
GitHub Username: mtzhisham

EventBoard

Description

EventBoard is an app for displaying users nearby events using Eventbrite API on Mashape.com based on their location or a location they choose, any relevant information for the event will be shown and let the user favorite events and display them in a stackview widget

Intended User

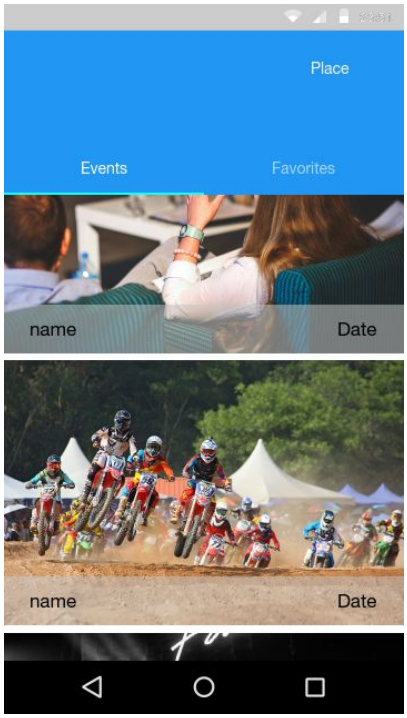
EventBoard is for anyone who is interested in knowing the events and concerts happening nearby them and keep track of their favorite events

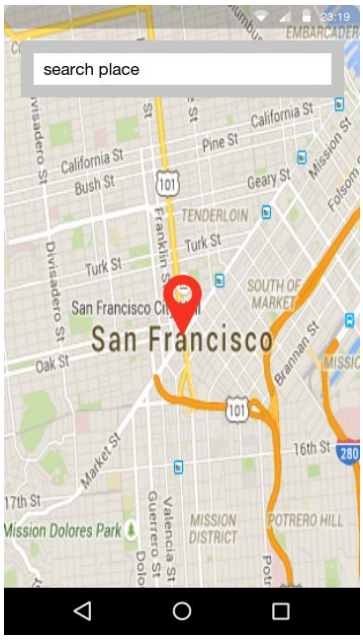
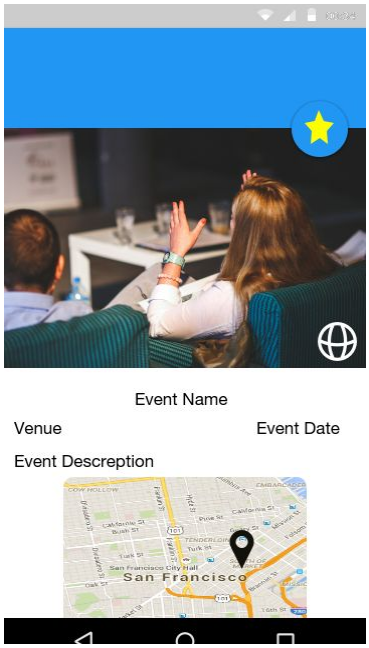
Features

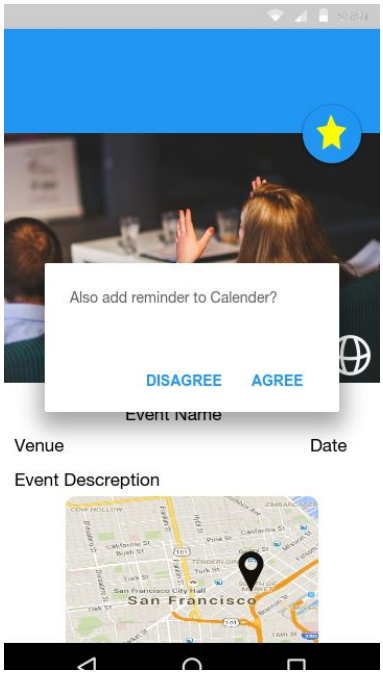
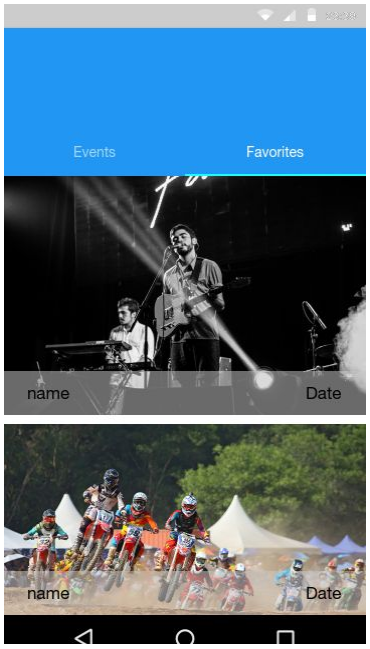
List the main features of your app. For example:

- View all nearby events
- View event information
- Share event with friends
- Chose user location on the map
- See the event location on the map
- Favorite events
- Add date to google calendar
- See favorite events in the widget

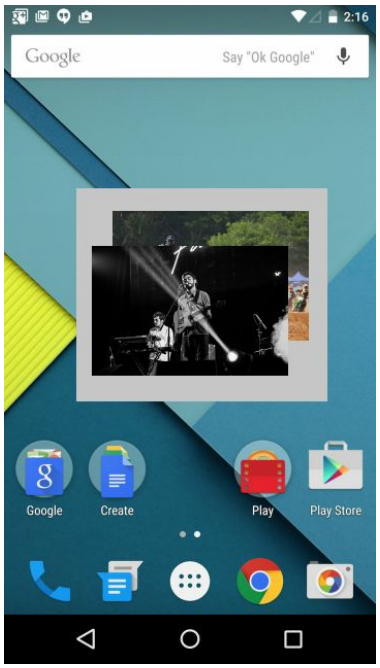
User Interface Mocks

Main Screen 	On the app main screen user can see the a list of the Events with name and date on the event image and a place picker on the toolbar

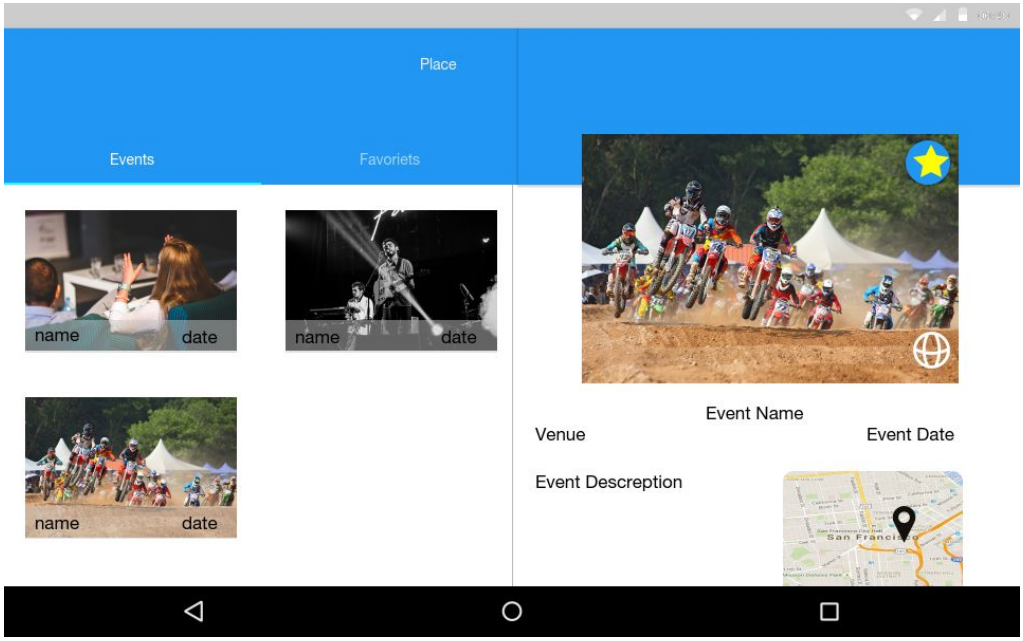
Place Picker 	Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…] Provide descriptive text for each screen
Detail Screen 	When the user taps on an event he/she sees the event details screen and can favorite, share and view the venue place on the map

Remainder Dialog 	If the user taps the favorite button a dialog asks whether to add a calendar reminder or not if the user agrees, an intent to the default calendar launched add event functionality launched with the event info
Favorite List 	On the favorite screen the user can see his favorite events

Widget	The app will provide a stackview widget for the user to view the favorited events



Tablet Screen



On tablets the app will utilize a two pane view

Key Considerations

How will your app handle data persistence?

A sqlite database and a content provider will be created to store the events data after fetching it Also it will store the favorite events as special rows, upon updating the events only events with ids that's not presented in the table will be added, and there will be separate tables for different location entries

Describe any corner cases in the UX.

UI corner cases will be dealt with during development like endless scroll to view all the events on the DB and preset event list position

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

- ajaysahani/EasyDatabase for creating a content provider and database
- Retrofit to get JSON source from Eventbrite endpoint
- LoganSquare for parsing JSON data fetched
- Icepick for presenting activity state without need to do DB query
- Glide for downloading and caching event image
- AVLoading indicator view for loading animation

Describe how you will implement Google Play Services.

Maps for letting the user choose a location and displaying the event location
Admob for displaying banner ads at the event detail screen

Next Steps: Required Tasks

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

- Create android studio project for the app
- Configure the Eventbrite api and crete application for the app and add Eventbrite API
- Add all the required dependencies for the 3rd party APIS

Task 2: Testing Eventbrite API

Testing the API endpoints and see if it returns the required fields correctly

- Implement Retrofit network call
- Implement LoganSquare parsing
- Make sure it's a valid response with no errors

Task 3: Add A/Sync functionality

Implement background operation for data retrieval

- Use syncAdapter for regularly update the events upon app opening
- Use AsyncTask for handling on demand Event data fetching if the user change the location

Task 4: Implement Data persistence support

Create the database and the content provider

- Create the Events Database
- Create the Content Provider
- Test the Cursor loaders and adapters
- Implement basic CRUD operations from the data retrieved from task 2&3

Task 5: Implement the UI

Implement the UI for all the views with a 2 pane tablet layout and material design guidelines like coordinate layout and collapsing toolbar layout in consideration

- Create a viewpager layout for the events list screen
- Create the events list and implement an endless scroll technique
- Create Event details screen
- Create Favorites Event screen

Task 6: Implement Event specific behavior

In the event detail screen the user can tab on a star to chose from the following

- Only add an event to the favorites
- Add an event to the favorites and add a reminder to the Calendar

Task 7: Implement favorite Events Widget

The app will provide a stack widget for the user to scroll through his/her favorite widget cards

Task 8: add google play service integration

Add google play services or firebase functionality

- Add Maps integration
- Add a location picker with custom map view let the user search or place a marker in the event detail screen
- Add admob integration and show a banner add on the detail screen

Task 9: Accessibility

Implement accessibility and localization

- Add RTL support
- Translate all the English strings to Arabic

Task 10: building

Describe the next task. List the subtasks. For example:

- handle Signing configuration
- Add keystore add passwords
- Test installRelese Gradle task