How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: pmatushkin

Still In Memphis

Description

Still In Memphis (https://xkcd.com/281/) is an app that utilizes the USPS Web Tools API to allow the users to track their USPS packages. It retrieves, stores, and displays the package statuses, issues the notifications about the latest status changes, and shows the package progress on a map.

Intended User

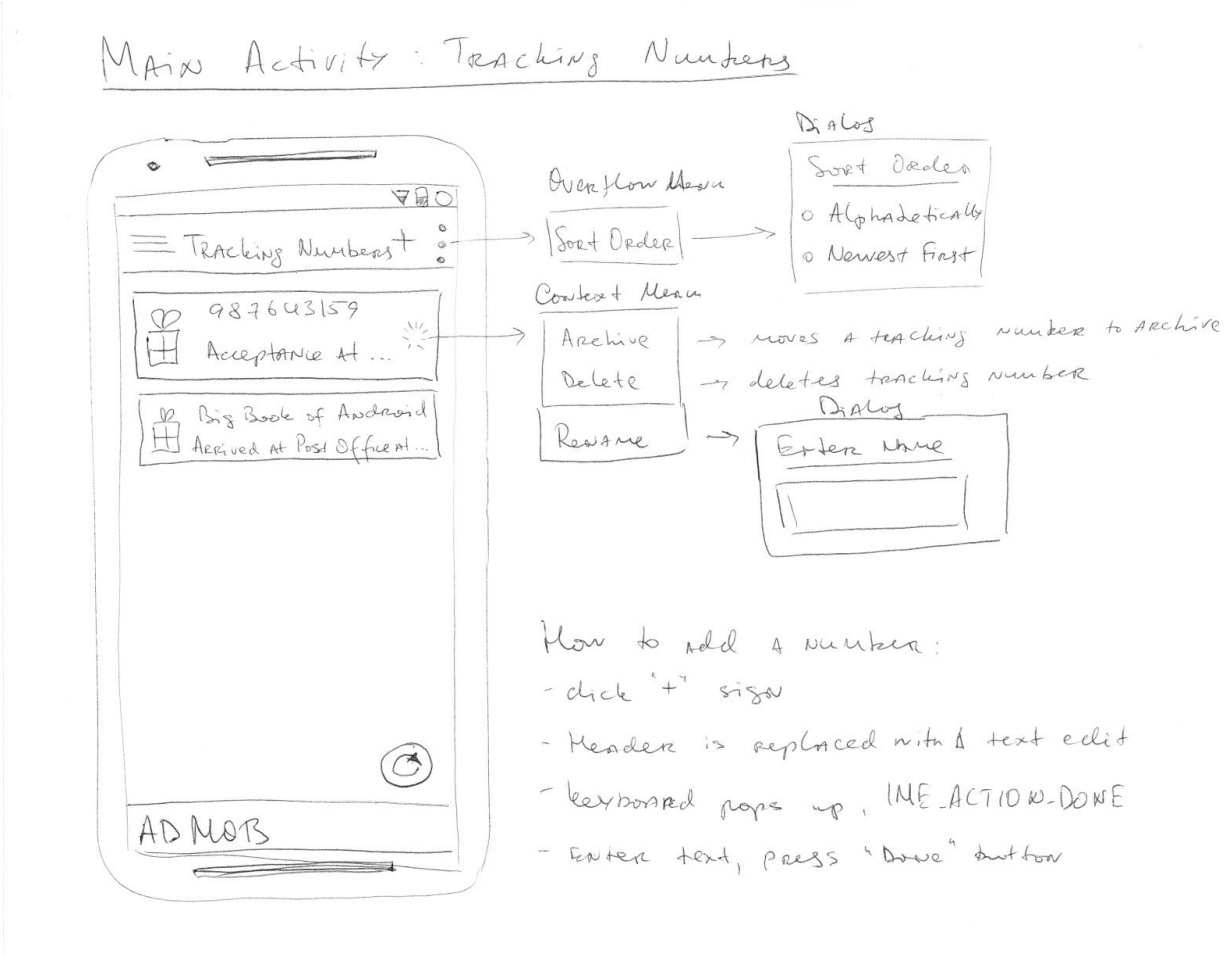
This app is intended for anyone who sends or receives the packages through the USPS and wants to know where they are.

Features

- Accepts the text input or barcode scanning of the package barcode or tracking number.
- Stores the entered tracking numbers along with the tracking history.
- Displays the list of the stored tracking numbers along with the latest status records on the main screen.
- Stores and displays the archived packages until the user deletes them.
- Displays a FAB to request a status update (block the execution of the subsequent requests if one is already in progress, to avoid hammering the USPS web service with the identical requests)
- Displays the list of the package statuses on the detail screen, with the latest status on top.
- Displays the map of the last known location of the package on the top of the detail screen.
- Displays a FAB to share the latest package status.

User Interface Mocks

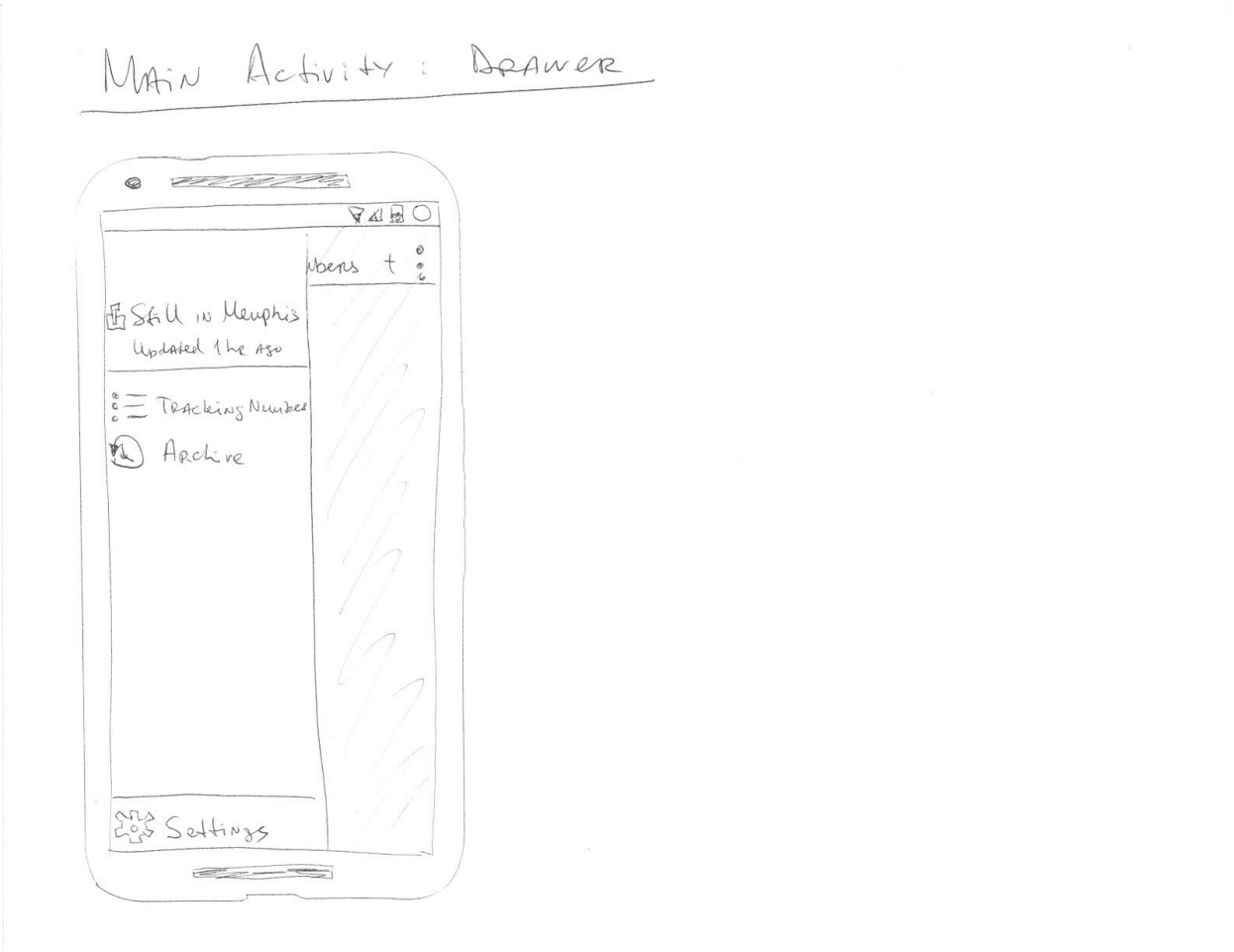
01 Main Activity: Tracking Numbers



This is the main activity of the application.

It displays the most recent statuses of the active (as in “not archived”) tracking numbers.
It provides access to the drawer menu (see below).
It displays “+” button which is used to add the tracking numbers to the application.
It provides access to the overflow menu which displays sorting options for the list of tracking numbers: alphabetically or newest status first.
Each tracking number can summon a context menu which allows the user to archive, delete, or rename the tracking number.
The activity displays a FAB button that triggers the status update.
The activity displays the AdMob fragment.

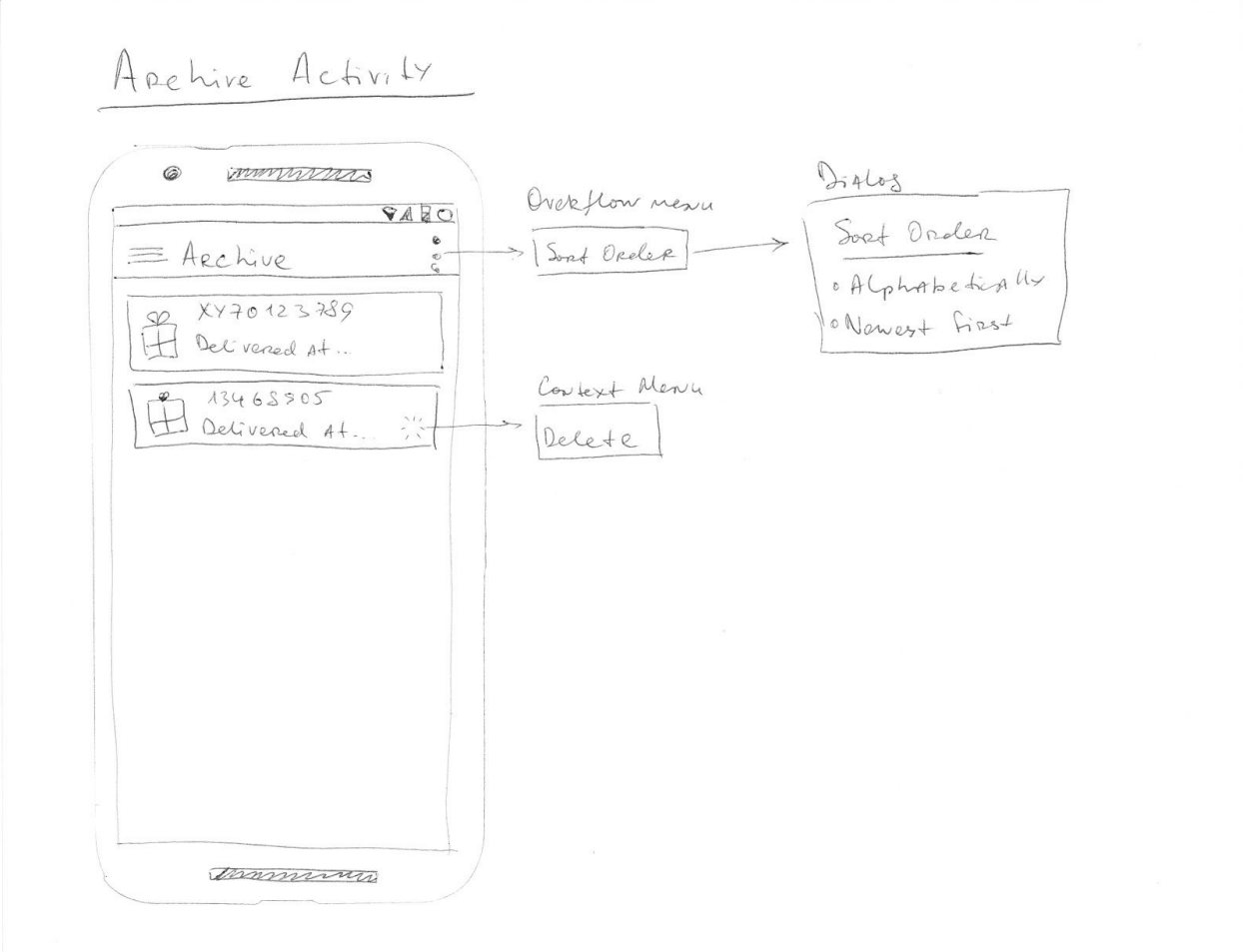
02 Main Activity: Drawer



The drawer displays the following:

- Application name and icon
- The time of a most recent status update
- Two options: Tracking Numbers and Archive to switch between active and archived tracking numbers
- Settings

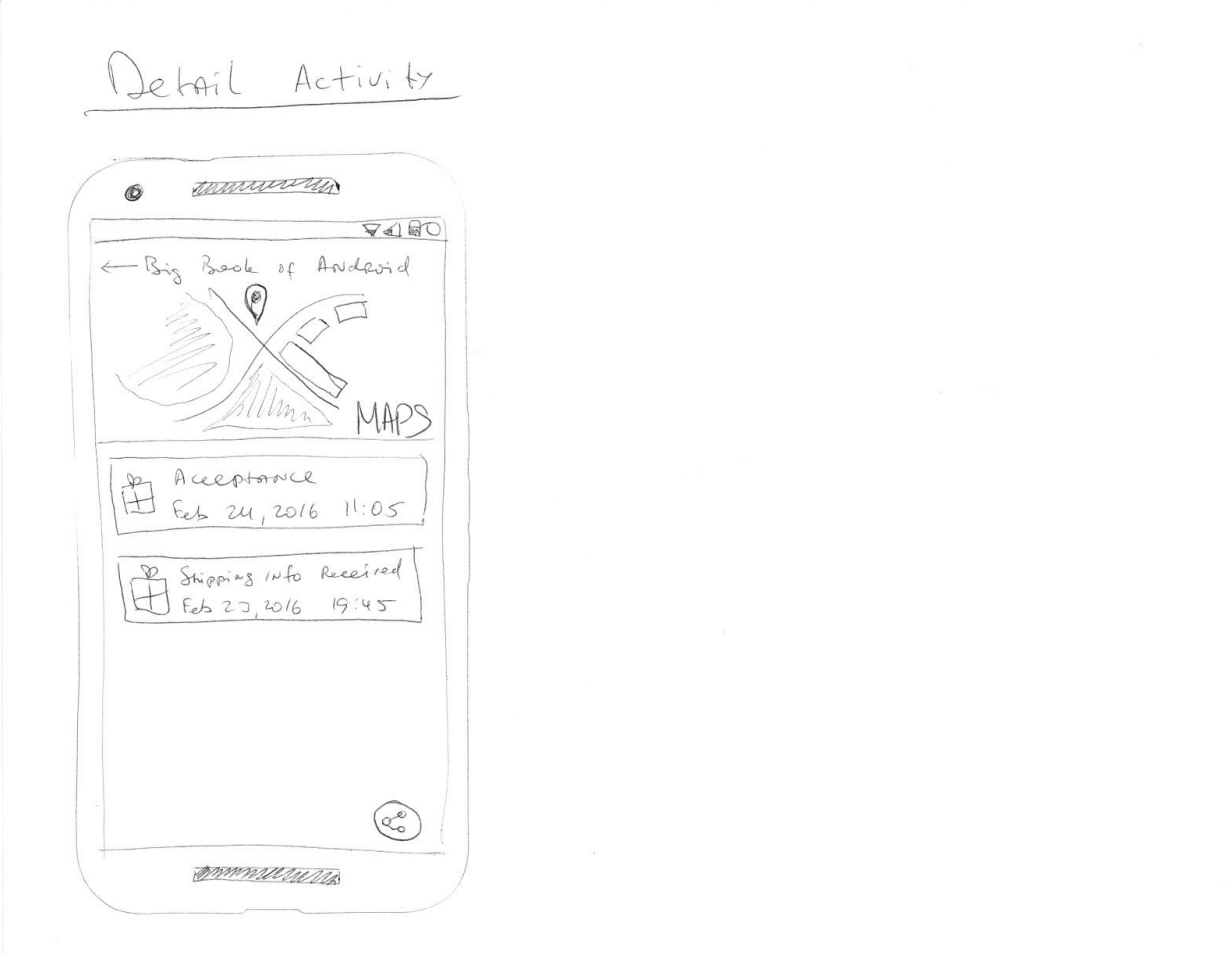
03 Archive Activity



This is the Archive activity. It's very similar to the main activity. I'm actually thinking of reusing the Main activity. The differences are:

- No "+" button
- A context menu has only Delete option
- No FAB button

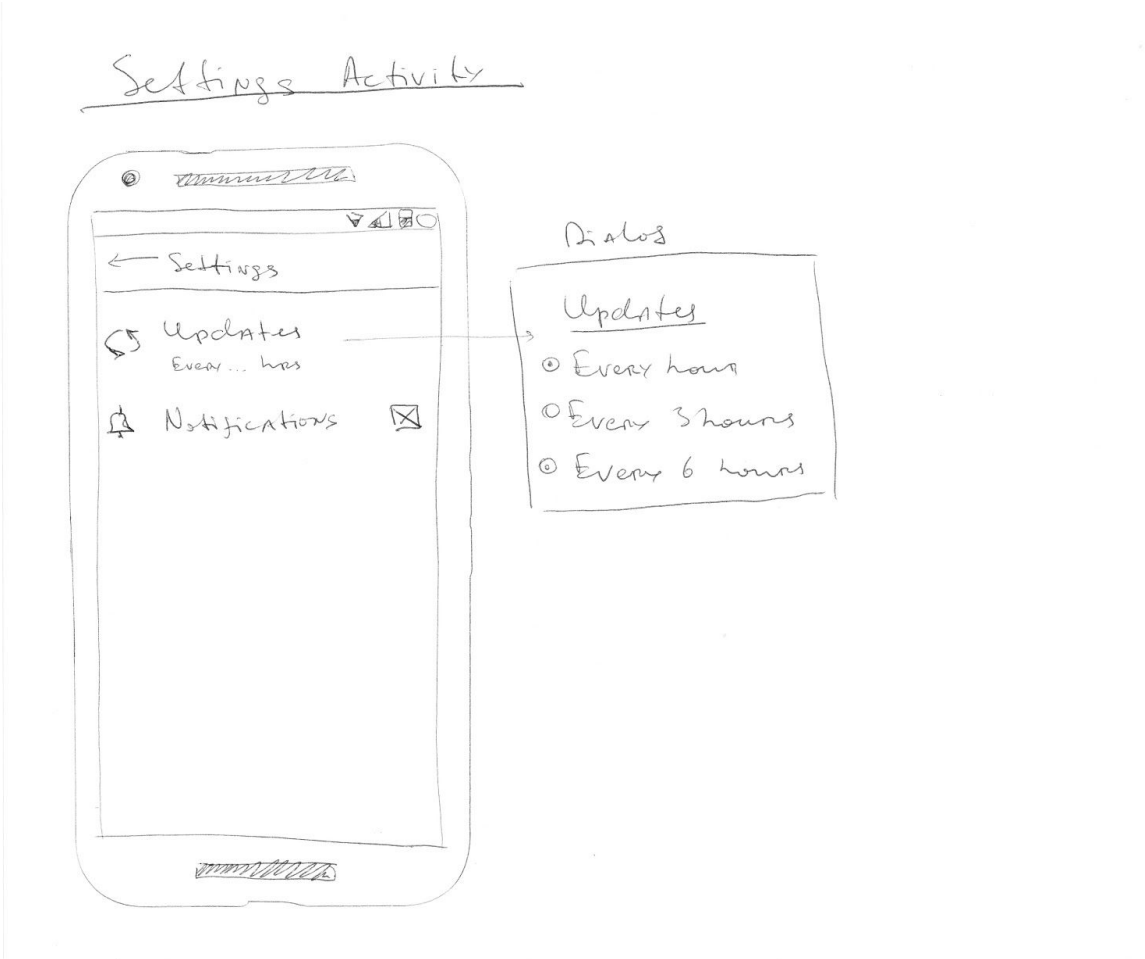
04 Detail Activity



This is the Detail activity. It displays all the accumulated statuses for a single tracking number. Its elements, top to bottom:

- A tracking number or a package name
- Maps fragment which displays the location of the most recent status event
- List of statuses, newest first
- A FAB button that allows to share the accumulated statuses

05 Settings Activity



This is the Settings activity as called from the drawer. It allows the user to choose the update interval with the options like “Manual”, “Every hour”, Every 3 hours”, and so on. Also the user can choose to disable the notifications.

Key Considerations

How will your app handle data persistence?

I will store the application data (tracking numbers and status records) in the SQLite database and will build a Content Provider to access them. The application preferences will be stored in the Shared Preference object.

Describe any corner cases in the UX.

Nothing to track: no tracking numbers entered yet, the application displays a short instruction on what to do next.
No status to display for a tracking number: the application with display a short prompt on what’s going on, and what is expected to happen soon.

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

OkHttp for the interactions with the tracking web service.
Glide for loading and displaying the package photos.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Set up the local and remote git repositories
- Link local repository to the remote one
- Set up the local dev branch and push it to the remote repository.
- Create a master-detail project from the template

Task 3: Implement sync

- Add a sync service
- Add a sync adapter
- Add a method to query the actual USPS tracking web service
- Test the tracking method

Task 4: Implement data layer

- Design the database
- Add a data contract
- Add a data provider
- Add a database helper
- Connect tracking method to the data layer

Task 5: Implement UI for Each Activity and Fragment

- Update the main activity layout to conform to the requirements of the Material design
- Build the main activity item layout
- Wire up the main activity to display the data
- Update the detail activity layout to conform to the requirements of the Material design
- Build the detail activity item layout
- Wire up the detail activity to display the data

Task 6: Improve the UI

- Implement the Overflow menu (sort order)
- Implement the method to enter the new tracking number
- Implement the method to delete the tracking number
- Implement the method for detecting the package arrival, and excluding it from the sync
- Add Refresh FAB to the main activity
- Implement sending a request to refresh the active tracking numbers

- Add Share FAB to the detail activity
- Implement sharing of the most recent status
- Add notifications for when the status changes

Task 8: Add Google Play Services

- Add a test AdMob fragment to the main page
- Add a Maps fragment to the details page

Task 9: Improve the UI pt.2

- Add a drawer panel to display the options for switching between the current tracking numbers and the archived ones
- Implement the Settings activity (update interval, notifications enabled/disabled)
- Implement moving the tracking number to the archive
- Add home screen widget

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"