screen 11 – Settings

Screen 12 – Settings (Change language)

Screen 13 – Settings (Change number of purchases in the Shopping History)

Screen 14 – Notifications

Screen 15 – Search items (Tablet)

Screen 16 – Search store (Tablet)

Screen 17 – Search results (Tablet)

Screen 18 – Add wishlist item (Tablet)

Key Considerations

How will your app handle data persistence?

Describe any edge or corner cases in the UX.

Describe any libraries you'll be using and share your reasoning for including them.

Describe how you will implement Google Play Services or other external services.

Next Steps: Required Tasks

Task 1: Prepare the data

Task 2: Project Setup in Android Studio

Task 3: Implement the database components

Task 4: Implement the UI for Wish list

Task 5: Implement the UI for search items in store

Task 6: Implement the UI for search a store

Task 7: Implement the UI for search item results

Task 8: Implement the UI for shopping cart

Task 9: Implement the UI for shopping history

Task 10: Implement the UI for Settings

Task 11: Implement the UI updates based on the shared preferences

Task 12: Implement the UI for notification list

Task 13: Implement the service to handle Firebase Cloud Messaging messages

Task 14: Implement the UI layout for Checkout summary

GitHub Username: [peterkwan](#)

My Supermarket

Description

“My Supermarket” – A mobile app that provides customers a better user experience in buying items in the supermarket, from building a wish list and searching for item’s availability and details in the store before the customer comes, tracking the status when the customer is in the store, for example, an item has been picked up by the customer, and checkout when they leaves the store.

Intended User

The intended users of this app are supermarket customers

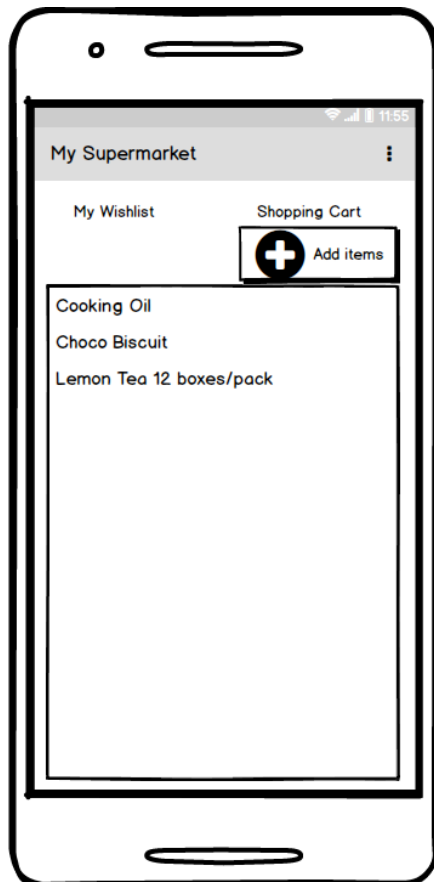
Features

Features include

- Building a Wish list
- Search for item availability and details in nearby store and selected store
- Building a shopping cart
- Track the status when an item is picked
- Notify customer for the items they checked out when they leaves
- Push notifications to customers for new products and promotions that may interest them.

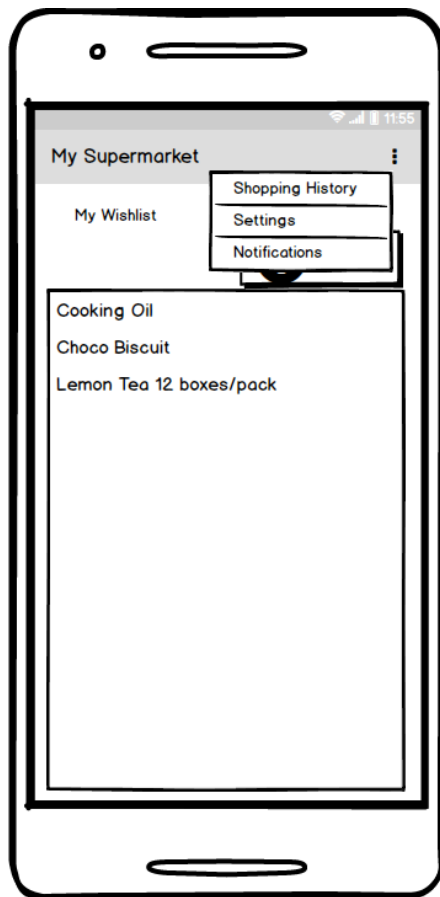
User Interface Mocks

Screen 1 - Main Screen - My wish list



The main screen will show two tabs, "My Wish list" and "Shopping cart". "My Wish list" tab will be selected by default. The lists at the bottom will show the list of items that the customer wishes to buy. They may click "Add item" to add an item to the list, click on the item to search for the availability in the nearby store or a selected store, and wipes left or right will remove the item from the list.

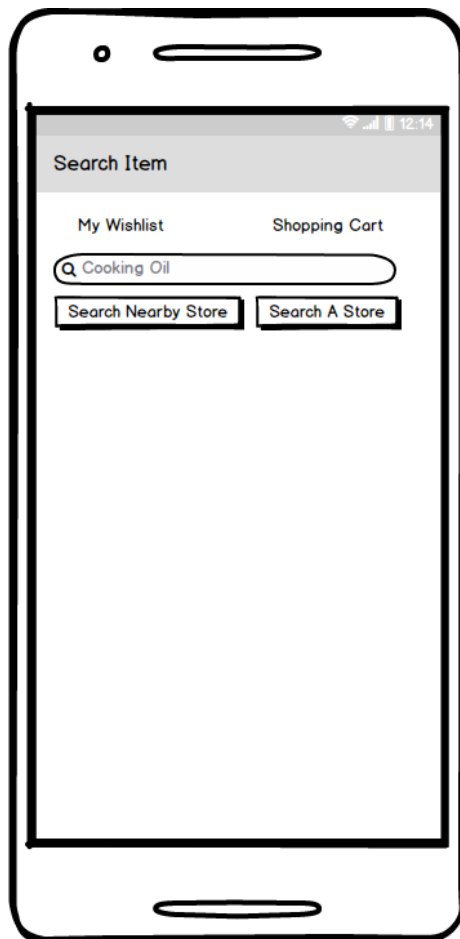
Screen 2 - Menu in the main screen



The menu will show three items – “Shopping History”, “Settings” and “Notifications”.

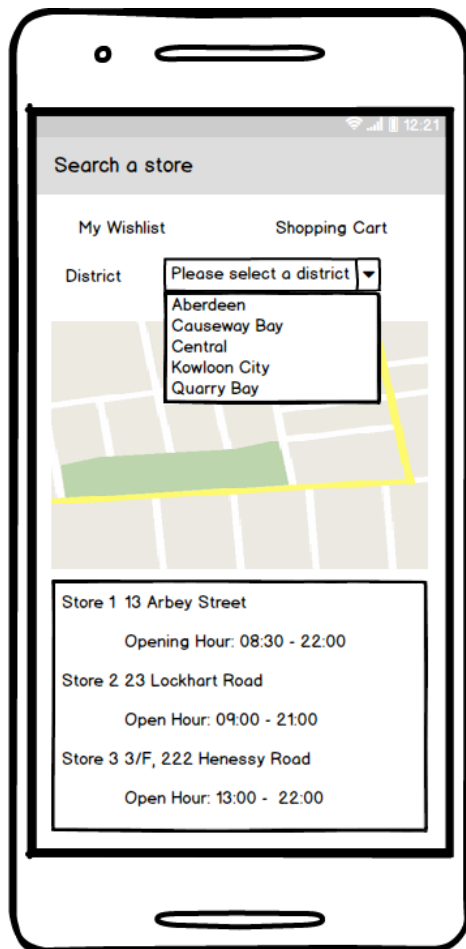
- “Shopping History” will show a list of previous purchases.
- “Settings” shows the preferences in the app.
- “Notifications” shows the list of push notifications (e.g. promotions, new products)

Screen 3 - Item Search (Phone)



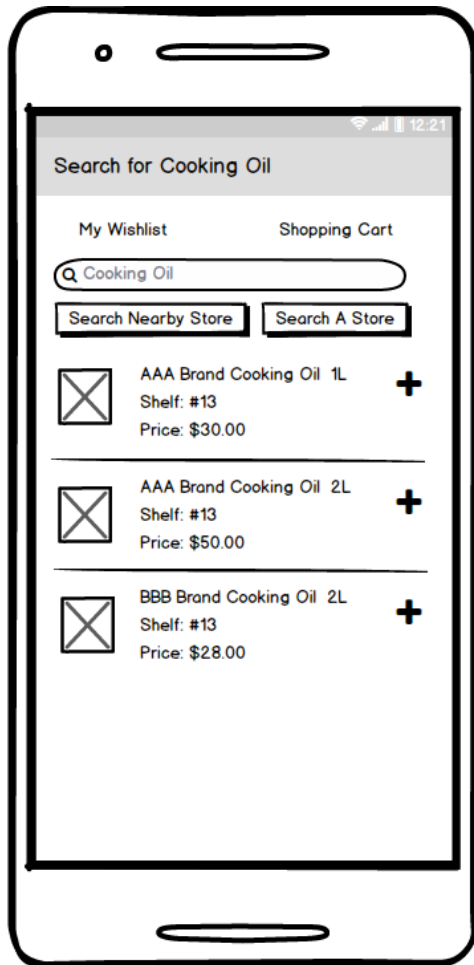
This screen is shown when an item in the wish list (Screen 1) is clicked. In this screen, the item name will be shown in the search box. "Search nearby store" will be enabled only when there is a store within 200 m from the location that the customer is at. "Search a store" will select a store and search for the item in that store.

Screen 4 - Search a store (Phone)



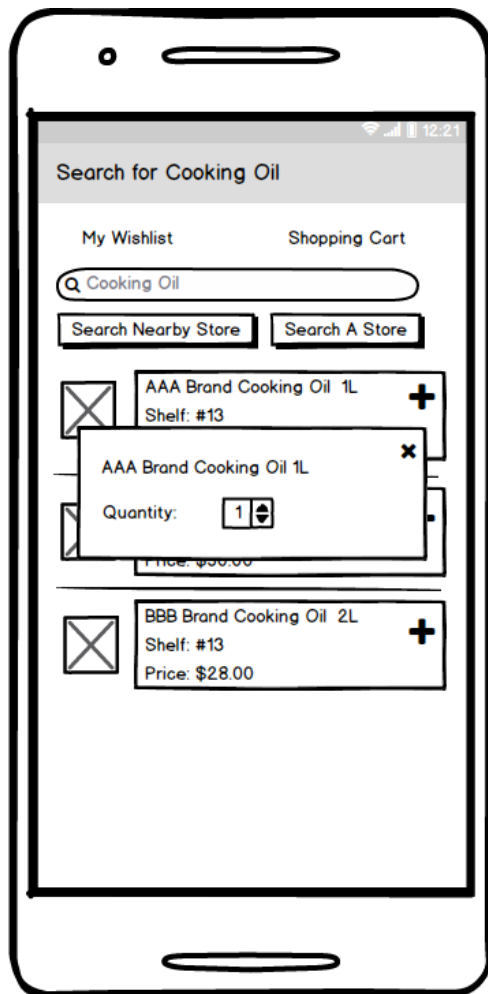
This screen is shown “Search a store” (Screen 3) is clicked. This will allow user to specify which store they would like to search the item in. The map will show the location of the stores in the selected district, and the list below will show the list of stores, their location and their opening hours. User may select one of the store by clicking on the store in the list or the map to specify the store they would like to search the item in.

Screen 5 - Item search results



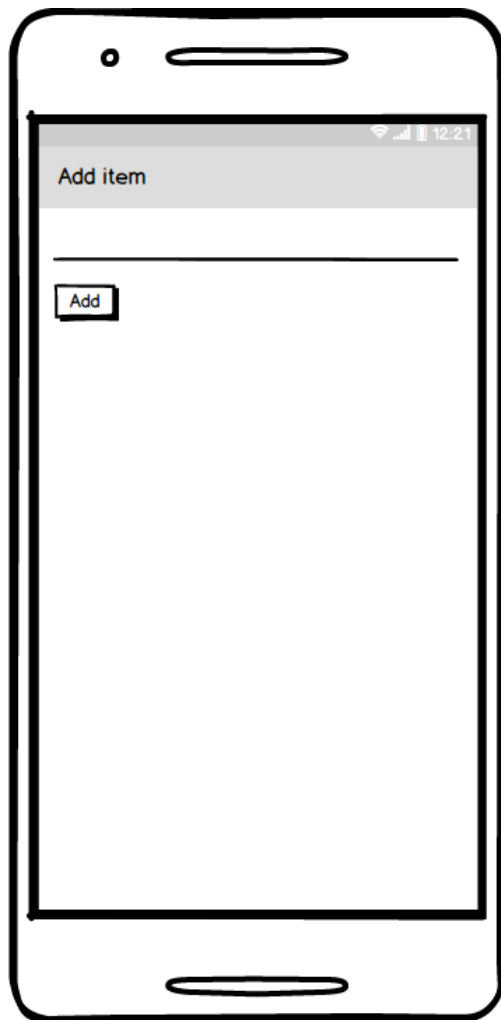
This screen is shown “Search nearby store” (Screen 3) or the store specified in the list in Screen 4 is clicked. This will show the list of items that matches the input, with their details and unit price. User may click “+” to add the item to their shopping cart.

Screen 6 - Item search results - Add quantity to shopping cart



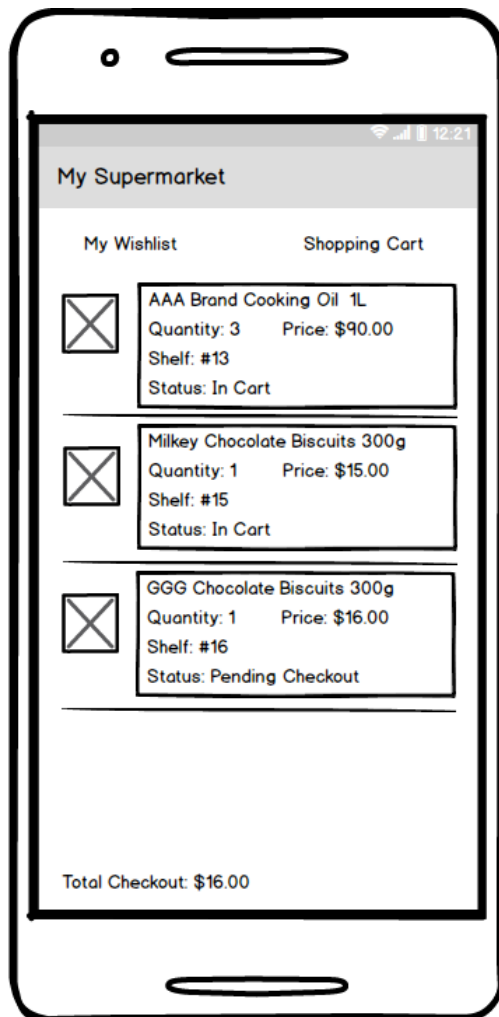
This is shown when a plus icon (“+”) in the list is clicked. User may specify the quantity they want to add to the shopping cart.

Screen 7 - Add item to wish list



This screen will be shown when the “Add item” in Screen 1 is clicked. User may input item name and click “Add” to add the item to the list. When the “Add” button is clicked, the item is added to the list and Screen 1 will be displayed.

Screen 8 - Shopping Cart



This screen will be shown when the “Shopping Cart” tab in Screen 1 is selected. When the item is picked up by the customer in the store, the status will be updated as “Pending checkout”. The “Total checkout” shows the total price that has the status “Pending checkout”. User may also swipe left to right to remove the item from the cart.

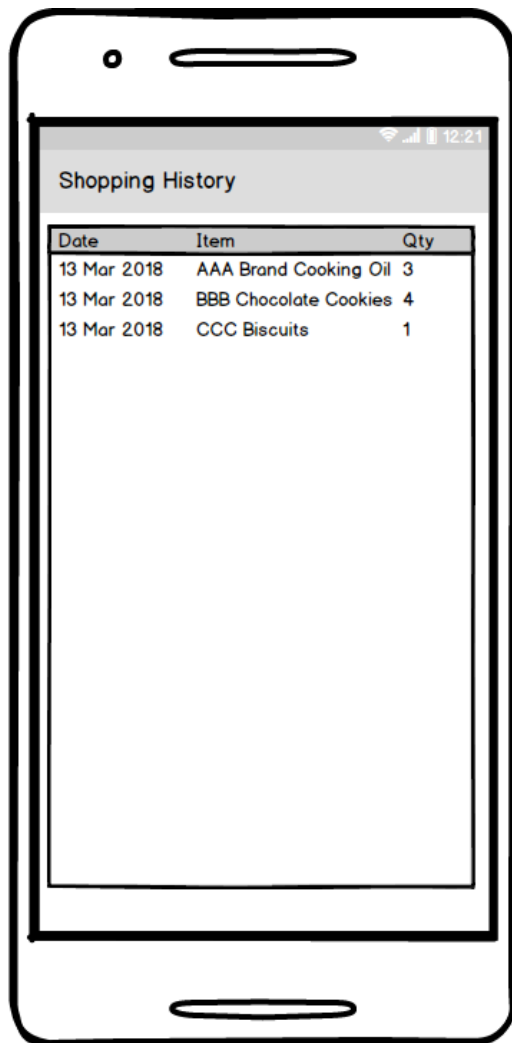
Screen 9 - Checkout summary

Checkout summary -

- 1. GGG Chocolate Biscuits 300g (Qty: 1) - \$16.00
- 2. BBB Cooking Oil 250ml (Qty: 1) - \$15.00
- Total checkout price: \$31.00

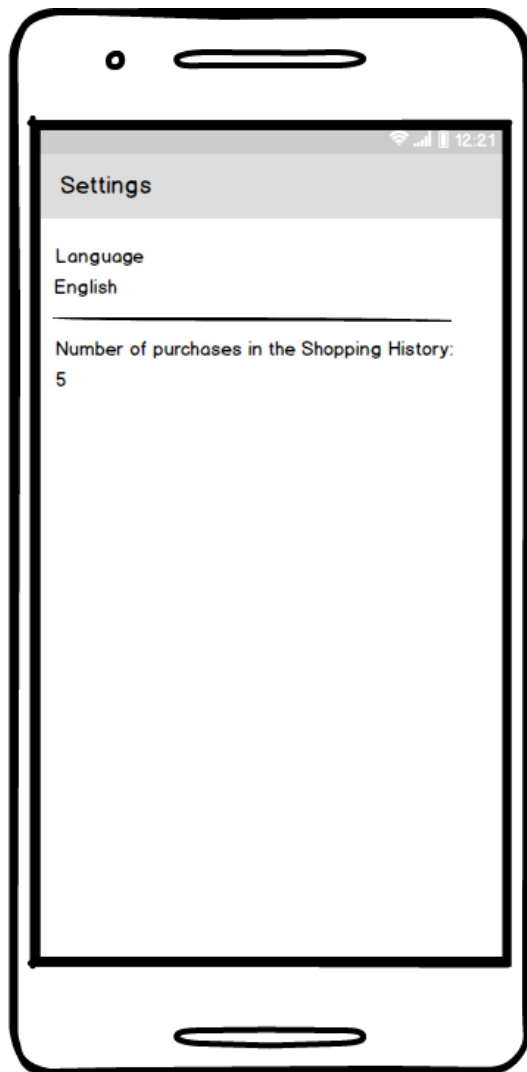
This notification will be shown when the customer leaves the store. This will show the list of items that has the “Pending checkout” status. The “Shopping Cart” will also update the status of the checked out items to “Checked out”.

Screen 10 - Shopping History



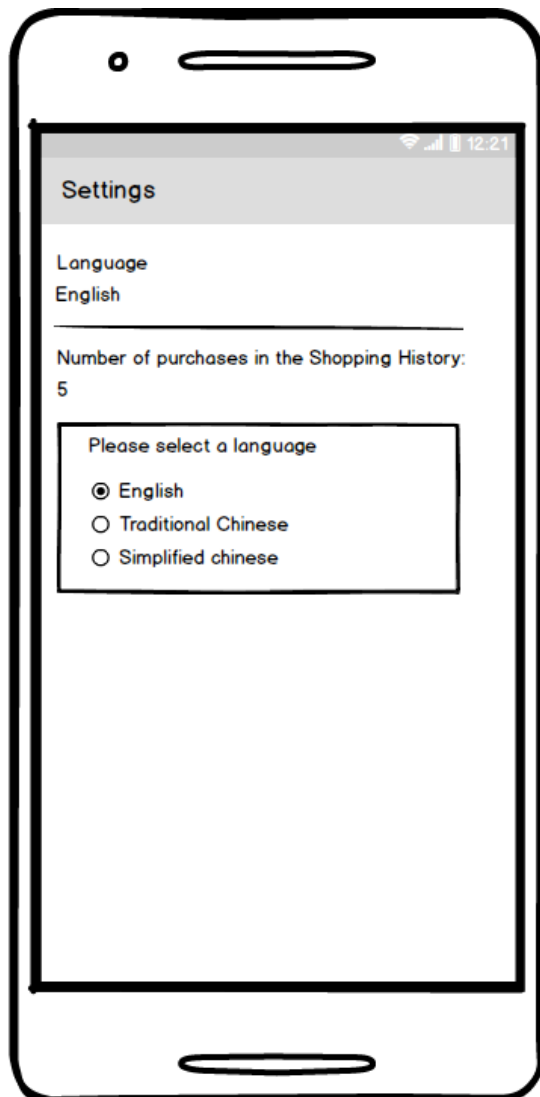
This screen will be shown when the “Shopping History” is selected in the menu in Screen 2. This will list the recent purchases. The number of purchases listed is configurable.

Screen 11 - Settings



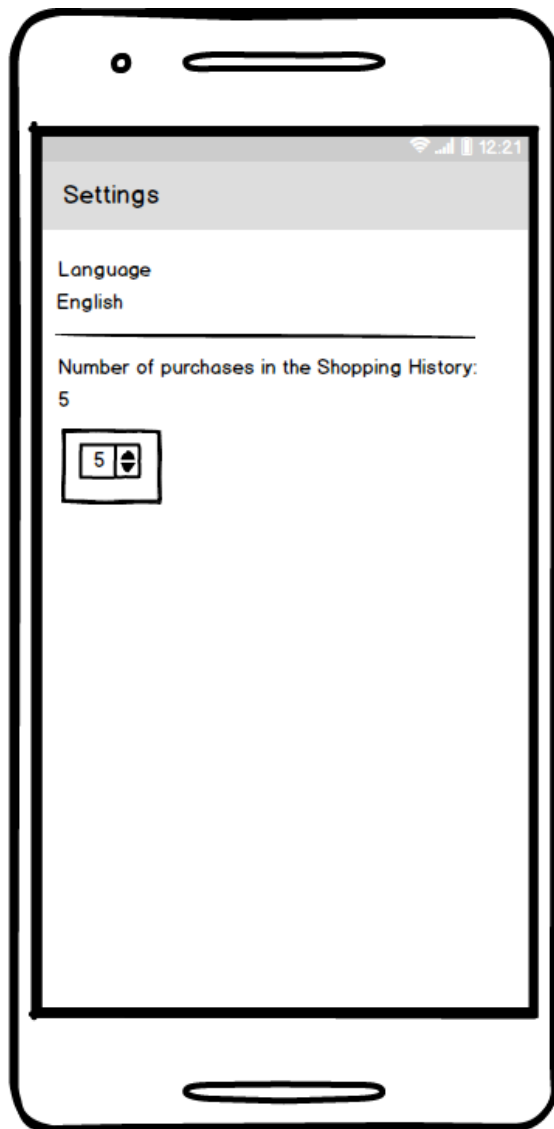
This screen will be shown when “Settings” is selected in the menu in Screen 2. This allow users to change default language of the app and the number of purchases in the shopping history page.

Screen 12 - Settings (Change language)



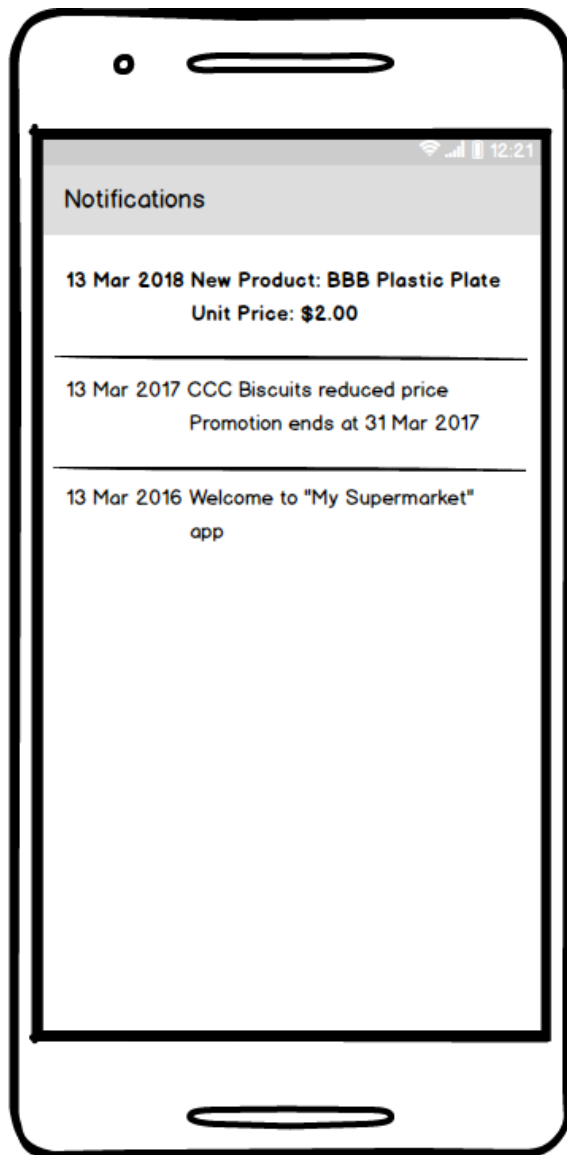
This is shown when the Language is selected. This allows user to specify the language they would like to display in the app.

Screen 13 - Settings (Change number of purchases in the Shopping History)



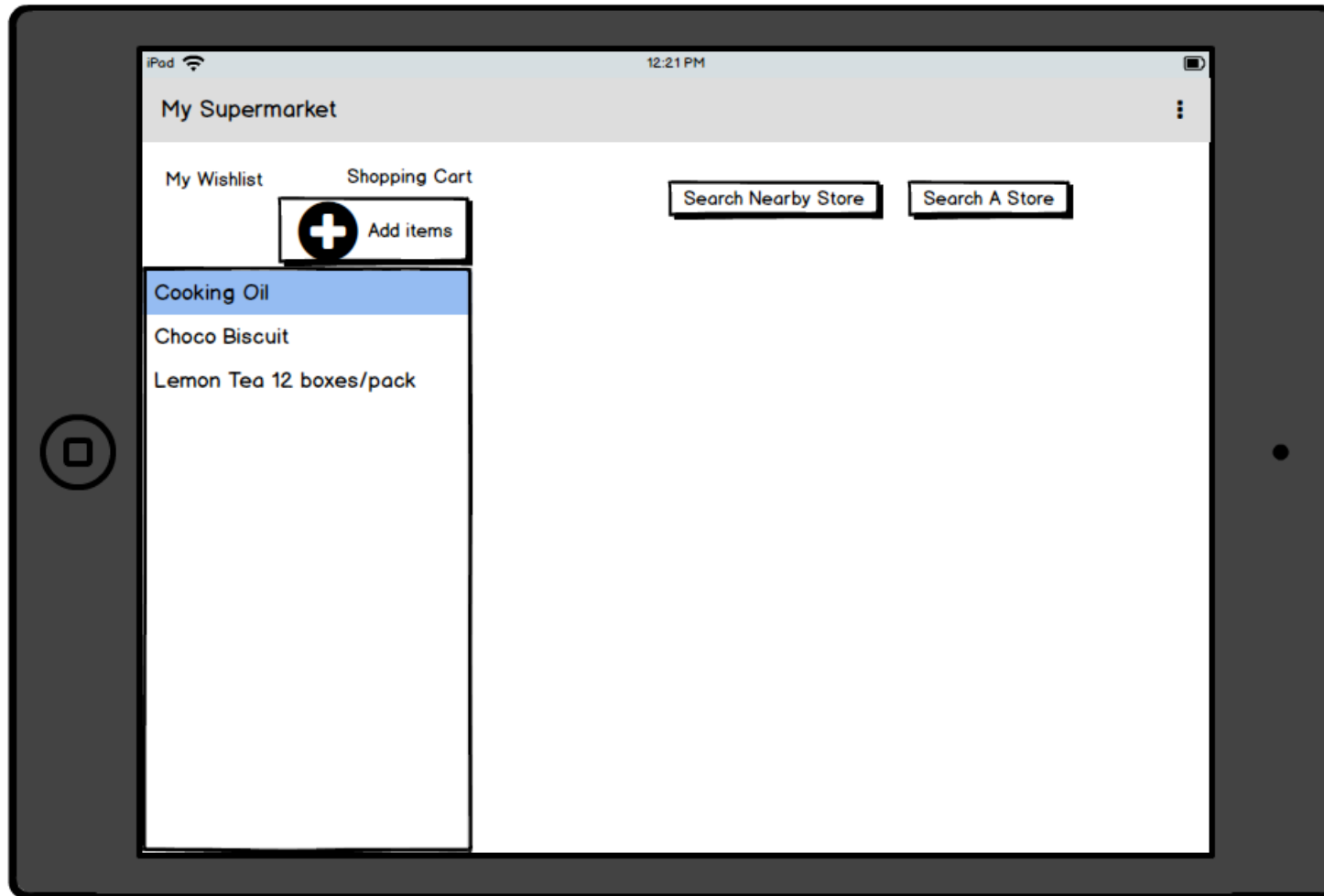
This will be shown when the number of purchases in the shopping history is clicked. This allow users to specify the number of purchases in the shopping history page is displayed. They input values from 1 to 30.

Screen 14 - Notifications



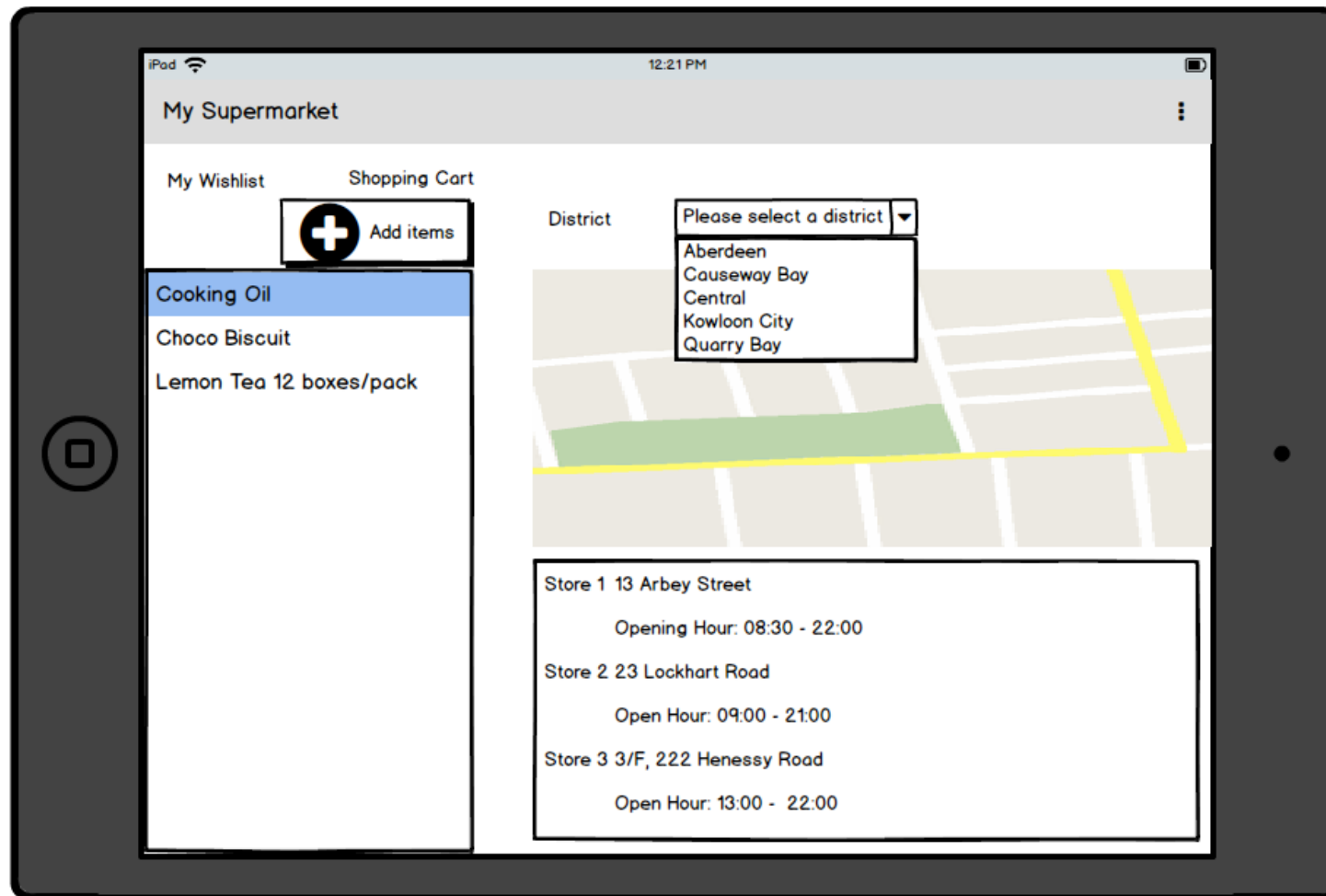
This screen will be shown when the “Notifications” is displayed in the menu in Screen 2. This will display all push notifications, including promotions and new products. The checkout summary notification will not be included.

Screen 15 - Search items (Tablet)



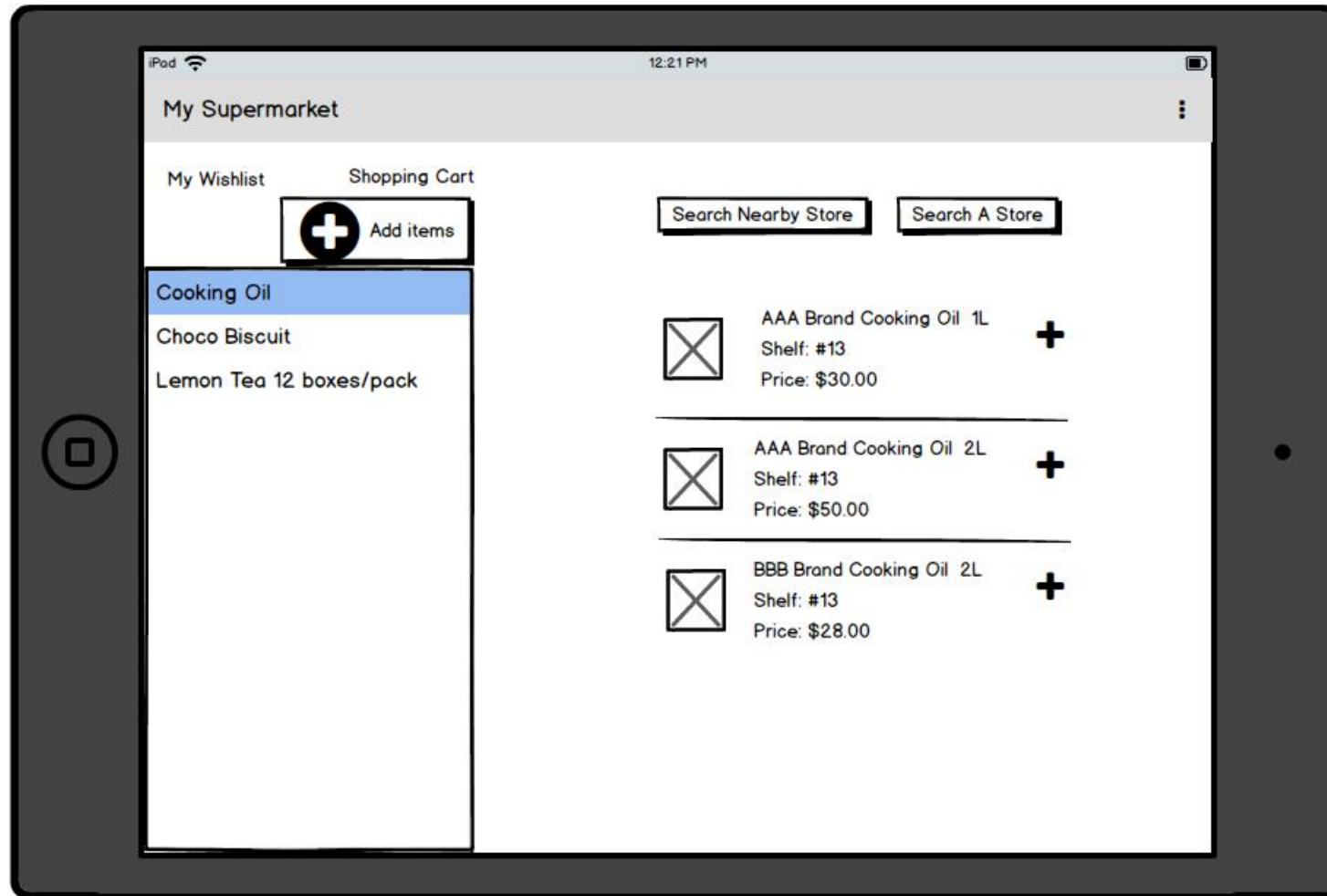
This is shown when an item is selected in the wish list for tablet. The right pane is the same as screen 3 for phone without Search box.

Screen 16 - Search store (Tablet)



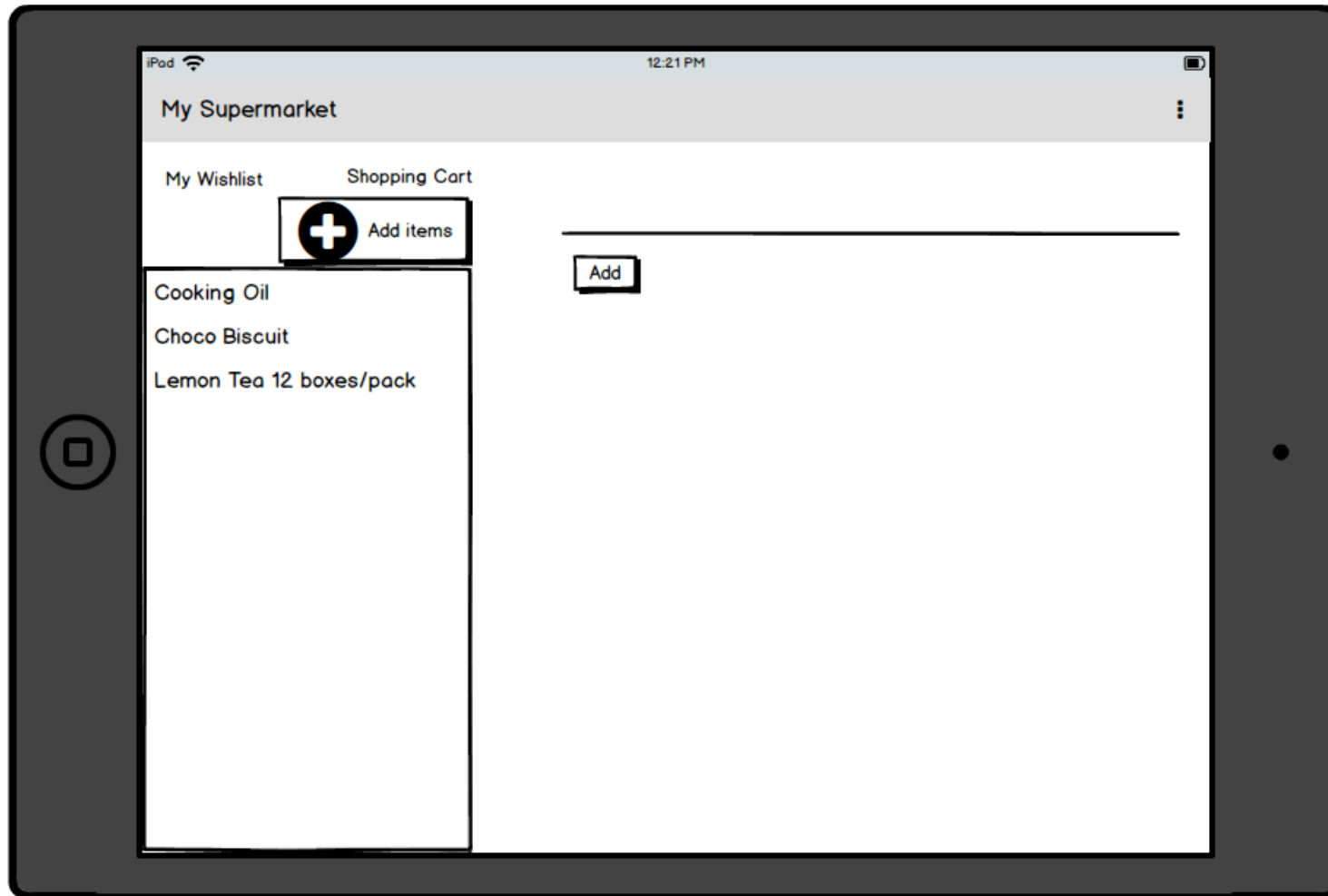
This is shown "Search A Store" button is clicked in Screen 15. The right pane is the same as Screen 4 for phone.

Screen 17 - Search results (Tablet)



This is shown "Search nearby store" button is clicked in Screen 15 or a store in screen 16 is specified. The right pane is similar to Screen 5 for phone.

Screen 18 - Add wishlist item (Tablet)



This is shown “Add item” button is clicked in Screen 15. The right pane is similar to Screen 7 for phone. When “Add” button is clicked, the list at the left pane will be updated.

Key Considerations

How will your app handle data persistence?

Our app will use Firebase Realtime Database to store the store locations and store items. The wish list, shopping cart and shopping history will be stored in Room database. For the setting preferences, we will use Shared Preferences.

Describe any edge or corner cases in the UX.

Corner cases / edge cases include:

- The user may not have internet connectivity. In this case, we will just display some information message in case there is no data that are stored locally. Otherwise, we will retrieve it from the cache / local storage.
- The user may not have turned on their GPS. In this case, the “Search nearby store” button will be disabled.

Describe any libraries you’ll be using and share your reasoning for including them.

We will be using

- the Picasso library for displaying images because this is a powerful image downloading and caching library for Android, which we have used it for other projects for this course;
- Room library for local data persistence
- Firebase libraries for integrating Firebase services with the app.
- Project Lombok for reducing boilerplate code
- Butter Knife library for reducing boilerplate code

Describe how you will implement Google Play Services or other external services.

We will implement our application using the following Google Play Services:

- Google Maps API for displaying maps and location
- Google Distance Matrix API for calculating the distance from user’s position to the nearby store.
- Geolocation API for locating user’s position.
- Firebase Realtime Database for storing items and locations
- Firebase Cloud Messaging for promotion notifications, as well as detecting user coming in and leaves store, and the items picked up/put down.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Prepare the data

Prepare the data (e.g. store location, store items and message templates for detecting user coming in and leaves store) in Firebase console.

Task 2: Project Setup in Android Studio

Setup the libraries in build.gradle.

Task 3: Implement the database components

Implement the Room database components to store the wish list, shopping cart and shopping history

Task 4: Implement the UI for Wish list

Implement the UI layout for wish lists, including Add item, list items and delete items

Task 5: Implement the UI for search items in store

Implement the UI layout for Search items, including detecting the location whether a nearby store exists.

Task 6: Implement the UI for search a store

Implement the UI layout for search a store, and the actions for clicking on the item in the search result / map to select the store.

Task 7: Implement the UI for search item results

Implement the UI layout for item search results, and the action to add the item to the shopping cart.

Task 8: Implement the UI for shopping cart

Implement the UI layout for shopping cart

Task 9: Implement the UI for shopping history

Implement the UI layout for shopping history

Task 10: Implement the UI for Settings

Implement the UI layout for Settings screen, and the update for the shared preferences.

Task 11: Implement the UI updates based on the shared preferences

Update the UI layout for language display and the number of records in the Shopping History list based on the selected value in the shared preferences.

Task 12: Implement the UI for notification list

Implement the UI layout for notifications.

Task 13: Implement the service to handle Firebase Cloud Messaging messages

Implement the service class to handle Firebase cloud messaging messages, including the push notifications for promotion, updating the state to indicate customer is entering the store or leaving the store, update the item state in shopping cart when an item is picked up / put down.

Task 14: Implement the UI layout for Checkout summary

Implement the notification when the items are checked out.