Description
Intended User
Features
User Interface Mocks
    Screen 1
    Screen 2
Key Considerations
        How will your app handle data persistence?
        Describe any 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.
Next Steps: Required Tasks
    Task 1: Project Setup
    Task 2: Implement UI for Each Activity and Fragment
    Task 3: Your Next Task
    Task 4: Your Next Task
    Task 5: Your Next Task

**GitHub Username**: nansheng

# Buoy Down

## Description

"Buoy Down" is a Geo app which will revolutionize the easiness of marking down a location because it is integrated with Google Voice action.  Starting of the app by touch or voice will save the geo coordinates into a list and sorted by date time. Or one simple touch of the Floating action button will suffice. Additional note can be added in the detail page of each location. This app also sends notification to Android Auto and Android Wear when a new location is saved.

## Intended User

It is specifically design for people currently on the move from A to B but would like to mark down locations in between without delaying the original trip. For example, hiking in the Summer and stumble upon a specific spot which would be ideal for Fall foliage. Or on the way driving kids to school and found a new Cafe store or spa interested. Or on a business trip and discover a unique place for a romantic getaway instead… etc.

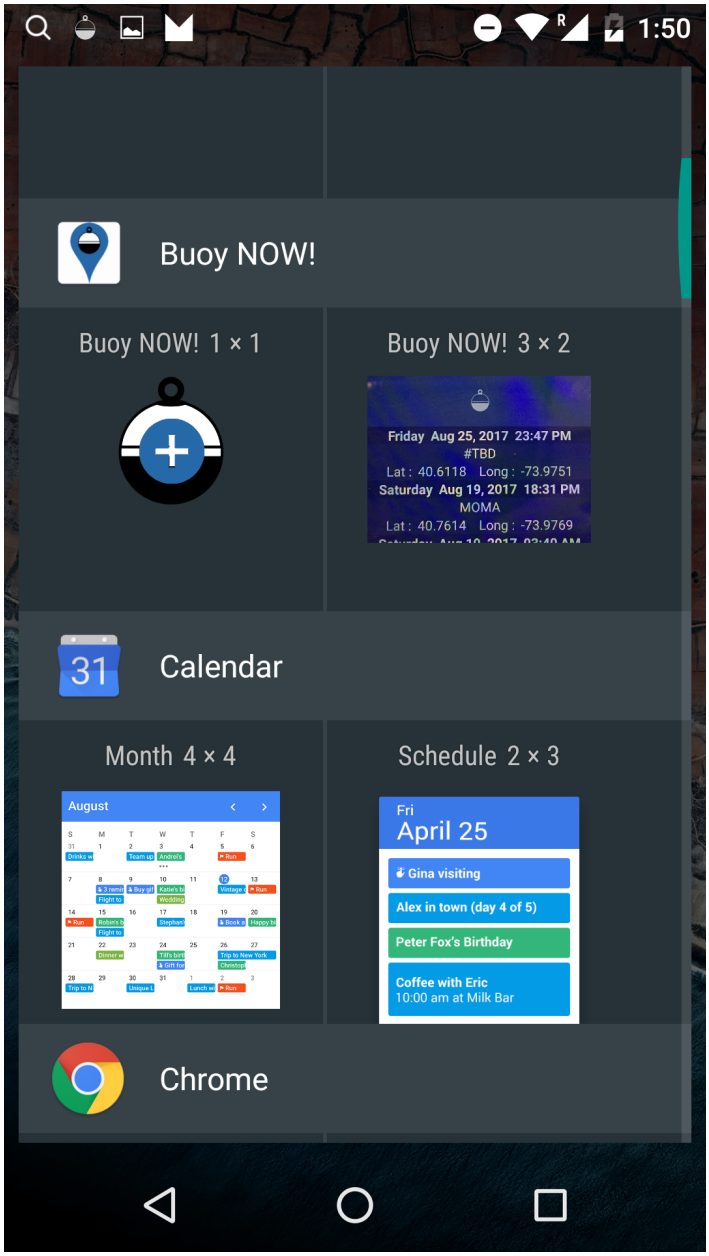
## Features

- Saves Location Data
- Display Saved Location
- Maps Location
- Direction Location

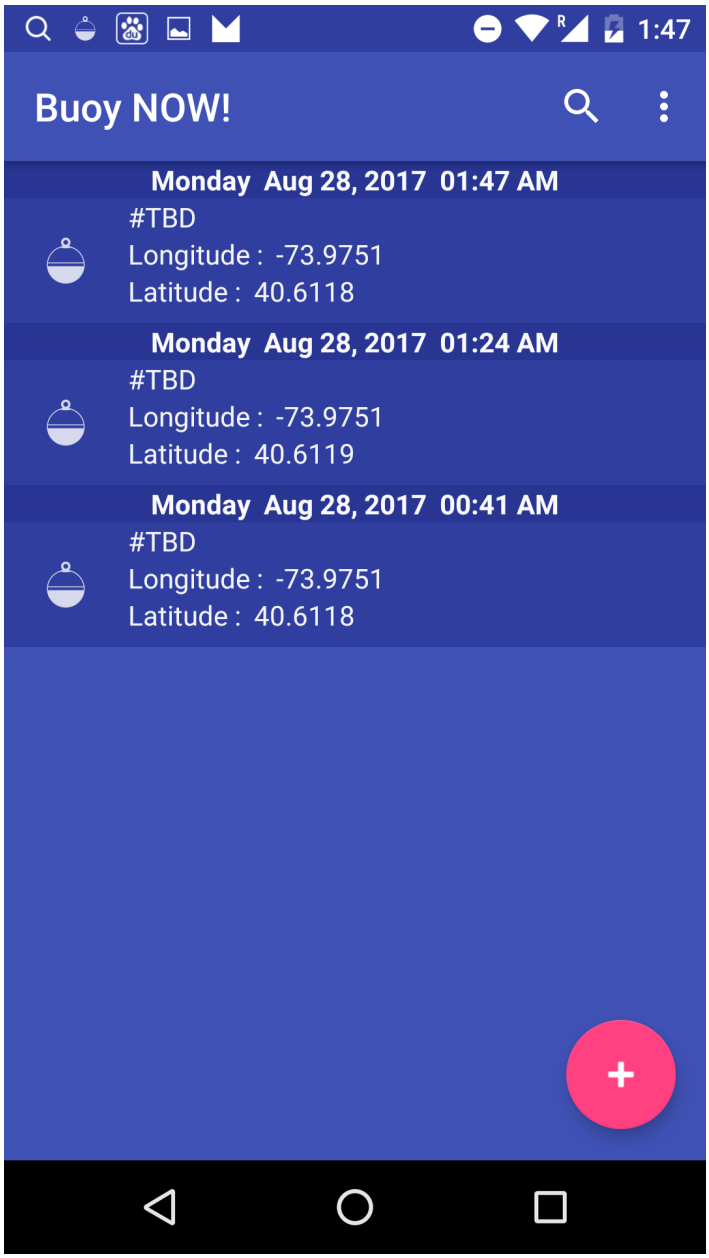
# User Interface Mocks

## Mobile UI Screen with App Icon and Widget

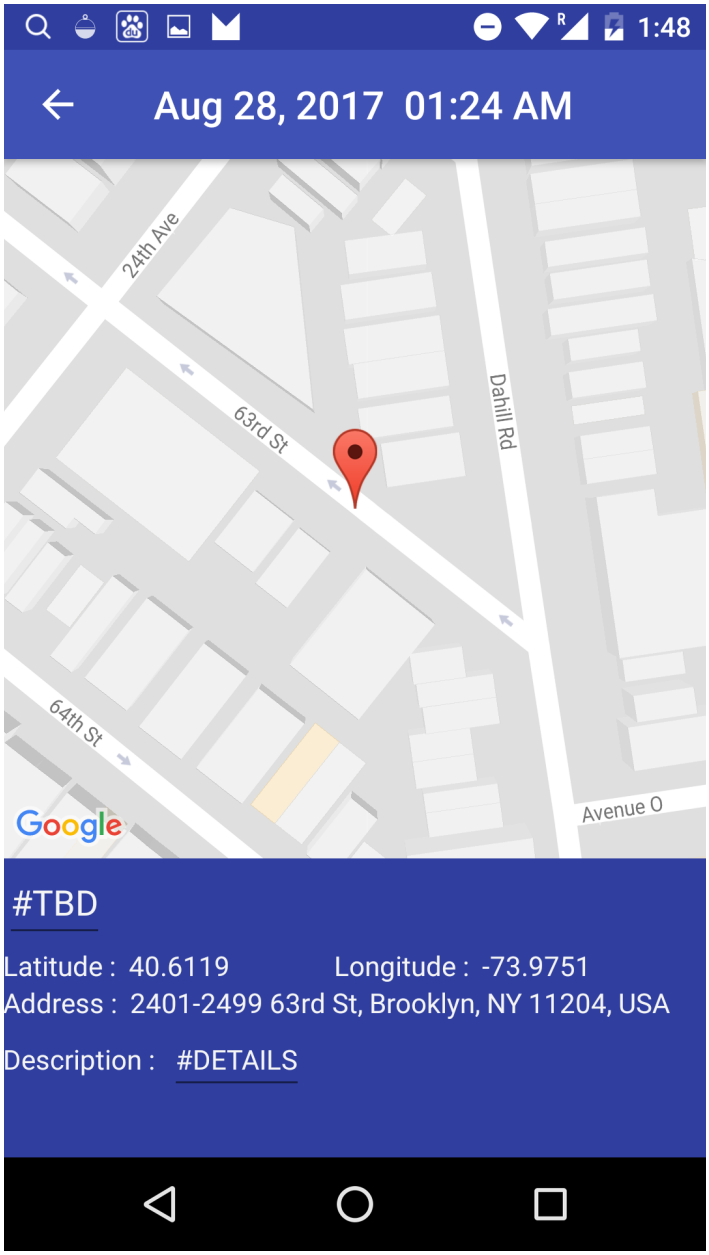
## Mobile Widget Screen



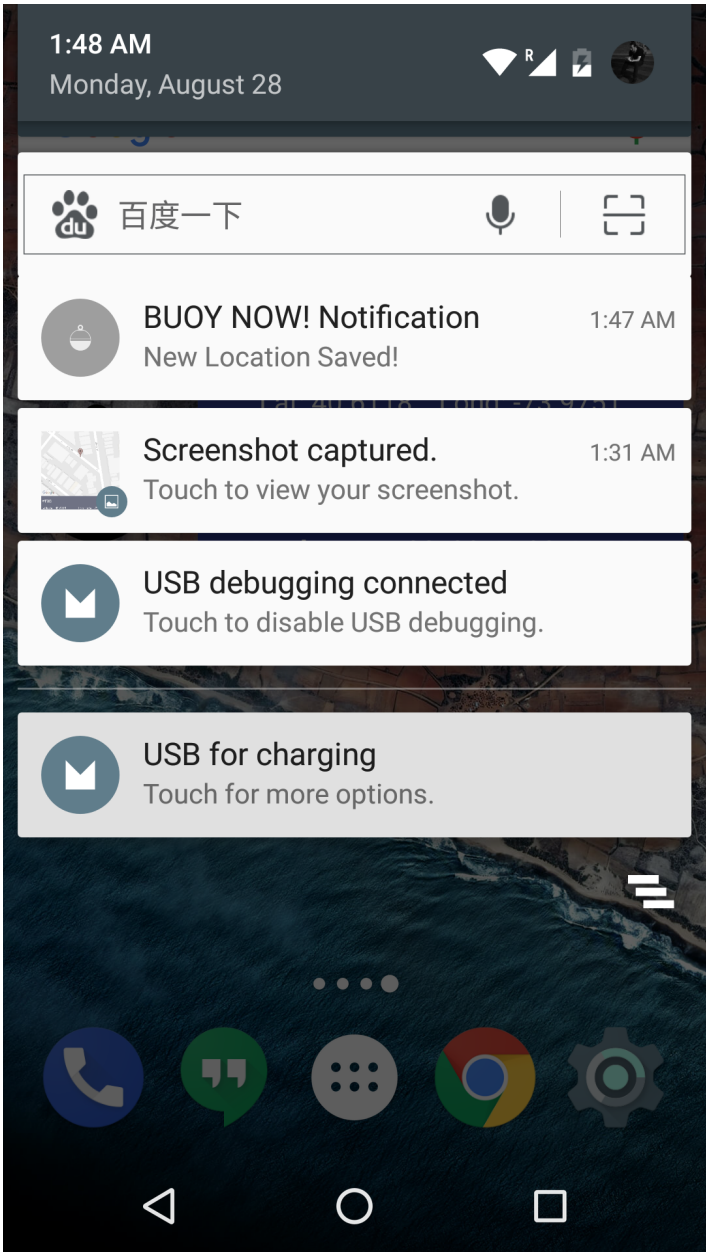
## Mobile Screen 1 (Master Screen)



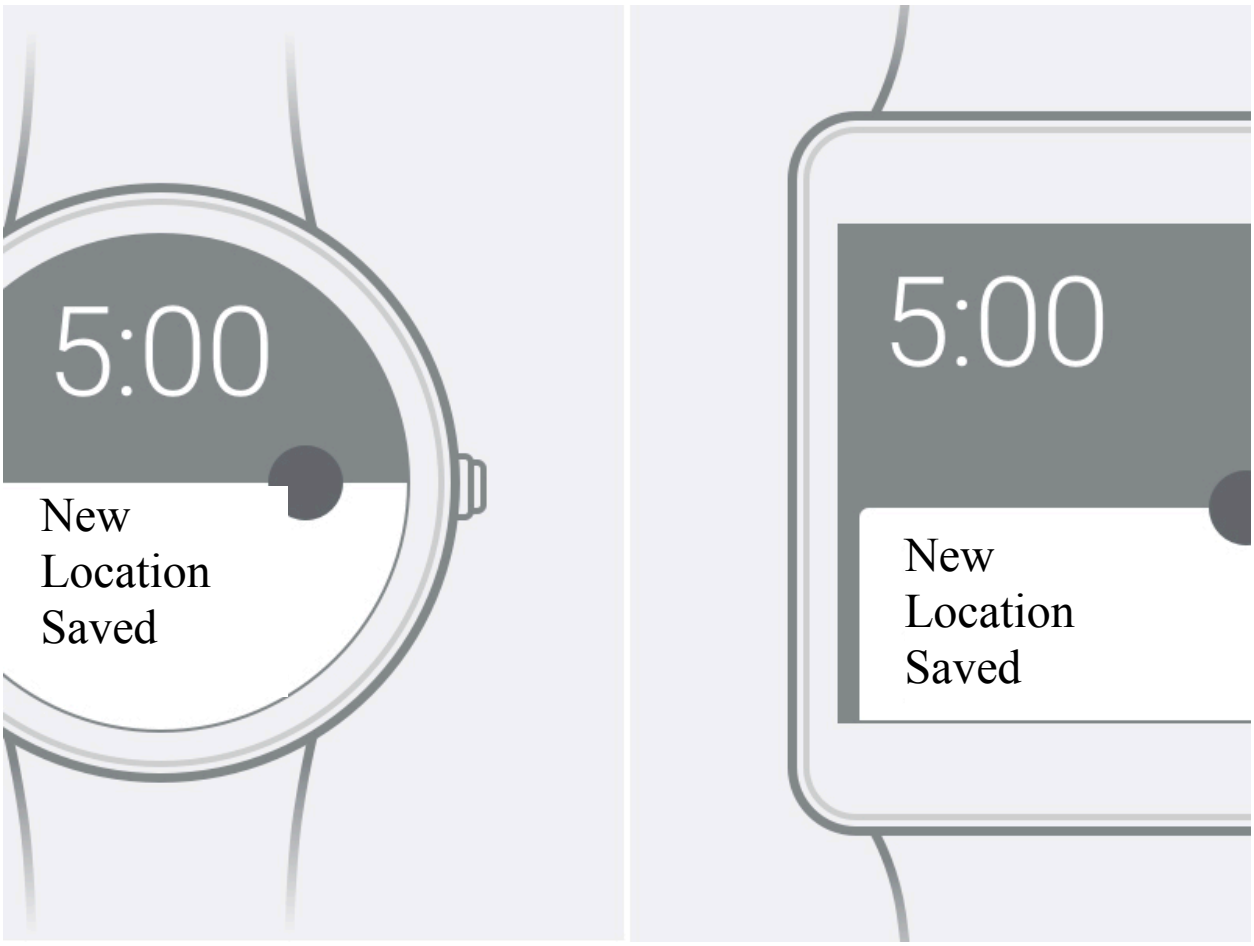
## Mobile Screen 2 (Detail Screen)



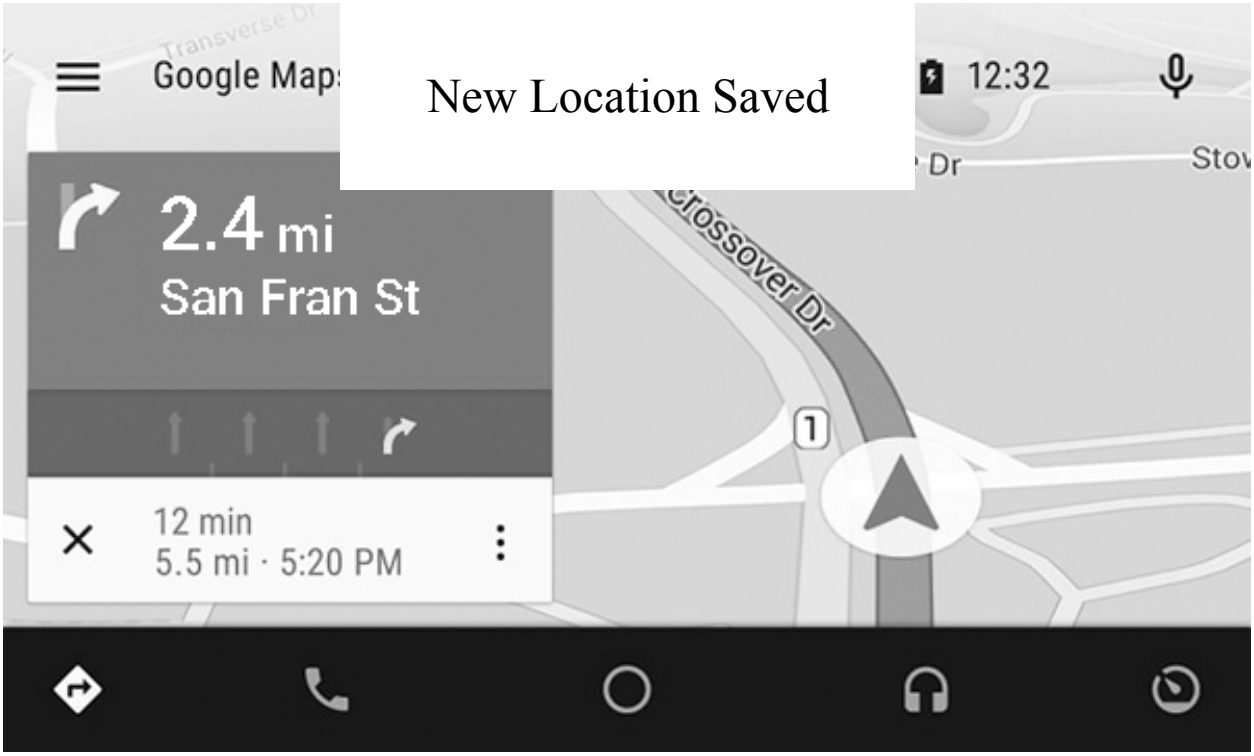
## Phone Notification Screen



**Watch Notification Screen**



**Auto Notification Screen**



# Key Considerations

### How will your app handle data persistence?

Will be building a content provider which will store the Coordination data by date.

### Describe any corner cases in the UX.

A FAB touch will save the current location.

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

Use Picasso library to display thumbnails of each saved location

### Describe how you will implement Google Play Services.

Google Map and Google Places APIs.

# Next Steps: Required Tasks

1. Three parts app for Phone/Tablet.

2. Upon opening the app, by default, it triggers an auto save of current location, the reason for the most efficient and "hand's-off" way of execution. (Option to turn it off in the app settings)

3. An oversize Floating Action Button on all activities/pages for convenient one-push/ one-touch save of location.

4. Previously marked locations will appear on a list sorted by Date or Title in the app settings. By swiping left, the saved location will be discarded.

5. Notification will be send to both android wear and auto every time a new location saves. Notification can be turn off in the apps setting.

6. Tapping on a specific location will invoke the detail page with Google maps showing the path from the current location to the location selected. In addition, a note/description text edit.

7. Two widgets. The Quick Add widget will trigger a new location saved without User Interface at all. The Quick List widget will list the saved location with a more compressed design. Tapping on the Icon will invoke the Main Page. Tapping on the item of the list itself will start from the detail page instead.

## Task 1: Project Setup

- Create a new Android project with Mobile, Wear and Auto.
- Loader to load from database if data pre-existed
- ListView and ListViewAdapter will be utilize

## Task 2: Implement UI for Each Activity and Fragment

- Build Mobile UI with Floating Action Button and Fragment (if needed).

## Task 3: Implement Adding Location

- Using AsyncTask to access Google Location API and Google Awareness API to retrieve current Geo Information including Longitude and Latitude.
- Implement Content Provider so the Geo Information can be saved into database.
- Create a content provider to store location data.
- Using Loader to detect changes and update the list

## Task 4: Implement Detail Page with Map

- Create detail page with the selected Long and Latitude including retrieving the Address
- Using AsyncTask to access Google Maps API to display a map with selected location
- Title and Description will be editable

## Task 5: Implement 2 Widgets

- 1. A 3 x 2 Quick List saved location list
- 2. A 1 x 1 Quick Add Current location save button

## Task 6: Implement Notification for Android Wear

- A notification will be sent to Android Wear when current location save is completed with “New Location Saved!”