Homework 9: Event Search Android App

1. Objectives

- Become familiar with Java, XML, Android Lifecycle and Android Studio for Android app development.
- Build a good-looking Android app.
- Learn to use the Google Maps APIs and Android SDK.
- Get familiar with third party libraries like Picasso, Glide and Volley.

2. Background

2.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android application development, based on IntelliJ IDEA - a powerful Java IDE. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle - based build system.
- Build variants and multiple apk file generation.
- Code templates to help you build common app features.
- Rich layout editor with support for drag and drop theme editing.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- ProGuard and app-signing capabilities.
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

More information about Android Studio can be found at:
http://developer.android.com/tools/studio/index.html

2.2. Android

Android is a mobile operating system initially developed by Android Inc., a firm purchased by Google in 2005. Android is based upon a modified version of the Linux kernel. As of Nov 2018, Android was the number 1 mobile OS, in unit sales, surpassing iOS, while iOS was still the most profitable platform.

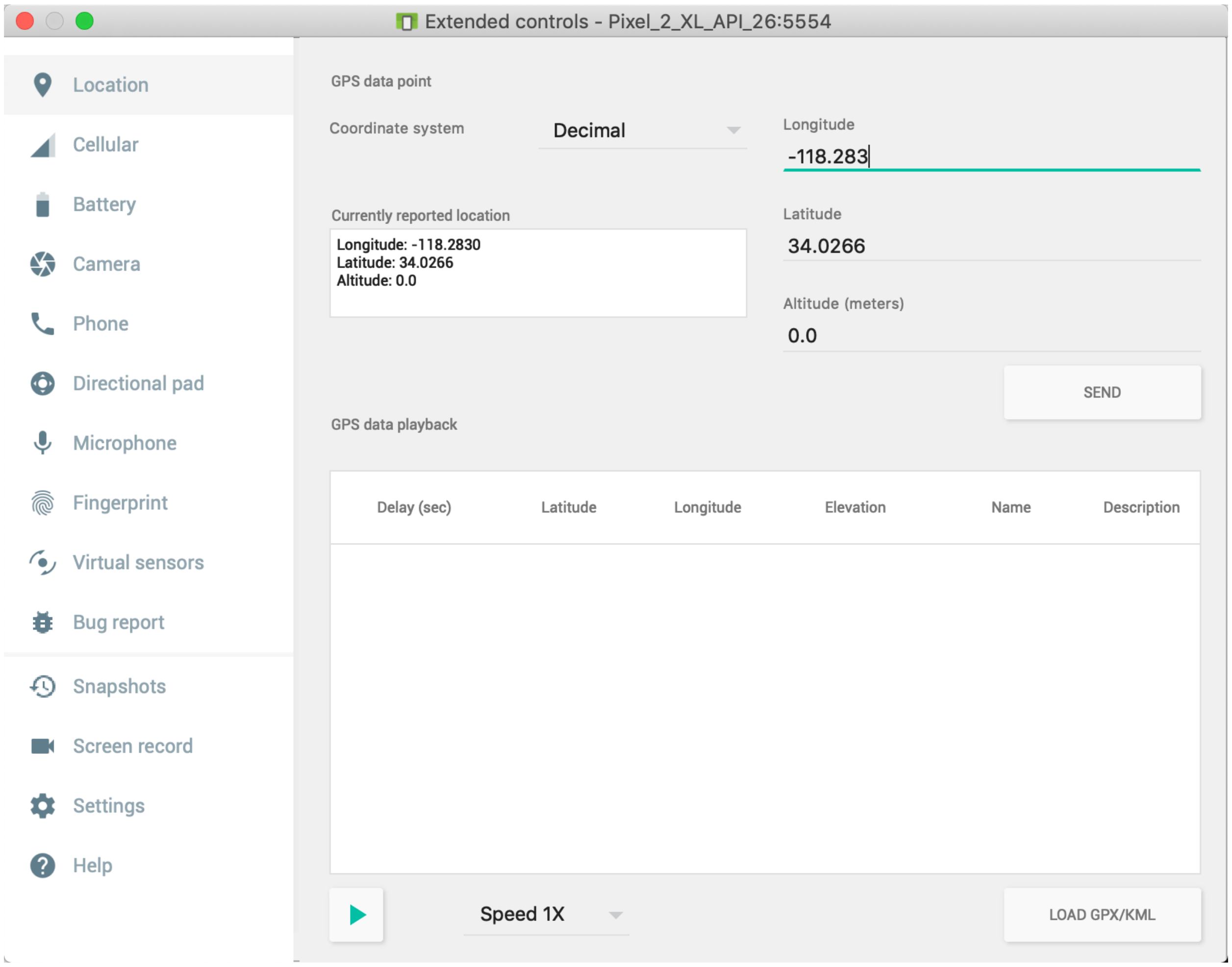
The Official Android home page is located at:
http://www.android.com/

The Official Android Developer home page is located at:
http://developer.android.com/

3. Prerequisites

This homework requires the use of the following components:

- Download and install Android Studio. Technically, you may use any other IDE other than Android Studio such as Eclipse, but the latest SDKs may not be supported with Eclipse. We will not be providing any help on problems arising due to your choice of alternate IDEs.
- You must use the **emulator**. Everything should just work out of the box. Before you start, you need to manually set the emulator's longitude and latitude nearby USC. An example is shown in Figure 0.



Coordinate system: Decimal
Longitude: -118.283
Latitude: 34.0266
Altitude (meters): 0.0

Currently reported location
Longitude: -118.2830
Latitude: 34.0266
Altitude: 0.0

Figure 0: Location Setting of Emulator

This app has been designed and implemented in a **Pixel 2XL emulator** by using **SDK API 26**. It is highly recommended that you use the same virtual device and API level to ensure consistency.

4. High Level Design

This homework is a mobile app version of Homework 8. In this exercise, you will develop an Android application, which allows users to search for the event ticket, look at information about it, save some as favorites and post on Twitter about the event. You should reuse the Node.js backend service you developed in Homework 8 and follow the same API call requirements.

5. Implementation

5.1 Search Form

The initial interface is shown in **Figure 1**. There are two tabs in this interface: search and favorite.

For the search tab, it has the following fields:

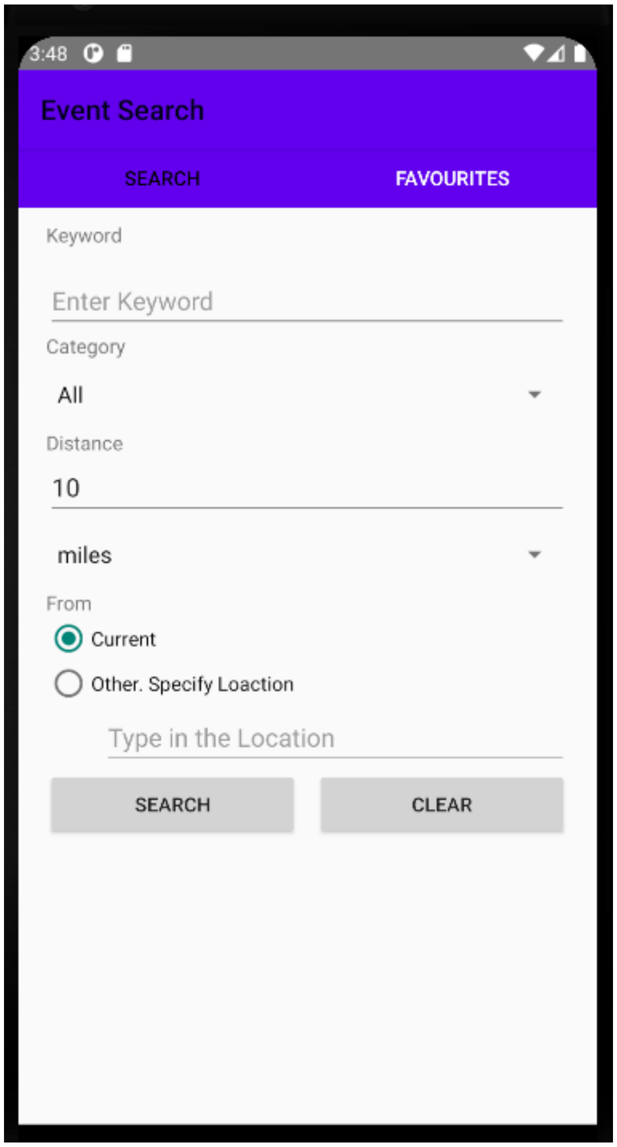
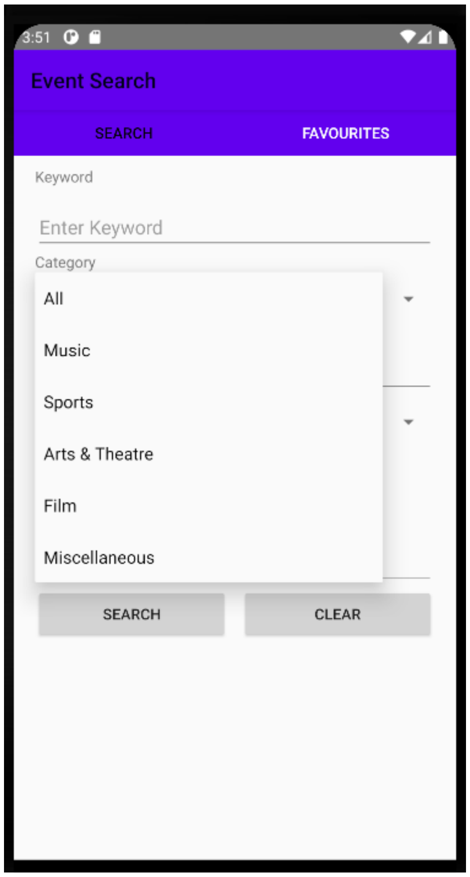


Figure 1 The Event Search App

- **Keyword**: An EditText component allowing the user to enter the keyword.
- **Category**: A Spinner view allowing the user to choose a category. When the user taps on this field, a dropdown list should display for selecting a category, as shown in Figure 2. Make sure you include all the categories in homework 8.
- **Distance**: An EditText (type:number) view allowing the user to enter the distance and the default value is 10.
  A Spinner for the user to select unit: “miles” or “kilometers”.
- **From**: Two Radio Buttons to select “Current Location” or “Other”. You should get the location from your emulator for users that choose “Here”
  An EditText component allowing the user to enter a location for “Other”. The EditText should only be enabled when the corresponding Radio Button is checked.
- **Search**: A button to get the input information of each field, after validation. If the validation is successful, then the events would be fetched from the server. However, if the validation is unsuccessful, appropriate messages should be displayed and no further requests would be made to the server.
- **Clear:** A button to clear the input fields and reset them to default values when applicable. It should also remove any validation error messages.



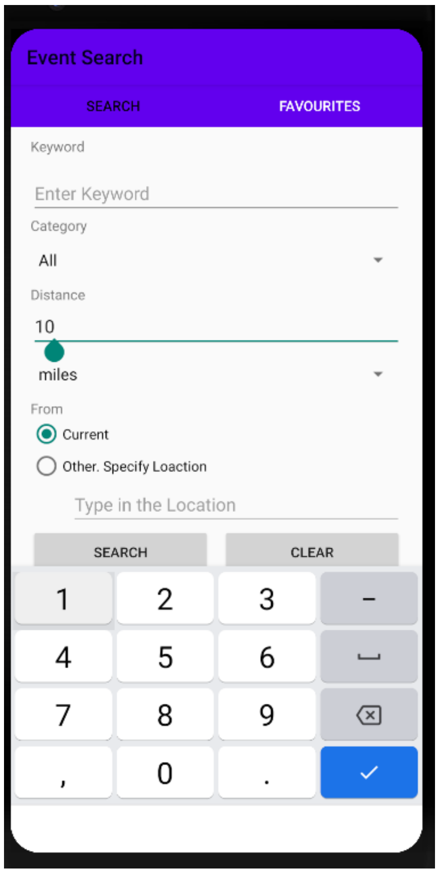
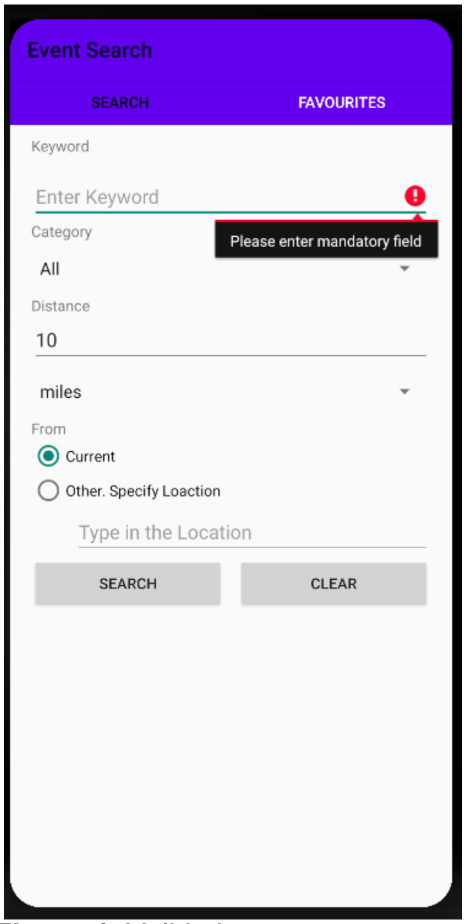


Figure 2 : Category

Figure 3 : Distance of type number

The validation for an **empty keyword** must be implemented. If the user does not enter anything in the EditText or just enters some empty spaces, when he/she presses the Search button you need to display an appropriate message to indicate the error, as shown in **Figure 5**. The same should be done when "**From**" location is not entered, and that option in enabled using the radio button as shown in **Figure 5**.



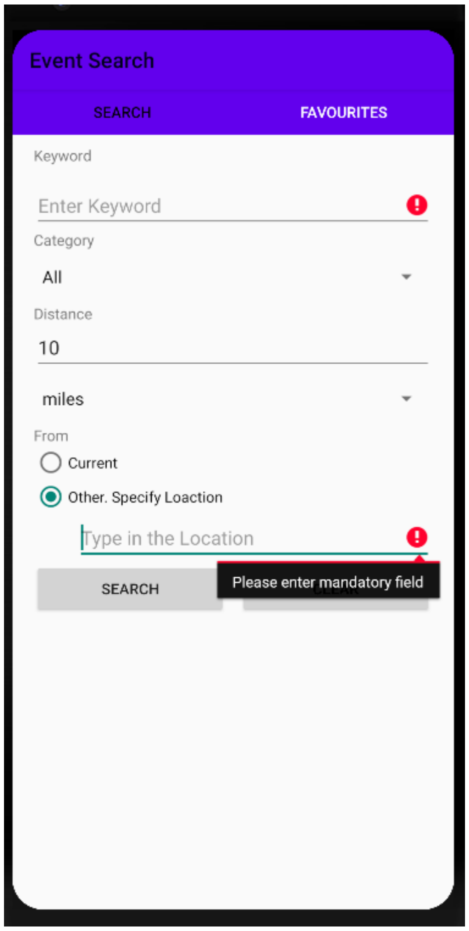
All

10

miles

Please enter mandatory field

Figure 4: Validation error messages



All

10

miles

Please enter mandatory field

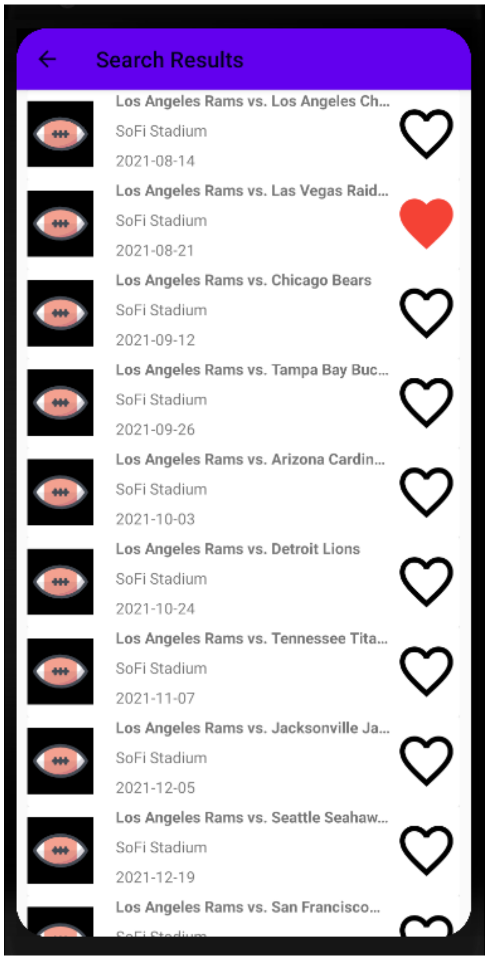
Figure 5: Validation error messages

5.2 Search Results

When the user taps the SEARCH button and all validations pass, your app loads the search results page. After you get the data from your backend display the result page as a list using *RecyclerView* or *ListView*, as shown in **Figure 7**. The *RecyclerView* or must be scrollable. They also provide a 'back button' to navigate back the search/favorite interface.

Each of the item in the list should have the following:

- Category image (See the mapping between segment and icons on section 6. 1)
- Name of the event
- Name of the venue
- Date of the event
- A heart-shaped "Favorite" button
  See homework 6 for more details about these fields.


Search Results

Los Angeles Rams vs. Los Angeles Ch...
SoFi Stadium
2021-08-14

Los Angeles Rams vs. Las Vegas Raid...
SoFi Stadium
2021-08-21

Los Angeles Rams vs. Chicago Bears
SoFi Stadium
2021-09-12

Los Angeles Rams vs. Tampa Bay Buc...
SoFi Stadium
2021-09-26

Los Angeles Rams vs. Arizona Cardin...
SoFi Stadium
2021-10-03

Los Angeles Rams vs. Detroit Lions
SoFi Stadium
2021-10-24

Los Angeles Rams vs. Tennessee Tita...
SoFi Stadium
2021-11-07

Los Angeles Rams vs. Jacksonville Ja...
SoFi Stadium
2021-12-05

Los Angeles Rams vs. Seattle Seahaw...
SoFi Stadium
2021-12-19

Los Angeles Rams vs. San Francisco...

Figure 6: List of search results

Tapping the **favorite** button (the heart) would add the corresponding event into the favorites list. If the Name of the events goes beyond the heart, it should be hidden and show "…" to indicate the name is longer than shown.

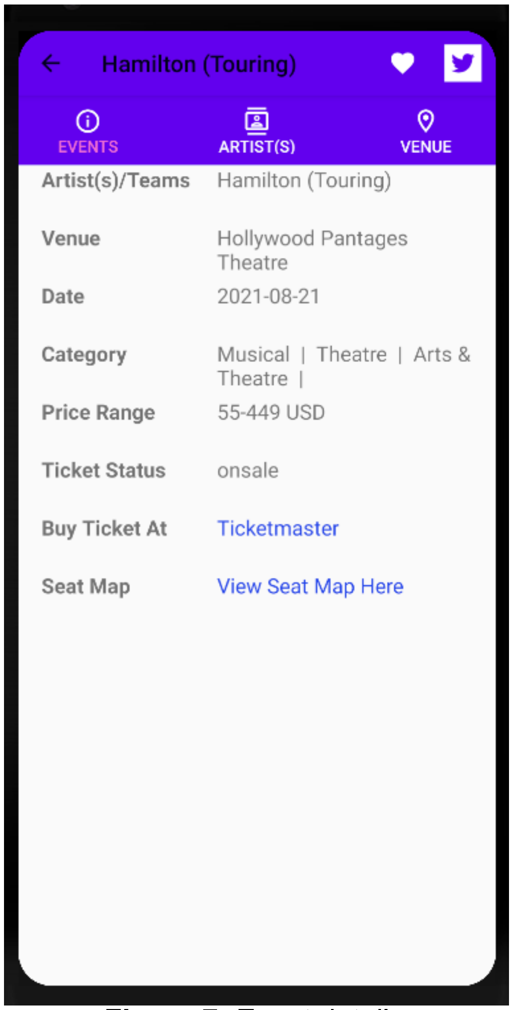
5.3 Event Details

Tapping on an item in the result list should show details of that event with four tabs: Event, Artists and Venue.

Users should be able to switch between tabs by both swiping and tapping on a tab. The *ActionBar* should also include the following elements:

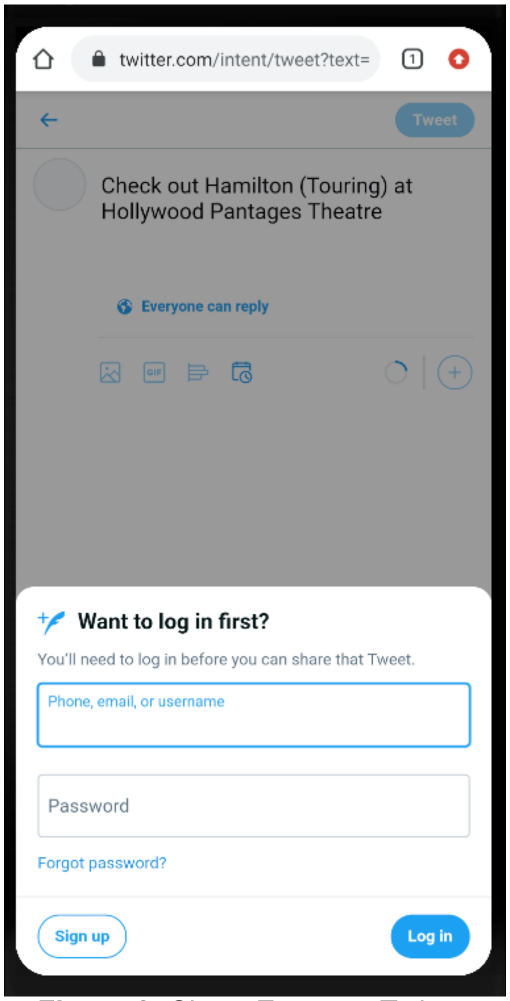
- A **back button,** which navigates back to the search results list.
- A **title**, which is the name of the event.
- A **favorite button** to add/remove the event to/from the favorite list.

- A **twitter button**, to share the event details on Twitter. Once the button is tapped, a web page should open to allow the user to share the event information on Twitter, as shown in **Figure 8**.



Artist(s)/Teams	Hamilton (Touring)
Venue	Hollywood Pantages Theatre
Date	2021-08-21
Category	Musical | Theatre | Arts & Theatre |
Price Range	55-449 USD
Ticket Status	onsale
Buy Ticket At	Ticketmaster
Seat Map	View Seat Map Here

Figure 7: Event details



Check out Hamilton (Touring) at Hollywood Pantages Theatre

Want to log in first?

You'll need to log in before you can share that Tweet.

Phone, email, or username

Password

Forgot password?

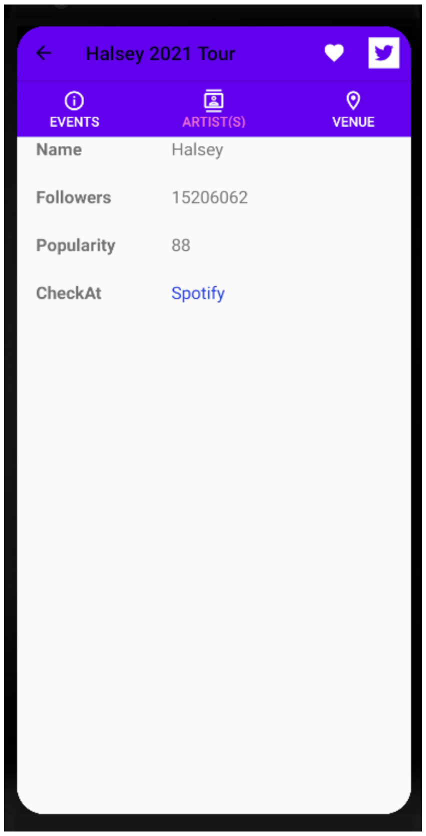
Figure 8: Share Event on Twitter

5.3.1 Event Tab

The fields in **Figure 11** should be shown in the Event tab. In homework 9, the seat map will be the URL of the image and when user click on this URL, it will go to the browser to show the seat map.

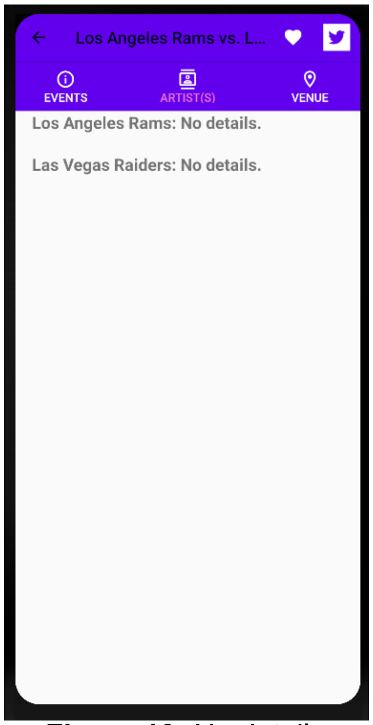
5.3.2 Artist(s) Tab

Shown below. For multiple artists in homework #9, you could only show the **first two** artist's music profiles (if applicable). If the there is no detail on **Spotify** (if for example it is a team), display "No details" for each. If there no records at all, please display an appropriate message as shown in the following pages.



Name	Halsey
Followers	15206062
Popularity	88
CheckAt	Spotify

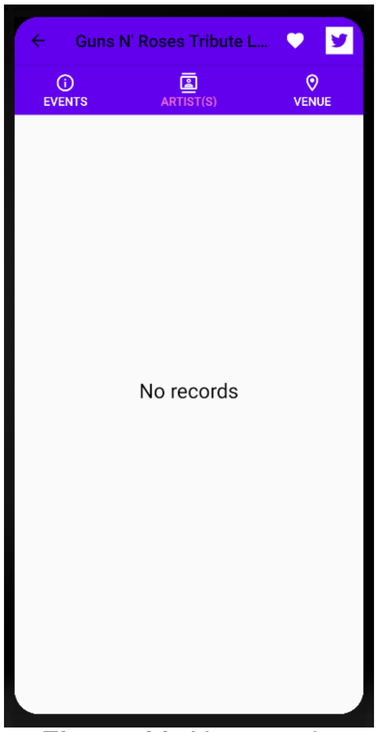
Figure 9: Artists Tab with Music Artist



Los Angeles Rams: No details.

Las Vegas Raiders: No details.

Figure 10: No details



No records

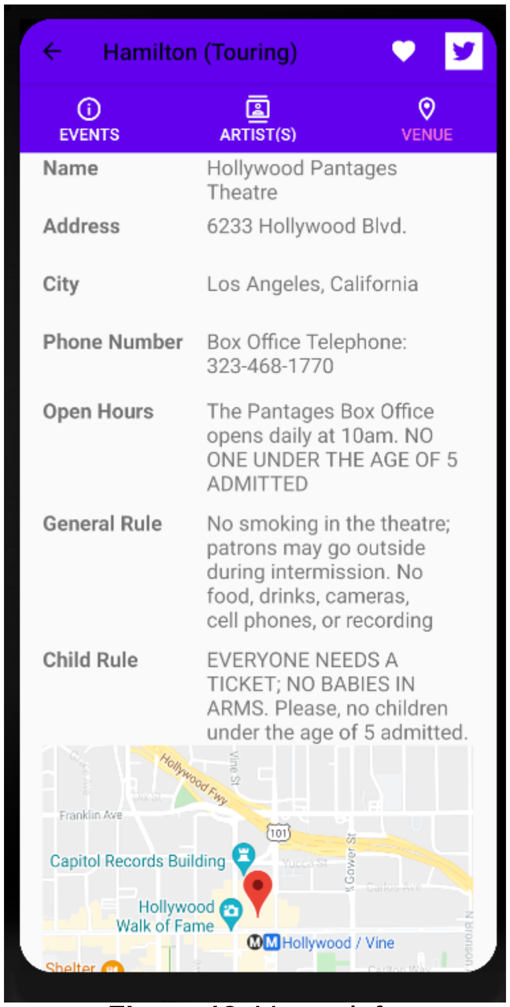
Figure 11: No records

5.3.3 Venue Tab

As shown in **Figure 12** and **Figure 13**, there are two elements in this tab:

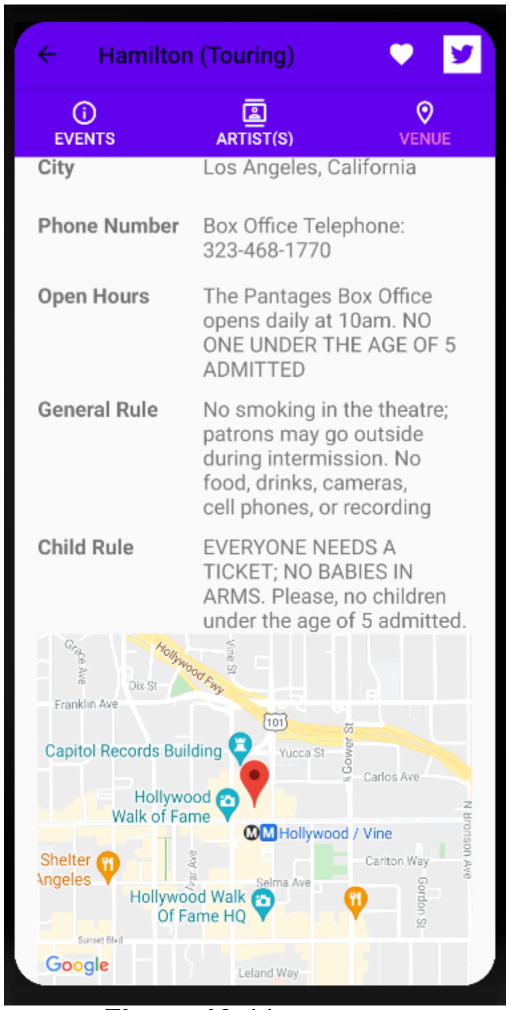
- **Details of the venue table:** Shown below in the figures.
- **Map:** Same as homework 8, you should render a google map with a maker centered in the map of the venue location.
  The maps should be rendered using the Google Maps SDK for Android.
  https://developers.google.com/maps/documentation/android-api/

This view should be scrollable since the details of the venue table may be too long.



Name	Hollywood Pantages Theatre
Address	6233 Hollywood Blvd.
City	Los Angeles, California
Phone Number	Box Office Telephone: 323-468-1770
Open Hours	The Pantages Box Office opens daily at 10am. NO ONE UNDER THE AGE OF 5 ADMITTED
General Rule	No smoking in the theatre; patrons may go outside during intermission. No food, drinks, cameras, cell phones, or recording
Child Rule	EVERYONE NEEDS A TICKET; NO BABIES IN ARMS. Please, no children under the age of 5 admitted.

Figure 12: Venue Info



City	Los Angeles, California
Phone Number	Box Office Telephone: 323-468-1770
Open Hours	The Pantages Box Office opens daily at 10am. NO ONE UNDER THE AGE OF 5 ADMITTED
General Rule	No smoking in the theatre; patrons may go outside during intermission. No food, drinks, cameras, cell phones, or recording
Child Rule	EVERYONE NEEDS A TICKET; NO BABIES IN ARMS. Please, no children under the age of 5 admitted.

Figure 13: Venue map

5.4 Favorite list

The favorite events should be displayed in a list using a RecyclerView/ListView. Each of

the items in the list includes an event catalog image, event name, venue name and date, as shown in **Figure 14**. If there are no favorite events, "No Favorites" should be displayed at the center of the screen, as shown in **Figure 15**.
Please note: As soon as you remove the last element, the tab should change immediately to show the “No Favorite” display.

Like in search results, pressing the favorite icon here should remove the event from the favorites list.

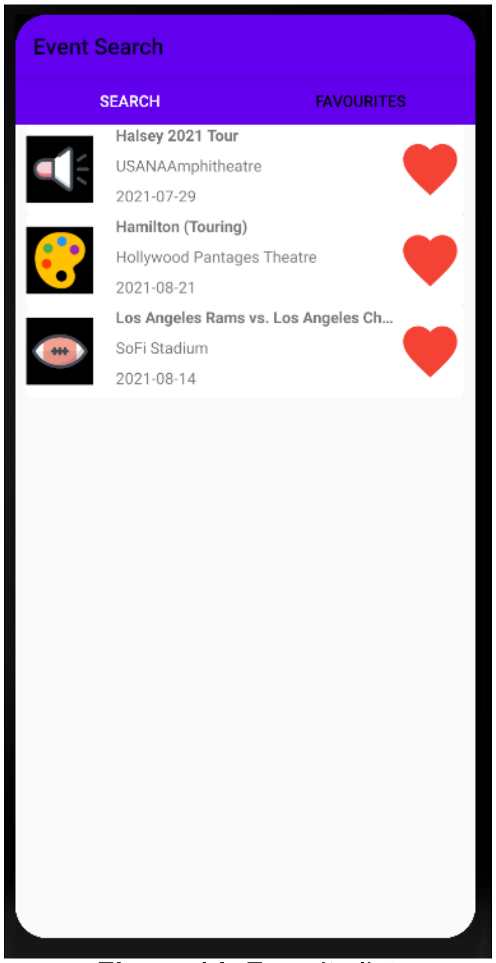


Figure 14: Favorite list

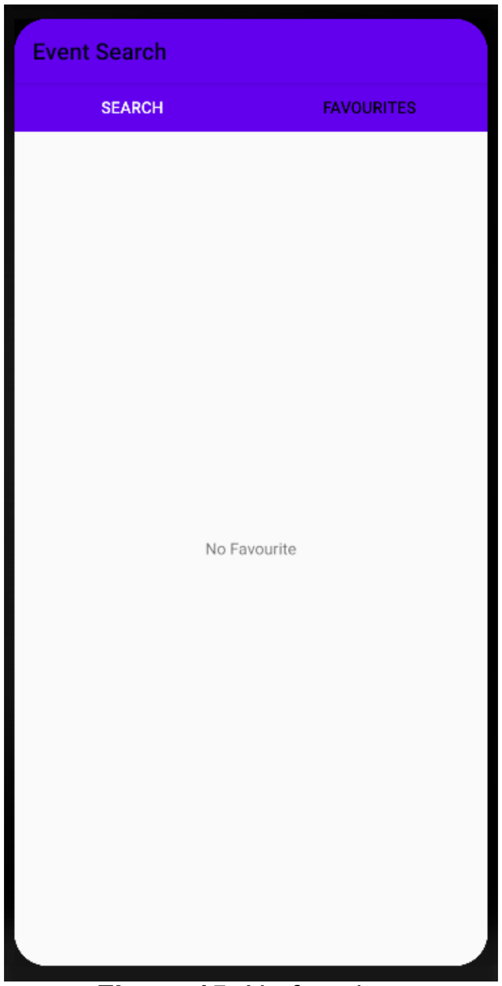


Figure 15: No favorites

5.5 Error handling

If no events are found given a keyword, a “no results” should be displayed, as shown in Figure 16. If for any reason an error occurs (no network, API failure, cannot get location etc.), an appropriate error messages should be displayed at the bottom of screen using a Toast.

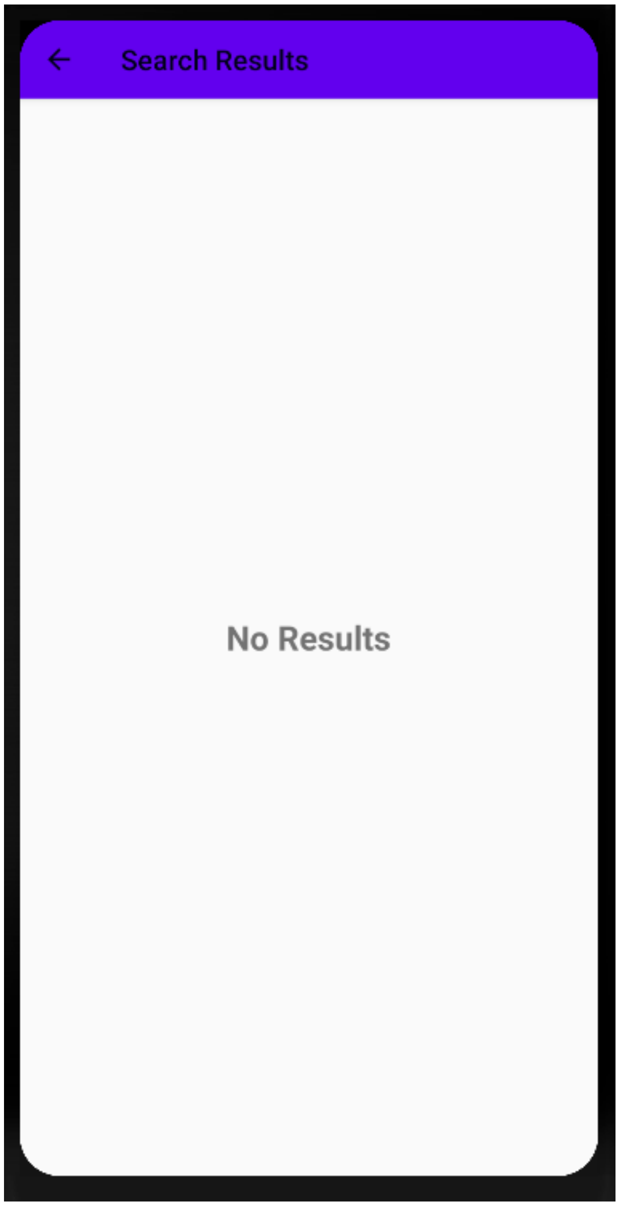


Figure 16: No search results

5.6 Additional

For things not specified in the document, grading guideline, or the video, you can make your own decisions. But keep in mind about the following points:

- Always display a proper message and don't crash if an error happens.
- Always display a loading message if the data is loading.
- You can only make HTTP requests to your backend Node.js on GAE and use the Google Map SDK for Android.
- All HTTP requests should be asynchronous and should not block the main UI thread. You can use third party libraries like Volley to achieve this in a simple manner.

6. Implementation Hints

6.1 Images

The images needed for the homework as provided as vector drawables and are available here:

http://csci571.com/hw/hw9/images/android/images.zip

6.2 Getting current location

For your location fetching code to work, you must request the permission from the user. You can read more about requesting permissions here: https://developer.android.com/training/permissions/requesting.html

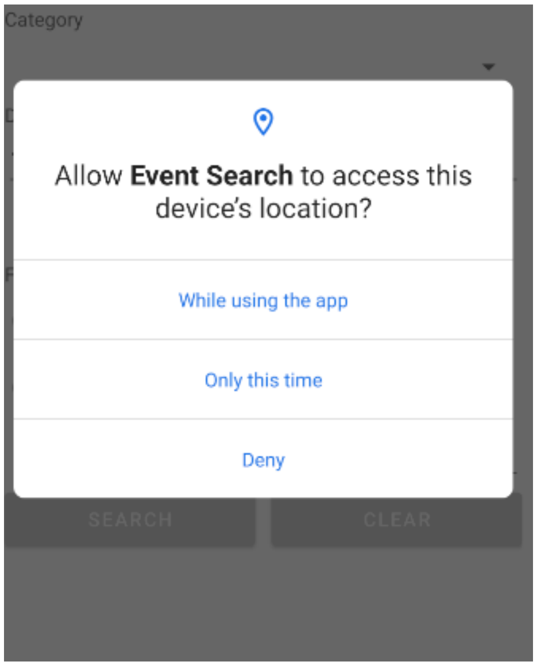


Figure 17: Requesting location permission

You may need to mock the location in your emulator. This can be done from the emulator settings.

6.3 Third party libraries

Almost all functionalities of the app can be implemented without using third party libraries, but sometimes using them can make your implementation much easier and quicker. Some libraries you may have to use are:

6.3.1 Google Play services

You will need this for various features like getting the current location and using Google Maps in your app.
You can learn about setting it up here:
https://developers.google.com/android/guides/setup

6.3.2 Volley HTTP requests

Volley can be helpful with asynchronously http request to load data. You can learn more about them here:

https://developer.android.com/training/volley/index.html

6.3.3 Picasso

Picasso is a powerful image downloading and caching library for Android.
http://square.github.io/picasso/

6.3.4 Glide

Glide is also powerful image downloading and caching library for Android. It is similar to Picasso.
https://bumptech.github.io/glide/

7. What to Upload to GiHub Classroom

You should also ZIP all your source code (the java/ and res/ directories excluding the vector drawables that we provide to you) and submit the resulting ZIP file by the end of the demo day.

You will have to submit a video of your assignment demo. Details for that will be sent separately.

****IMPORTANT****

All videos and grading guidelines are part of the homework description. All discussions and explanations on Piazza related to this homework are part of the homework description and will be accounted into grading. So please review all Piazza threads before finishing the assignment.