How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: detrraxic

eventomic

Description

Eventomic app helps users discover local events, get recommendations based on their preference, and sends notifications about upcoming events. It is a handy app for users that wish to quickly browse events in their proximity when they are on the go and to remind them of the events that they have saved as favorite.

Intended User

Anybody interested in attending events such as concerts, music festival, and performing arts.

Features

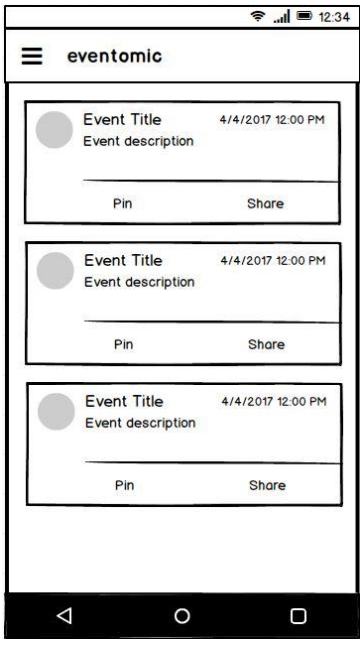
App will primarily enable the user to:

- browse local events
- save favorite events and receive notifications about incoming events
- shares events with others
- receive recommendations based on their category preference

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.

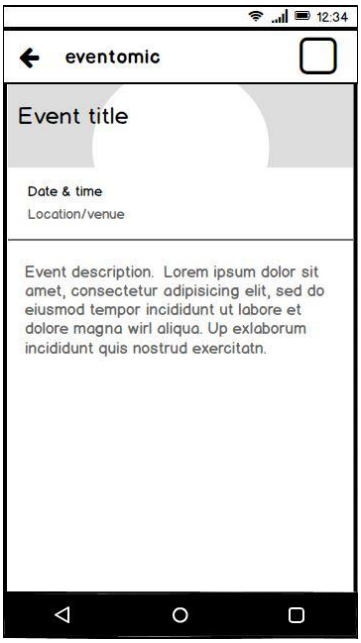
Screen 1



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]

Main screen will contain the list of events in user's area (based on GPS). Clicking on the cards displays event details page. Cards contain event image, title, location, date and time, and options to save and share the event.

Screen 2



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]

Details page contains back button to the main activity. Details page consists of further event detail with the description text and similar events.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Data will be pulled from eventful API. I will build a Content Provider for storing offline event data. Settings will be saved using the SharedPreferences.

Describe any corner cases in the UX.

App will only be available in vertical orientation.

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

For image loading and caching I will use Glide. For view injections I will user Butter knife and AwesomeSplash for splash screen. I am familiar with these libraries and they are very easy to implement and they reduce boilerplate code.

Describe how you will implement Google Play Services.

App will use AdMob for displaying interstitial test ads and Google Analytics.

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

Project dependencies – setup Gradle by importing required libraries:

- Android Design Support Library
- CardView
- RecyclerView
- AdMob
- Google Analytics
- Butterknife
- AwesomeSplash

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

Create UI:

- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for Settings
- Include support for accessibility and enable RTL layout switching
- Implement shared element transitions and parallax scrolling

Task 3: Data persistence

Build data persistance:

- Create SQLite Database for storing event data
- Create ContentProvider for accessing this data
- Create Model classes for stored data
- Create data validation methods

Task 4: Sync data

- Implement CursorLoader
- Implement SyncAdapter

Task 5: Remaining tasks

- Create widget that displays saved events
- Implement notifications
- Implement AdMob and Analytics
- Implement splash screen with app logo

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"