

[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.](#)

[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: [wolfpaulus](#)

Hwy67

The Back Story

The fire season started early in 2013. Early in July, the Chariot Fire burned 7,000 acres, about 9 miles southeast of Julian, at the Sunrise Highway. At one point, more than 2,000 fire fighters were involved, but couldn't stop the fire from destroying 149 structures. The Chariot Fire was burning about 30 miles (42 miles by road) from my house and I was even more interested, since I had been hiking close to the area, just a day prior. Fire related online resources provided us with almost real-time information and through cameras that the High Performance Wireless Research and Education Network HPWREN has permanently deployed on Mount Laguna, we literally got the picture.

Google maps, overlaid with HMS Thermal Satellite data, clearly showed the direction and expansion of the fire. Additionally, we followed several live audio feeds, which we found on Broadcastify, and of course all the RSS feeds available from local news stations, the California State government, Inciweb, and many more. While we were discovering and sharing this information, an ever growing group of people was turning to my twitter feed and web site to find current information about the Chariot Fire. This experience and the absence of a single, comprehensive news source, covering events of local importance, sparked the idea of building an aggregator that would automatically find and provide easily access to relevant information. The first application, of what we dubbed "The Highway Platform", implemented mostly in Python, can now be found at <http://hwy67.us>, a Website we named after California State-Route 67,

which goes through the geographical region covered on this site. Currently, information is gathered from over 30 news sources at varying frequencies, but no less than 4 times per hour. A sophisticated filter algorithm identifies and categorizes relevant items, which are then published and linked on the site. Regionally important information, including local weather, jobs, and live audio feeds are mapped into the site as well. The Highway Platform's aggregator pushes data, including text, images, GPS location information, and seismic data that passed through the filter, into a remote, highly customized WordPress content management system. A visually pleasing theme, featuring responsive web design (RWD), is then used to present the gathered information.

Description

This app makes the content of the <http://hwy67.us> Wordpress CMS driven site available through a mobile app. The user can select the areas (e.g. names of the covered towns) he most interested, basically filtering the content by geographical region.

The content server implement Google Cloud Message notifications, which will be displayed as news alerts and also trigger content synchronization.

Finally, the user can setup quiet periods, during which notifications would not be made visible or audible.

Intended User

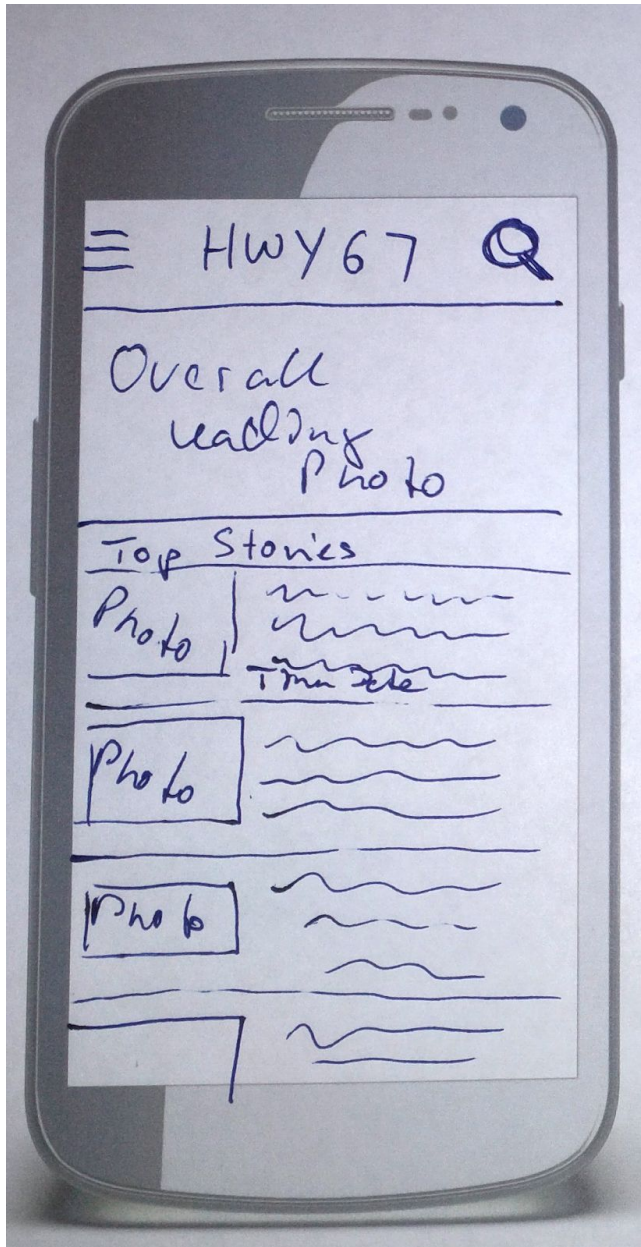
The intended users are the people who live or are otherwise interested in the area that is covered by the <http://hwy67.us> news site.

Features

- Saves and displays information pulled from a wordpress driven content management
- Configurable content filter (name of towns covered by the site).
- Google Cloud message notifications will pushed to the app, based on the content filter settings.
- Configurable quiet periods during which notifications will be consumed but not exposed.

User Interface Mocks

Main Screen / Photo of the day / Headlines



The main screen show the the big photo of one of today's stories.

Below is a list of smaller images and headlines, which can be click on to navigate to that story

Main Screen with Menu exposed



Menu allows to navigate to a view that shows stories of just one, the selected town.
Menu also allows to open settings (see below) and contact information screen.

A single story

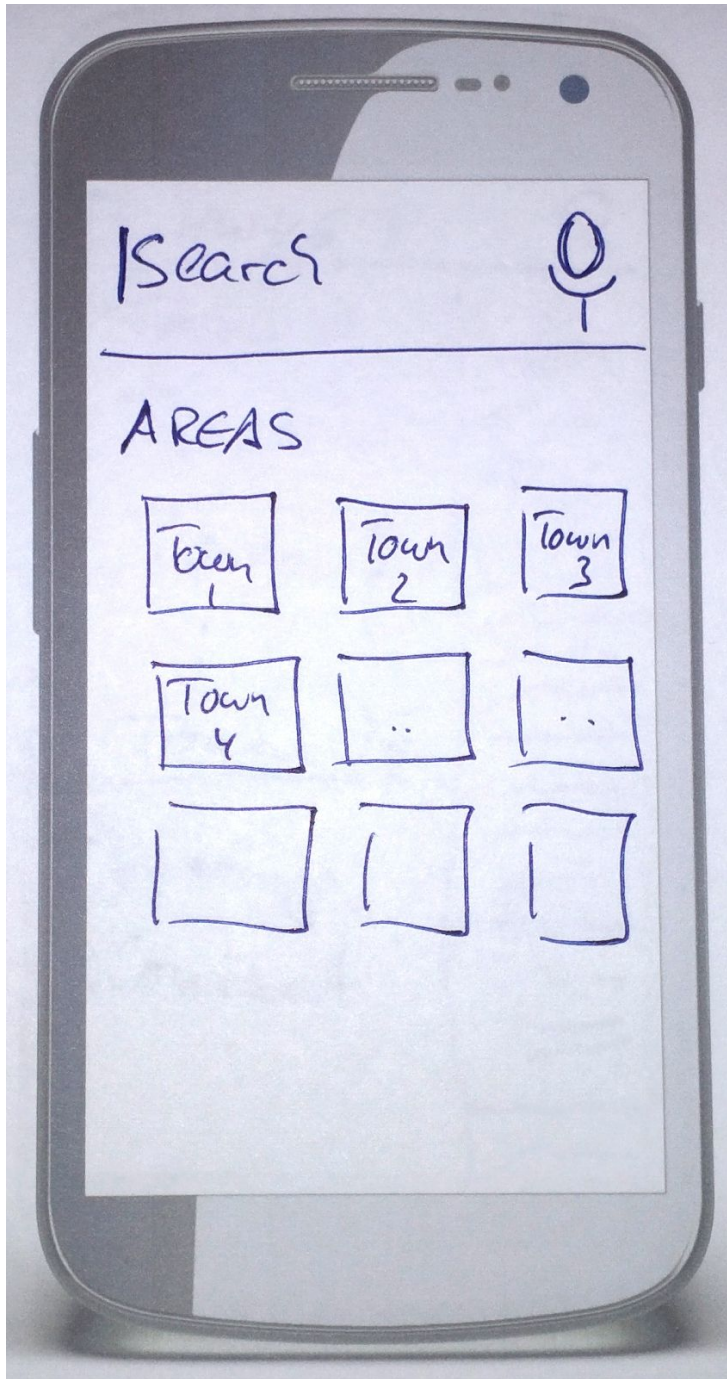


Big photo of that story (in case we have one)

Content source, author, date, time

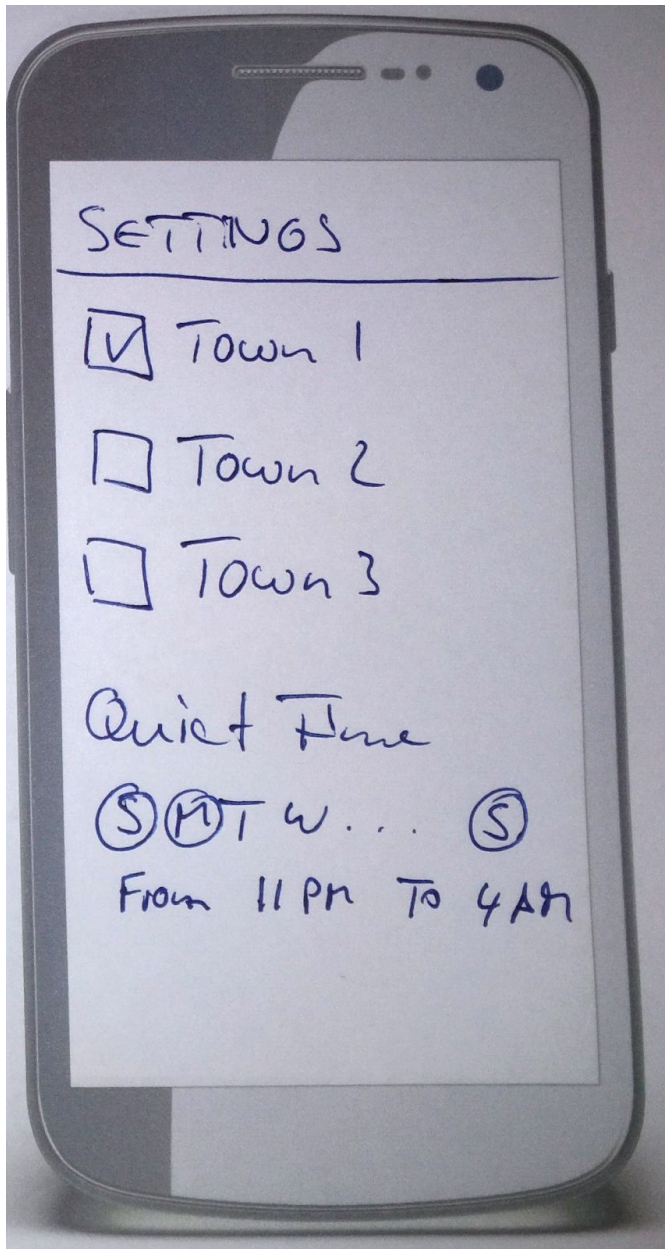
The complete story

Search Page



Search page for voice enable search also shows towns images to quickly navigate to stories of just the selected town. Towns that have been un-selected in the settings page would be shown disabled.

Settings Page



The settings view allows the selection of only some towns (by default all are selected)
A quiet period can be selected, at which incoming push notifications will not be visible or audible

Key Considerations

How will your app handle data persistence?

Content Provider will be created to access data.

A fixed (exact number TBD) number of stories for each town (i.e. the text) will be stored on the phone.

Describe any corner cases in the UX.

Large photos are not stored locally, i.e. empty space or placeholder is displayed instead.

At least one town needs to be selected in settings - otherwise there would be nothing to display.

Still app needs to be prepared to have no data available from the server at first launch. Content server could be down or have no stories for the selected area.

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

I have not decide which but will use libraries for photo display (most likely glide), HTTP/Rest access of wordpress xml interface, maybe GSON, if needed. I'm still looking for a lib that is optimized for accessing wordpress cms.

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

Starting point is Android Studio and the templates for setting up multiple activities:

- Navigation Drawer Activity for Main screen with menu
- Blank Activity for Single Story
- Settings Activity for selecting town names
- etc

With the library selection finalized, finding there maven home and configuring the gradle project file to include their home, name, and version

Task 2: Content

- Implement the Content Provider to access the content, originating at the wordpress site
- Decide how much data should be kept on the phone, how many stories over all, how many for each town.
- Implement content search feature that can find stories based on text search.

Task 3: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for Navigation draw
- Build UI for Single story view
- Build UI for Search
- Build a UI for settings (towns / quiet period)
- Build UI displaying Contact information

Task 4: Your Next Task

- Implement subscribing to push notifications based on selected towns.
- Implement displaying of incoming alerts (if outside quiettime)

Task 5: Server Side push

- Implement the GCM backend triggering sending push notifications to subscribed clients