

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1 & 2](#)

[Screen 3, 4 & 5](#)

[Screen 6](#)

[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: Setup](#)

[Task 2: Database](#)

[Task 3: Network](#)

[Task 4: User interface](#)

[Task 5: Pull data](#)

[Task 6: Handle Error Cases](#)

[Task 7: Implement Google Play Services](#)

[Task 8: Improve design](#)

[Task 9: Generate and Deploy](#)

GitHub Username: sigma-s

Movie Mania

Description

Are you a movie buff? Have you spent countless hours watching your favorite movies and debating with friends on the best lines from movies? Do you like movies so much that you use famous dialogues in your everyday conversation/ Whatsapp chat with friends? Do you know all the films in Netflix? Is your movie hard disk collection bigger than shoe collection of your girlfriend? Do you feel you can't remember the answers in your exams but can recite movie dialogues verbatim? Do you wish there was an exam over movie dialogues that you could have

easily nailed? Well here is a chance to prove your mania about movies. A quiz to prove your ninja like skill in movies.

There are many apps which do the same for Hollywood movies but hardly any for Bollywood movies. This app combines the best of both so you can enjoy both English and Hindi dialogues.

In this quiz you will be asked ten questions each round. Each questions is a famous line or dialogue from a movie. Choose the right film from the given answers. The faster you are the higher your score at the end of each round.

Intended User

Movie enthusiasts - all age groups

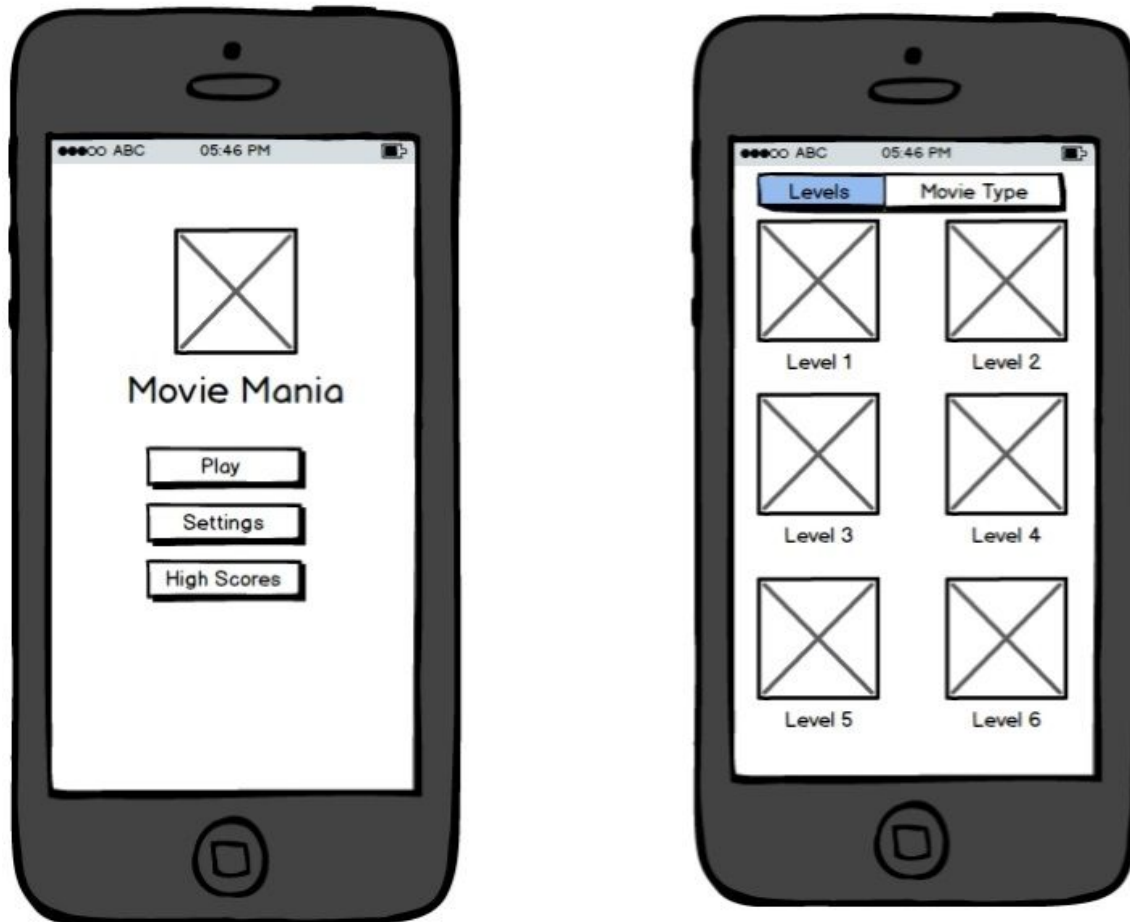
Features

Main features of the app:

- Retrieve movie quotes and movies from a database
- Allow selection between Hollywood and Bollywood movies
- Create a question based on the data retrieved above
- Calculate points based on the correctness of the answer above
- Retrieve a movie poster/video related to the question asked
- Allow the poster/video to be shared via social media
- Save the current state/level of the user in a database
- Show multiple levels and the points required to unlock each level

User Interface Mocks

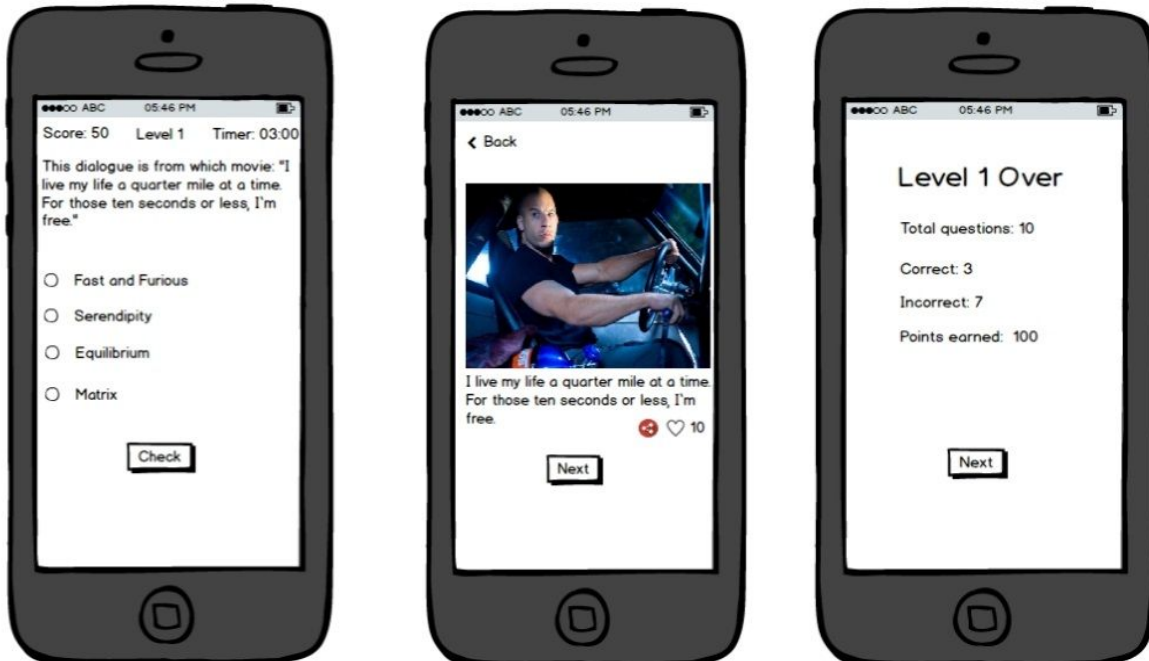
Screen 1 & 2



Screen 1 is a basic introduction screen for the user. It has 3 options - Play, Settings and High Scores and the name and logo of the app.

Screen 2 consists of 2 tabs - levels and movie type. There are 2 movie types - Hollywood and Bollywood. Within each type there are 10 levels. Each lower level should be cleared before advancing on to the next level.

Screen 3, 4 & 5



Screen 3 is the main quiz screen. It consists of a movie dialogue as a question and 4 movies as options for answers in the form of radio button. Each question also has a 3 minute timer after which the quiz is considered unanswered. Each correct answer fetches 10 points whereas each incorrect/no answer fetches 0 points.

Screen 4 comes after checking the answer to the question in screen 3. It contains a poster of the movie whose dialogue is part of the quiz in screen 3. You can share the image or favorite it. Pressing on Next takes you to screen 3 for the next question in the quiz.

Screen 5 comes after 10 questions are answered in any particular level and the level gets over. It gives a summary of the performance in that particular level. Pressing on Next takes you to Screen 2 above.

Screen 6



Screen 6 is the widget component of the app. It basically consists of a shortened version of the main quiz screen i.e. Screen 3. A user can think about the question at hand while on the widget and then click on the widget to go to Screen 3 for answering the question.

Key Considerations

How will your app handle data persistence?

The app will have a new Content Provider to store the score and the level completion of the user at any point of time. Content Provider will also be used to store the data fetches from theMovieDB API.

Describe any corner cases in the UX.

If the user quits at a certain question and comes back again to the app, he will be able to observe its saved state in the Screen 2 above in the respective level.

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

Picasso to handle loading and caching of images. It is a powerful image downloading and caching library for Android.

Retrofit for fetching question and answers asynchronously from theMovieDB API. It is a type-safe REST client for Android and Java which intelligently maps an API into a client interface using annotations.

Describe how you will implement Google Play Services.

Google play game services to allow a more competitive and social experience through the use of leaderboards and achievements

Google Admob for displaying ads in the app

Next Steps: Required Tasks

Task 1: Setup

- Configure libraries
- Choose an Architecture and implement skeleton Android Architecture Blueprints
- Create a debug and build variant
- Configure tools

Task 2: Database

- Database
- Content Provider
- Data models

Task 3: Network

- Data parsing
- Data syncing service

Task 4: User interface

- String and image resources
- UI for main activity
- UI for detail activities
- Gestures, transition, ...

Task 5: Pull data

- CursorLoader
- Ask for data

Task 6: Handle Error Cases

- Empty views
- Error message

Task 7: Implement Google Play Services

- Google Play Game
- Google AdMob

Task 8: Improve design

- Landscape mode
- Large screen
- Tablets
- Transitional screen

Task 9: Generate and Deploy

- Generate app flavor keys
- Create Google Play Account
- Create APK
- Create Google Play image and text
- Push the APK on the store