Mobile Application Development

62415

Joshua Rudaitis s182783

Supervisor: Prof. Jacob Nordfalk

December 6, 2018

Table of Contents

Introduction

The primary purpose of the software presented in this report is to facilitate the process of creating, managing, utilizing and sharing a shopping list. These four core "pillars" are the foundation of the user experience and the main issues which the app aims to solve. This report will focus on the development process and features found in "ReList" *(Figure 1)*.

The goal of the app is to make using a shopping list as easy and natural as possible, while adding useful features not found in the Google Assistant list. The flagship features needed to make this possible include:

- Make a Shopping List (swipe or tap to remove items)
- Use voice commands to intuitively add items
- Store a 'default' list, with the user's favorite items which can propagate an empty list quickly and easily
- Save and get directions to the user's favorite stores
- Share List with friends and family
- Switch between a Light and Dark theme

By combining said features into one program, the user will be able to more easily manage their lists while having useful tools a tap away.

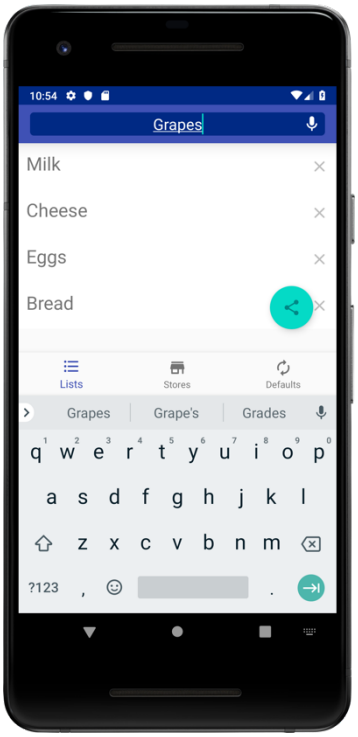


Figure 1: ReList App

Key Features

Shopping List

1. Must have an easily accessibly text box from which one can enter items
2. Needs to have natural voice input to maintain feature parity with Google Assistant
3. Individual Items can be removed by swiping left or right, or by tapping an ‘X’
4. Should be easily shareable with user contacts
5. User should be prompted to load a default list saved in memory in the empty case

Stores

1. Must have a list of selectable locations which user has saved
2. When a location is selected, view should zoom over to new location
3. Any selected location has to have a marker
4. Text address needs to be translated into lat/long coordinates
5. User should be able to get directions to location via external map *(reduce permissions)*
6. Change ‘Theme’ of map based on app settings *(Light/Dark Theme)*

Defaults

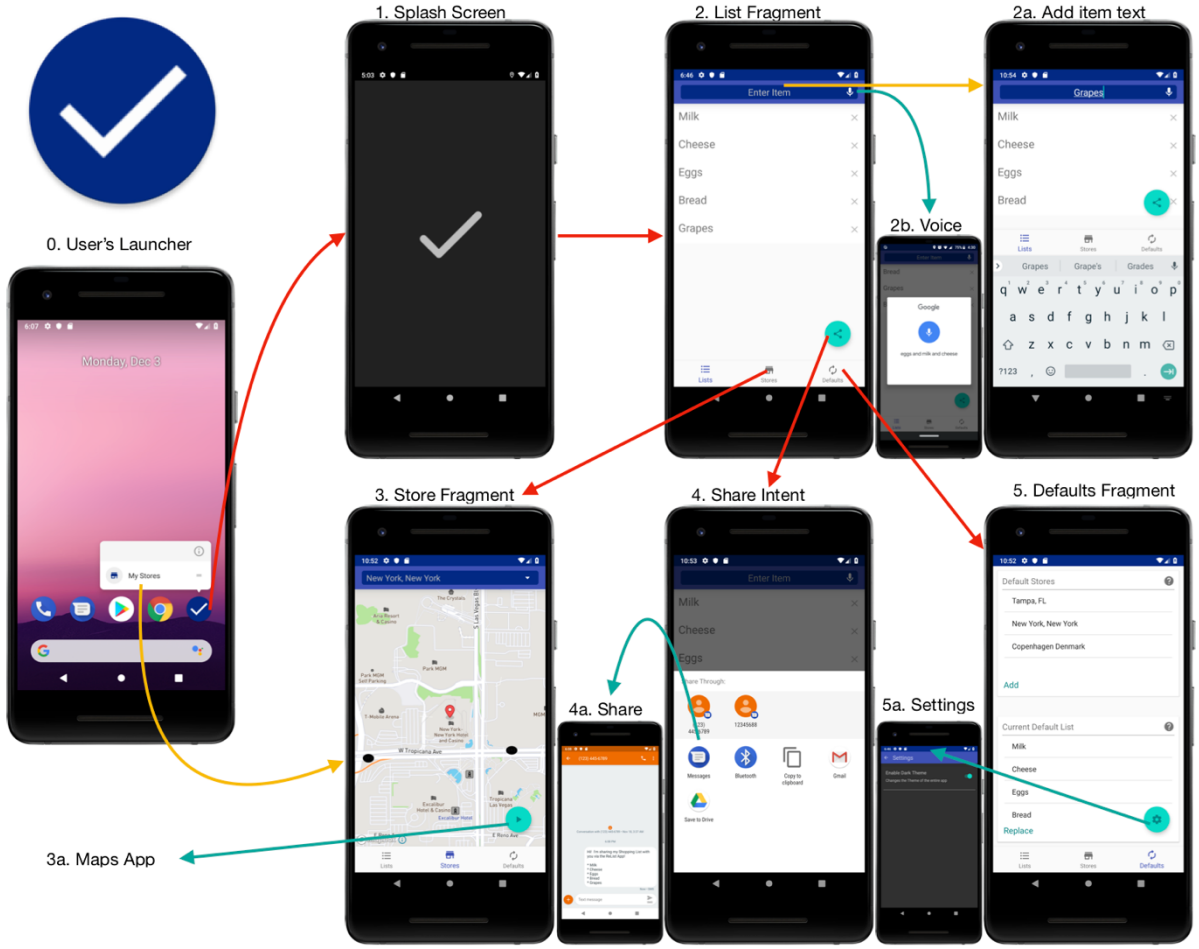
1. Must be able to add/remove to list of user’s favorite stores for use in **Stores** functionality
2. Must be able to update the current default list for use in the **List** functionality
3. Must be able to access app settings
4. App settings must have Light/Dark ‘Theme’ toggle

User Interface / Experience

1. Reduce the amount of permissions needed *(use Google App for voice input, use external map for directions, etc.)*
2. Include a ‘Theme’ toggle in settings for user to decide between Light and Dark mode
3. Follow Material Design as much as possible for UI continuity between apps

Design

Below you can see a flow chart depicting how a user will interact with the program. The user begins their journey at stage 0 (the launcher), where they can either open the app directly or long press the icon and access a shortcut into the store fragment (3). The user will see a splash screen (1) while the contents of the list fragment (2) are loaded. From the list fragment, the user can add items to the list via a textbox (2a) or by speaking the items out loud (2b). Additionally, they can share their current list (4) through a share intent and send it to supported applications such as the messaging app (4a). From the list fragment (2) one may also navigate to the store fragment (3) and receive directions to their saved stores (in the top spinner) through an external map program (3a). Again, from the list fragment (2), the user may access the defaults fragment (5) where they are able to update their saved stores and items, as well as access the settings (5a).



List Page

This page shows the user their current shopping list as vertically stacked cards in a recycler view *(Figure 2)*. Items are added either via the text box at the top of the view *(Figure 2)*, or through tapping the mic icons and saying the names out loud. When using one's voice to add items, individual items need to be separated by an 'and'. Additionally, each individual item in the list is dismissible either by a side swipe or by tapping the 'x' on the right-hand side. The shopping list is also shareable through other communication apps by a share intent *(Figure 4)*. There is an omnipresent bottom navigation bar located at the lower part of the view. This allows for navigation to 'Lists', 'Stores' and 'Defaults' pages from any of the three. If there are no items, a button will appear allowing the user to fill it with their saved default *(Figure 3)*.

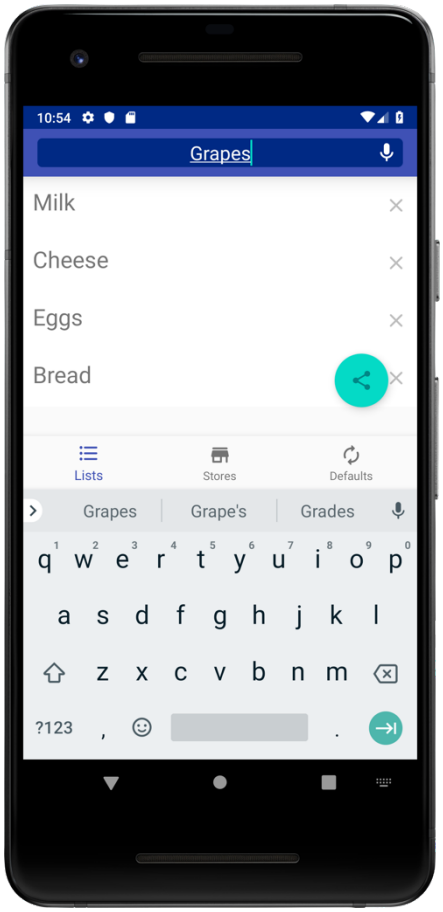


Figure 2: Adding items via text

Figure 3: Zero Case Default Button Appears

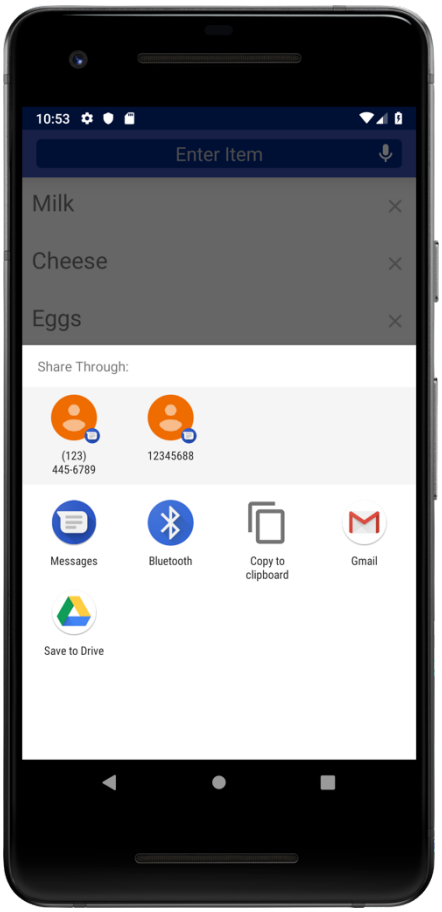


Figure 4: Share Intent

Store Page

This page allows the user to select their favorite stores using the mapbox API. Stores are saved locally and can be selected through the dropdown spinner at the top of the view *(Figure 5)*. The store page can also be loaded directly from the user's launcher by long pressing the icon and selecting the 'My Stores' shortcut *(Figure 6)*. Additionally, upon loading the fragment, the map checks if dark mode is enabled and subsequently uses the corresponding map style *(Figure 7)*. The below code snippet shows how this works. "isDark" is loaded from a SharedPreferences.

```
if (isDark) {

    mapboxMap.setStyleUrl("mapbox://styles/joshrudi/cjohxeb2l2l8e2rpcoexccvwc");
} else {

    mapboxMap.setStyleUrl("mapbox://styles/joshrudi/cjoeyvytq4sfy2qnx2j13xzj6");
}
```

Code 1: Setting Map Theme (mapbox API)

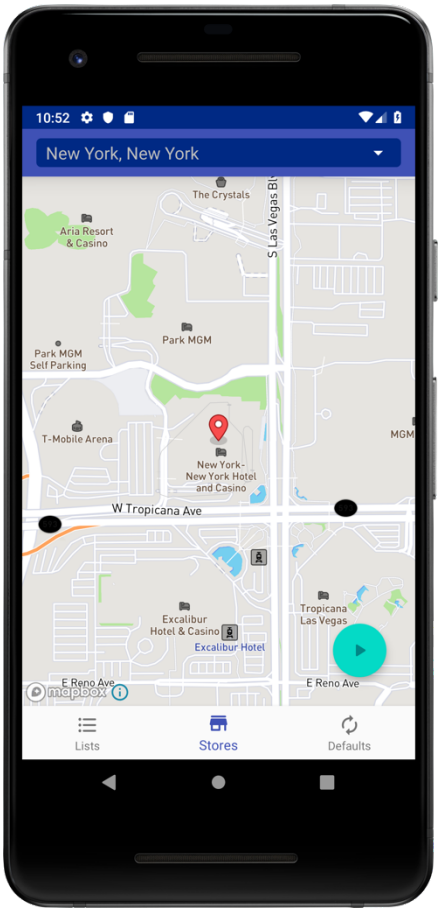


Figure 5: Store Page

Figure 6: User Launcher

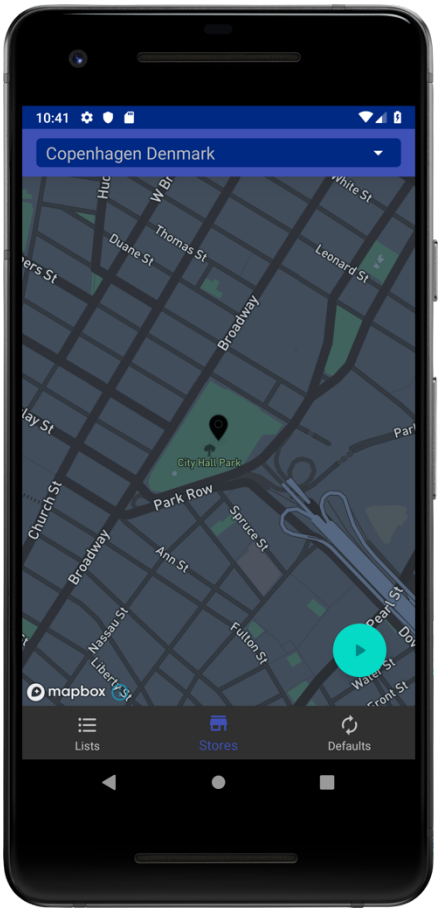


Figure 7: Dark Mode Map

Defaults Page

This page allows the user to add and remove default stores and items *(Figure 8)*. Store addresses can be added by tapping on the 'Add' button on the first card *(Figure 8)*, and then typing in the name/address of the location in the popup action dialog *(Figure 10)*. Store locations can be removed simply by tapping on the name of the store, and then confirming the choice on an action dialog. The default list can be updated with the current list on the 'Lists' page by tapping on the 'Replace' button on the second card *(Figure 8)*. Additionally, the FAB (floating action button) opens the settings page *(Figure 9)*. The settings contain a switch which when toggled writes to a SharedPreferences storing the Boolean theme state *(Code 2)*.

```
AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_YES);
prefs.edit().putBoolean("theme", true).apply();
```

Code 2: Setting Theme to 'Dark'

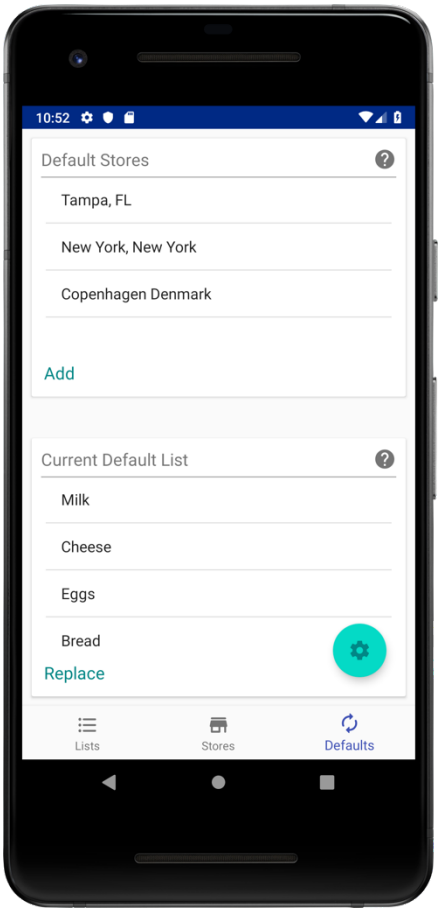


Figure 8: Defaults Page

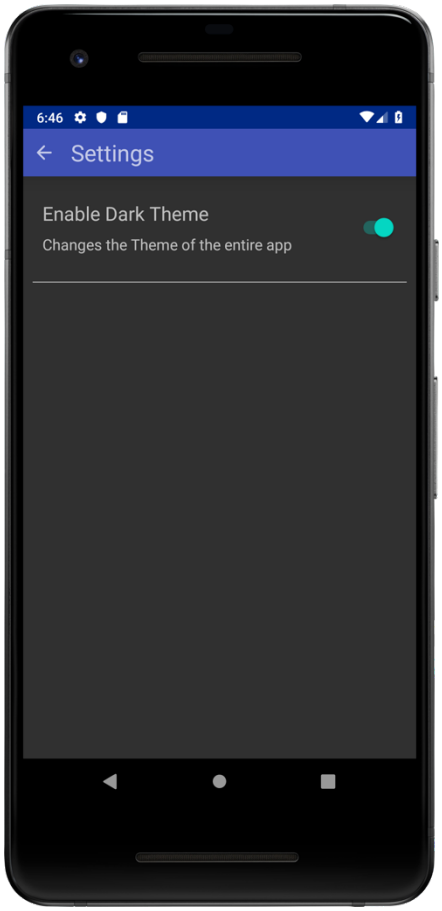


Figure 9: Settings

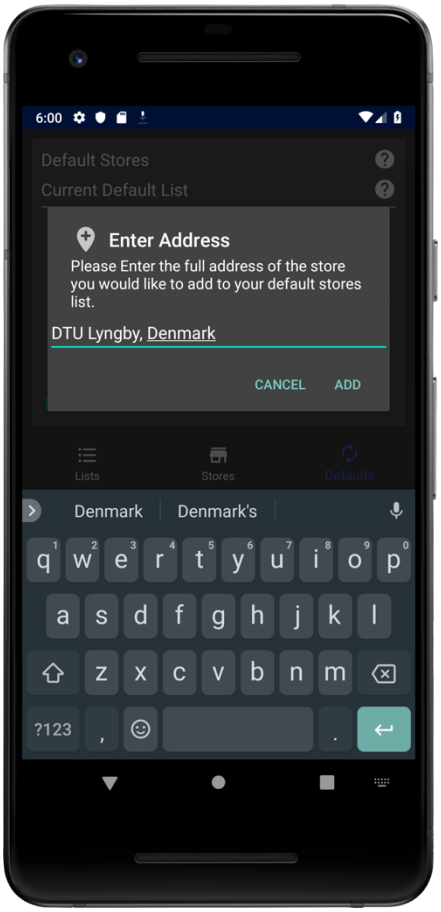
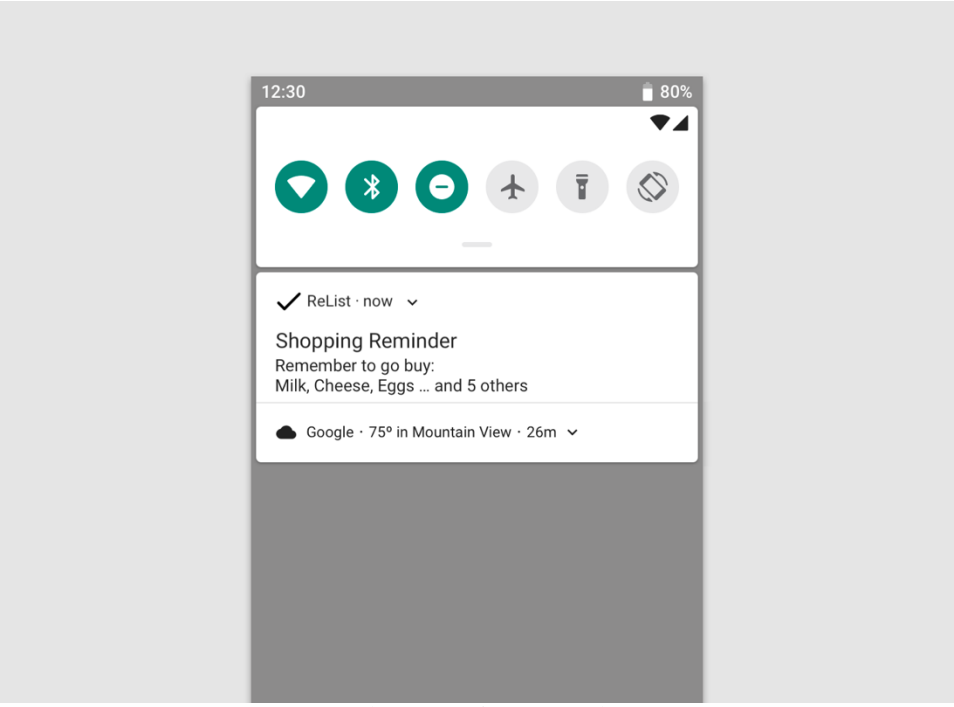


Figure 10: Entering New Store

Conclusion

The goal outlined earlier of this document to make a “shopping list as easy and natural as possible, while adding useful features not found in the Google Assistant list” was accomplished through means of adding beneficial features to the user without sacrificing simplicity and privacy. All data is stored locally, and all pages are clearly outlined and easily accessible. If more time were to be put into further developing this project, there are a couple new features that would be a natural evolution of the product. One improvement which would be useful to the end user include the ability to send a generated link through messaging apps, which would direct the recipient to a webpage allowing for the modification of the current user’s list. Another improvement would be giving the user the option to set a reminder to go purchase the items on their list at a specific time. A mockup of this idea can be seen below *(Mockup 1)*.

Taking into consideration the areas of expansion possible, there are still substantial ways for this app to grow and mature provided continued maintenance. While a shopping list is something that might be considered trivial, this project gave the concept a good refresh.



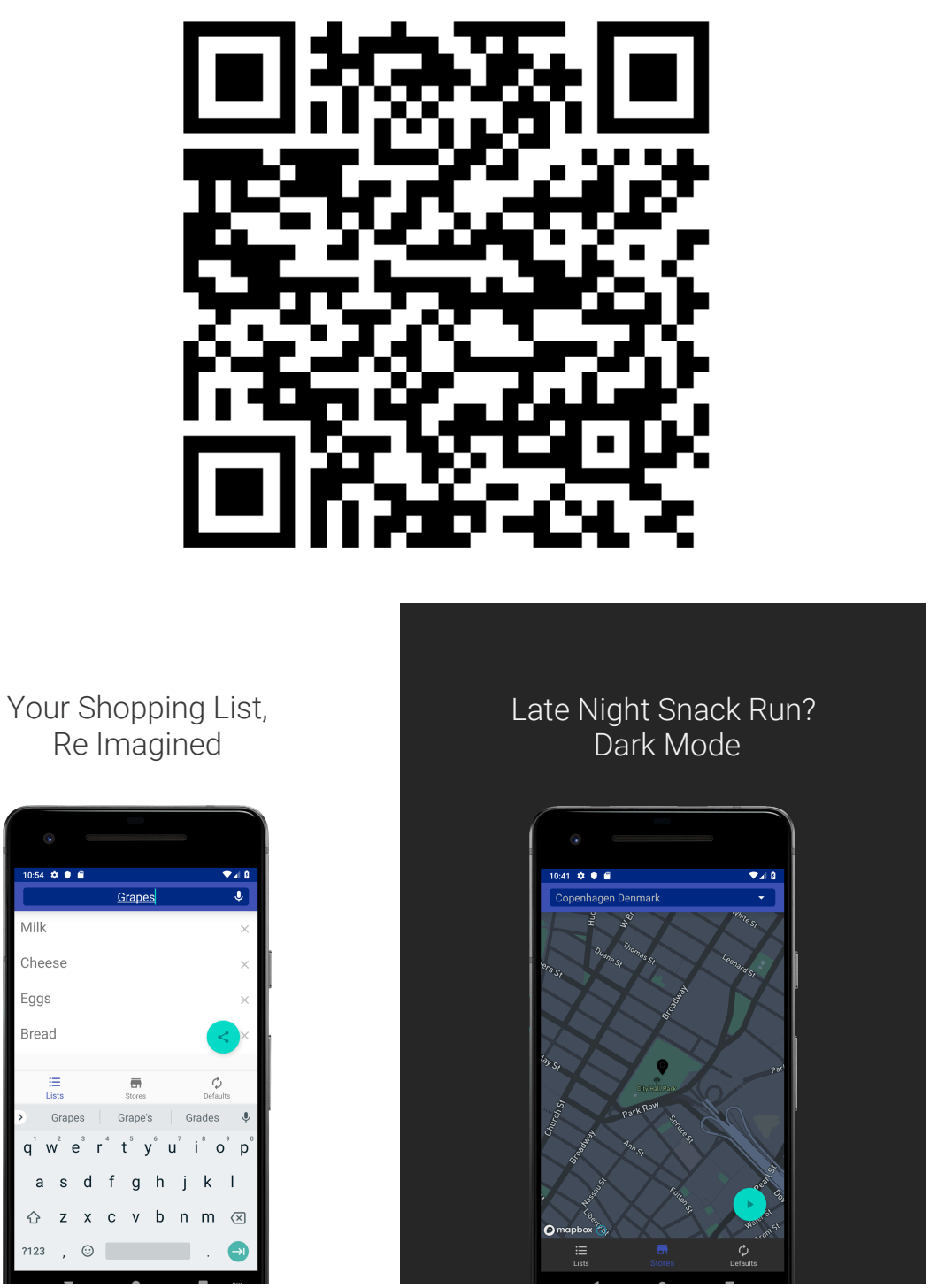
Mockup 1: Notification Mockup

Appendix

The App is downloadable from the Google Play Store from this direct link:

https://play.google.com/store/apps/details?id=com.rudi.soft.relist&hl=en

Or through this QR Code:



Source Code and Resources

The source code for this project is located on the author's (Josh Rudaitis) GitHub page.

Source Code:

https://github.com/joshrudi/ReList

Resources/Additional Software:

Mapbox API: https://www.mapbox.com/android-docs/maps/overview/