

**GitHub Username:** gkap720

# Cricket

## Description

In a new place and just want to go and hear some music? Download Cricket!

Cricket automatically finds concerts at popular venues near you to check out, letting you filter for genre to match your tastes. You can even choose to be updated daily about concerts at your current location so that you never miss a show!

This app is powered by JamBase.

## Intended User

This app is for travelers, but also for music lovers who want to be constantly updated about shows in their area.

## Features

Stores user preferences.

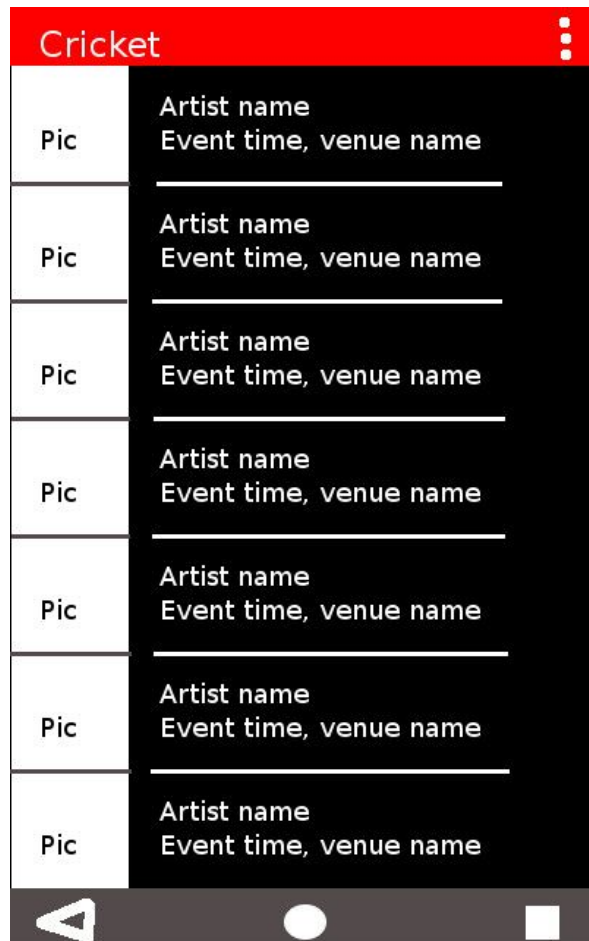
Uses user location and preferences to find concerts.

Displays nearby concerts and venues in an attractive and easy to read format.

## User Interface Mocks

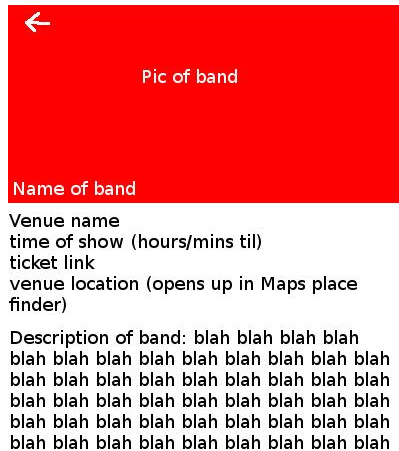
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

## Screen 1



This is the main screen which will be populated by concerts happening near the user or near the user's preferred location in settings.

## Screen 2



This is an example of the detail page for a specific concert. The top third of the screen will be taken up by a large photo of the band with the band's name overlayed on it. This will be a collapsible view. Underneath the app bar will be info on the venue and tickets as well as information about the band playing. Clicking on the venue's location will bring up the location in google maps.

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

## Key Considerations

### How will your app handle data persistence?

I will make a content provider using data taken from the JamBase archive. This data will be preserved daily and then redone as new data comes in.

### Describe any corner cases in the UX.

Since there are only two main screens in the app, the user will simply use the back button to navigate between them. The main activity will have a simple scrollview which can be scrolled until there are no more entries. The detail activity will also have a scrollview that will change in size depending on the content.

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

I'll be using Picasso to handle images.

**Describe how you will implement Google Play Services.**

I'll be using Google location services in order to provide a relevant experience to the user. I'll also use Google identity services in order to provide settings consistency with a Google account.

## 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

- Configure setup (classes of folders to organize aspects of the app)
- Add all dependencies in gradle build
- Configure a main theme (choose colors etc.)

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for SettingsActivity

### Task 3: Make Calls to API successfully and populate listviews

- Test JamBase API and find out data structure
- Parse data correctly
- Populate listviews
- Make place to store API\_Key in gradle build

### Task 4: Construct the content provider

- Make an SQLite database with appropriate tables
- Make a syncadapter to sync data at a set interval so it is relevant/current
- Make sure syncadapter in manifest.xml
- Make sure that data is being stored in database correctly
- Make sure content provider can be accessed easily/correctly

### Task 5: Implement Google Play Services

- Implement Location services so that the user's current location affects concert listings
- Implement Identity services so that the user can have a unique account and can store their settings
- Add services to manifest.xml

### Task 5: Implement notifications

- Implement notifications that tell the user how many concerts there are today in their selected genre
- This notification could be turned on/off and would appear in the beginning of the day (also customizable)
- Make changes in manifest.xml

### Task 5: Make app accessible

- Add content descriptions
- Add support for directional pad
- Make sure text is in sp

---

### 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**"