GitHub Username: suryachintu

QuakeAlert

Description

QuakeAlert is an app which shows the list of earthquakes information that occurred in the past day and past hour.

Intended User

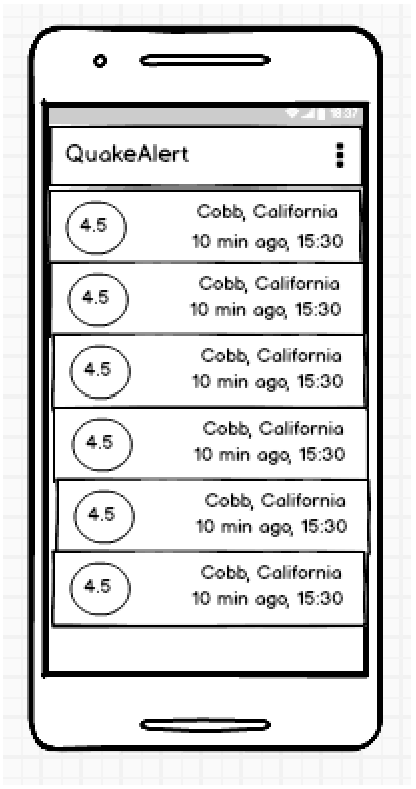
QuakeAlert is intended for all the users who are eager to know about the earthquakes occurring nearby and around the world. It engages the users with notifications when there is earthquake alert nearby them.

Features

- Live earthquake information around the world.
- Push notification for earthquake alerts.
- Able to see the location on a map.
- Magnitude and nearby cities that the earthquake occurred.
- Distance away from u where the earthquake has occurred.

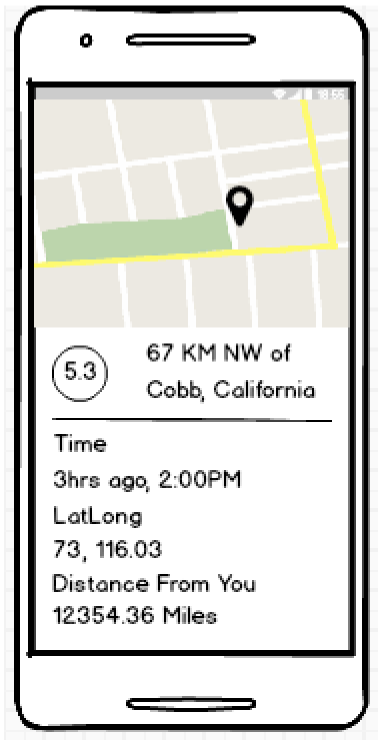
User Interface Mocks

Main Screen



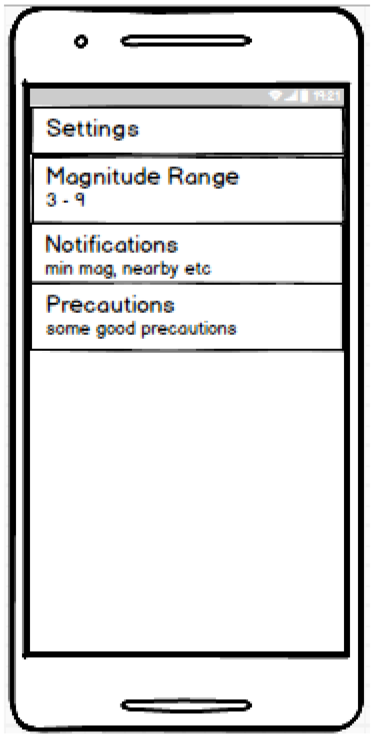
On the main screen, there will be a recycler view showing the list of earthquakes and place where it has occurred, magnitude, and time.

Details Screen



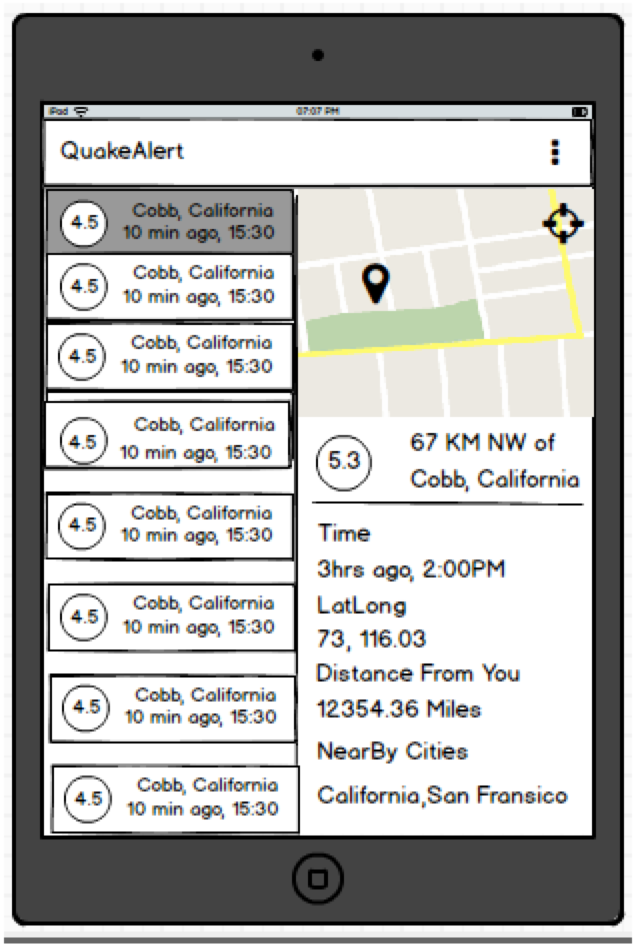
On the details screen user, can able to see the location of the earthquake in a google map, its magnitude, time, latitude and longitude, how far is from you, nearby cities where the earthquake has occurred.

Settings Screen



On the settings screen user, can able to set the magnitude range of the earthquakes, change the notification settings like only show the notifications that occurred nearby me or country or worldwide or specific magnitude and precautions tutorial when an earthquake occurred.

Tablet Screen



4.5 Cobb, California 10 min ago, 15:30

4.5 Cobb, California 10 min ago, 15:30

4.5 Cobb, California 10 min ago, 15:30

4.5 Cobb, California 10 min ago, 15:30

4.5 Cobb, California 10 min ago, 15:30

4.5 Cobb, California 10 min ago, 15:30

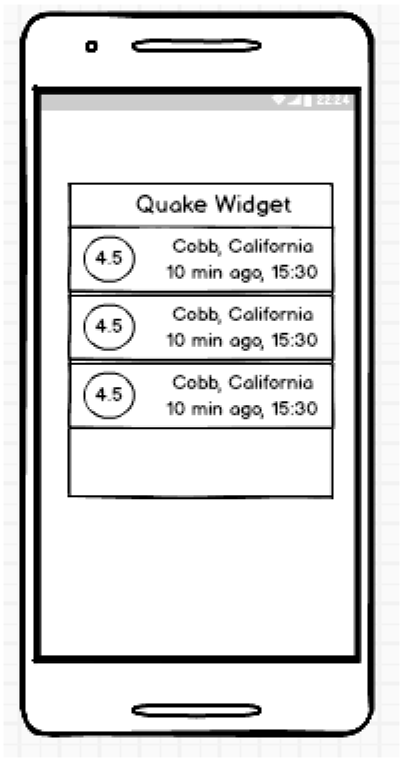
4.5 Cobb, California 10 min ago, 15:30

4.5 Cobb, California 10 min ago, 15:30

5.3 67 KM NW of Cobb, California

Time
3hrs ago, 2:00PM
LatLong
73, 116.03
Distance From You
12354.36 Miles
NearBy Cities
California,San Fransico

Widget Screen



Key Considerations

How will your app handle data persistence?

App handles data persistence by use of content providers.

Describe any corner cases in the UX.

User can return to the main screen by pressing the back button or back arrow in toolbar. If there is no internet connection while fetching the data for the first time when the user installs the app it will displays a text showing no internet connection.

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

OkHttp: To simplify the networking calls.
ButterKnife: Field and method binding for android views.

Describe how you will implement Google Play Services.

Google play location services: To display the location of the place where earthquake has occurred and to show the users current location.

AdMob: To show ads at the bottom each screen.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries and set up git repository

Task 2: Implementation of main screen

- Set up recycler views to handle the data.
- Get the data from the web.
- Parse the data according to the requirements.

Task 3: Implement the database

- Implement database to store the data of the earthquake feeds.
- Implement Content Providers.
- Populate the recycler views using Cursor Loaders.

Task 4: Implement Details screen

- Get the key from the google developer console.
- Display the location on the map.
- Fill the data according to the details screen UI.

Task 5: Implement Settings Screen

- Store the magnitude settings using shared preferences and fetch data according to it.
- Implement Notification settings like enable notifications or not, if notifications enabled when to trigger a notification like when there is earthquake nearby or in the country or worldwide by fixing a certain magnitude.
- Implement a quick tutorial to take precautions when earthquake occurs.

Task 6: Implement Sync adapters

- Use sync adapters to fetch the data for every three hours

Task 7: UI Polishing

- Use of Material Design Principles to make app UI look beautiful.

Task 8: Set Up Multi Pane UI

- Implement master details view for the tablet usage.

Task 9: Implement Widgets

- Implement widgets.

Task 10: Publishing app in playstore

- Publishing app.