

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: R-K-1.

Lord of The Trivia.

Description

Make any commute or get-together a lot more fun by playing one or many memorable trivia sets.

Intended User

Public transportation users and friends/families.

Features

- Scrolls through trivia question sets on a variety of subject.
- Pick and play any set of questions.
- View score and correct answers.
- Create Account or Login against Firebase.
- Reset password
- View your best scores and share them with all app users.

- View all app users best scores.

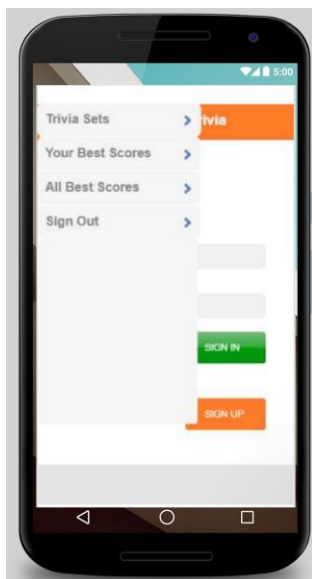
User Interface Mocks

Login



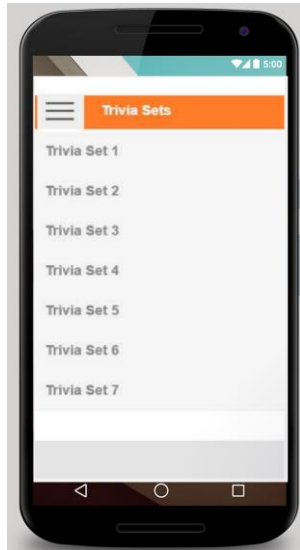
This is the home screen allowing users to create an account or login.

OverLay Menu



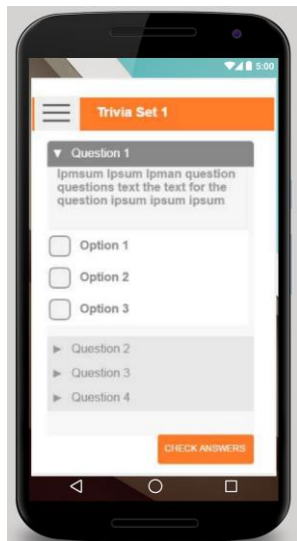
Clicking the icon within the app bar displays an overlay menu allowing users to navigate to the different sections of the app.

Trivia Sets



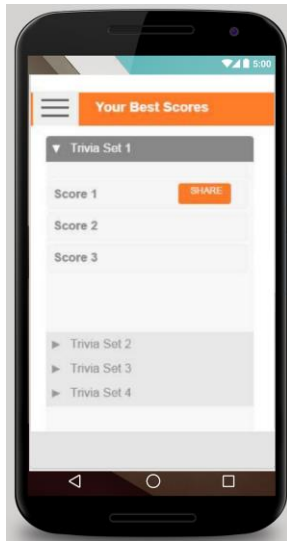
One of the keys screens of the app allowing users to scroll through the list of available trivia sets and selecting which one they want to play.

Trivia Set



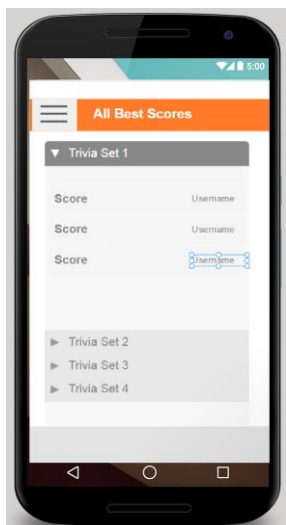
Once users select a set to play, they are lead to a screen with questions and checkable options. Once they have answered the questions they can chose to view the answers.

Your Best Scores



On this screen, users can view how they have scored on each trivia set and chose to share their highest score per set.

All Best Scores



This is the screen where the best scores of shared by all app users can be viewed.

Key Considerations

How will your app handle data persistence?

Questions, best scores and images will be stored in firebase.

The first time the application is started, the resources will be retrieved and stored in a local relational database and internal storage.

A Content provider will allow the app UI to communicate with the local relational database.

Describe any corner cases in the UX.

All selected answers for a currently played trivia set will be saved in Shared Preferences as a pipe delimited string. Therefore, if users rotate the screen while playing a trivia set or come back to it after navigating to another screen/app, they answers will still be selected.

The same logic will be followed for all accordion item on all screens.

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

Picasso for saving images in the Internal storage and loading them in app UI.

Describe how you will implement Google Play Services.

Ad Banners will be displayed at the bottom of each screen.

Interstitial Ads will be displayed before launching the trivia set screen.

Trivia sets, images and scores will be stored in Firebase.

Next Steps: Required Tasks

Task 1: Project Setup

Tasks to perform for project setup:

- Configuring Firebase
- Creating database for app in Firebase console.
- Adding Firebase to Graddle.
- Configuring Picasso.
- Adding Picasso to Graddle.

Task 2: Implement UI for Each Activity and Fragment

Implies:

1. Building UI for User Account Creation/Authentication Screen.
2. Building UI for Trivia Sets List Screen.
3. Building UI for Playing Trivia Set Screen.
4. Building UI for Current User Best Scores.
5. Building UI for All Users Best Scores.

Task 2: Mock up Test Data in Firebase Console

Including:

1. Questions sets.
2. User accounts.
3. Top scores.

Task 3: Prepare Communication With Firebase

Divided in the following subtasks:

1. Developing User Creation code.
2. Authoring User Authentication code.
3. Building question sets and top scores retrieval from firebase code.

Task 4: Design Local SQL Database

Entails the following:

1. Creating an SQL Database
2. Creating tables holding trivia questions, scores and image paths.

Task 5: Produce Content Provider

Code logic for:

1. Persisting questions and scores retrieved from Firebase to SQL database.
2. Saving authenticated user scores into SQL database.
3. Retrieving questions and scores from SQL database.

Task 6: Implement Update Checker

Create task/job which, every 24 hours, will check if trivia question sets and top scores have been updated and then download those updates.