GitHub Username: Almaral-Engineering

Earthquake Monitor

Description

Earthquake Monitor keeps you informed about the latest earthquakes around your location using the information given by the USGS database.

Intended User

This app is suitable for anyone, mainly for people who lives or has family in zones where earthquakes are abundant or have high probability to occur.

Features

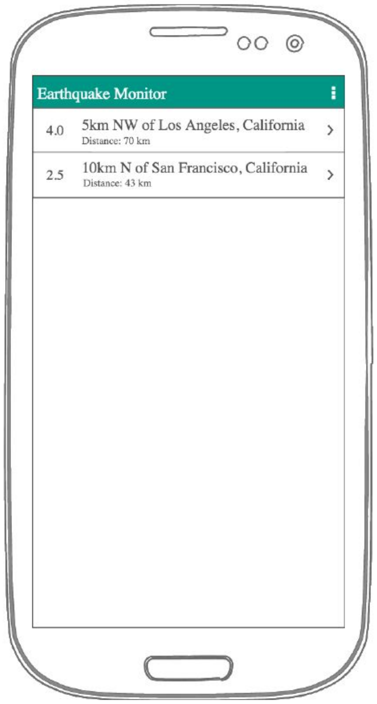
List the main features of your app. For example:

- Saves earthquake data to see it without connection
- Capability to share the data
- Set the area, timing and magnitude to see the latest earthquakes.
- View the earthquake location in google maps.

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1

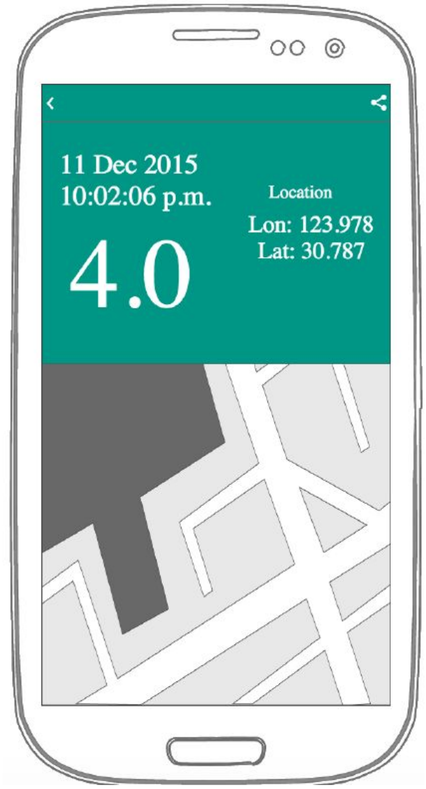


4.0 5km NW of Los Angeles, California
Distance: 70 km

2.5 10km N of San Francisco, California
Distance: 43 km

This is the main screen, on it the user is going to see all the earthquakes registered for the chosen settings.

Screen 2



11 Dec 2015
10:02:06 p.m.

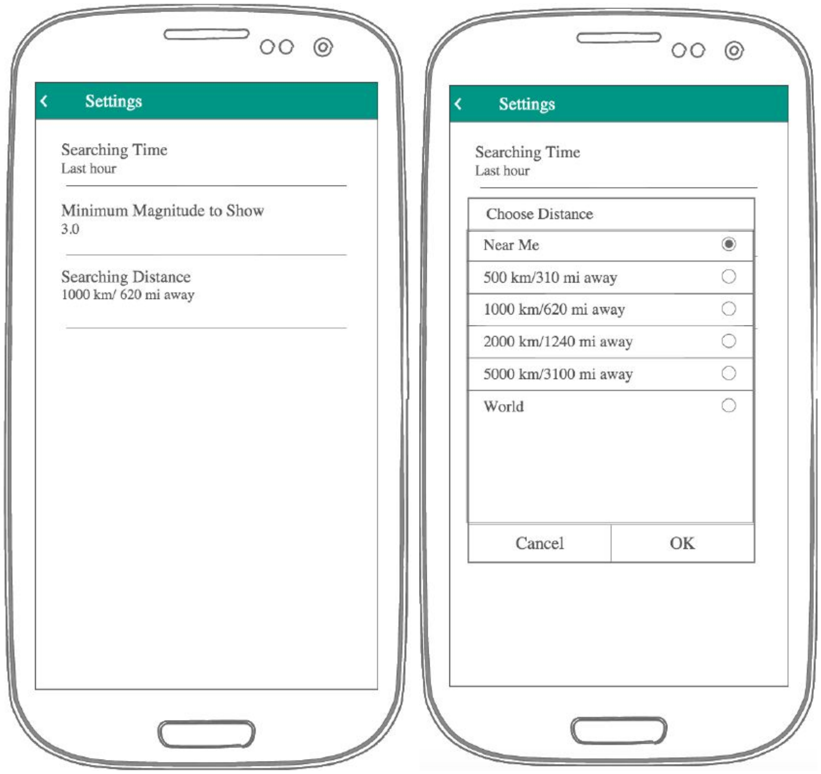
4.0

Location

Lon: 123.978
Lat: 30.787

This is the details screen, this screen will show the exact location of the earthquake in a map, besides of the hour, date and coordinates. The user is going to have the possibility of sharing the data using other applications as facebook or whatsapp.

Screen 3



Settings

Searching Time
Last hour

Minimum Magnitude to Show
3.0

Searching Distance
1000 km/ 620 mi away

Settings

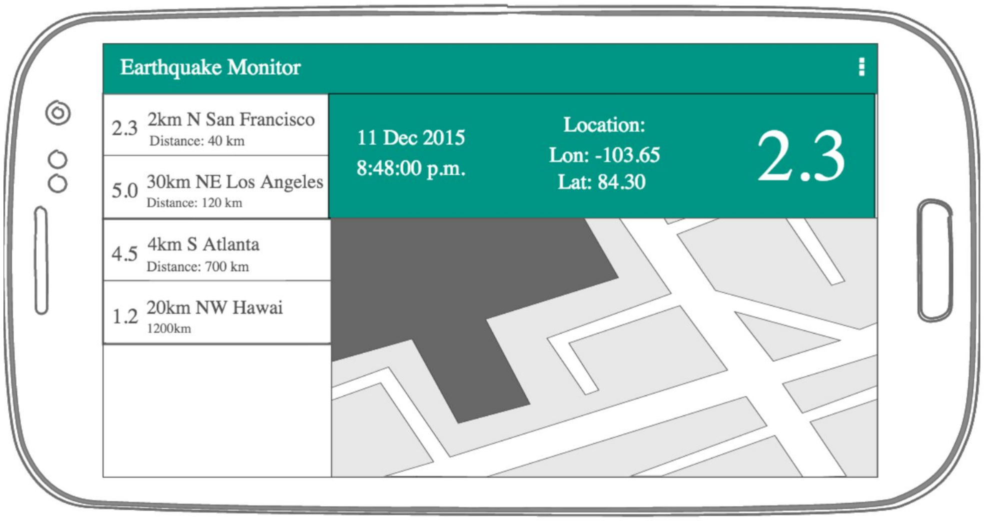
Searching Time
Last hour

Choose Distance

- Near Me ●
- 500 km/310 mi away ○
- 1000 km/620 mi away ○
- 2000 km/1240 mi away ○
- 5000 km/3100 mi away ○
- World ○

Settings screen will allow the user to set the time to show earthquakes (Last hour, day and week), the minimum Magnitude to show and the searching distance.

Tablet screen 1



Earthquake Monitor

2.3	2km N San Francisco Distance: 40 km
5.0	30km NE Los Angeles Distance: 120 km
4.5	4km S Atlanta Distance: 700 km
1.2	20km NW Hawai 1200km

11 Dec 2015
8:48:00 p.m.

Location:
Lon: -103.65
Lat: 84.30

2.3

Tablets are going to show all the data in a single screen, so the user is not going to have to go forth and back to look the data.

Key Considerations

How will your app handle data persistence?

I am going to use Shared Preferences for the settings and SQLite with a Content Provider for the earthquake data to see the last fetched data without connection.

Describe any corner cases in the UX.

There is an internet lost connection: Validate the connection in any required part, if there is no connection, do whatever necessary to handle it.

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

Volley for the requests and Google Maps API to show the earthquake locations, also Google Location API to get the user's location.

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

Create and setup the new project.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for Detail Activity

Task 3: Implement Web Service fetch data

- Use Volley to fetch the data from web service

Task 4: Implement Google Maps API

- Add Google Maps API Support to see the earthquake location

Task 5: Implement Settings Screen

Setup settings screen to filter the data for time, magnitude and distance

- Build Settings UI.
- Add Location API to know the distance to the earthquake having the coordinates.

Task 5: Add offline support

Create a database to show the last fetched data even when if the user lost internet connection

- Implement SQLite
- Create a content provider
- Test the database

Task 6: Add offline support

Create visual tests for app

- Integrate espresso to test the different screens of the app.

Task 7: Last Tweaks

Last tweaks to make the app more robust

- Handle internet lost connection.
- Validate volley errors and for things as null values.
- Add empty views.