GitHub Username: Paul Gureghian

Capstone_Project

Description

My app helps new coders learn by watching professional coders code live via streaming. They can choose from many languages including Java, Python, Ruby.

Intended User

New coders wanting to learn something about a language, experienced coders needing a refresher, experienced coders wanting to network with other coders.

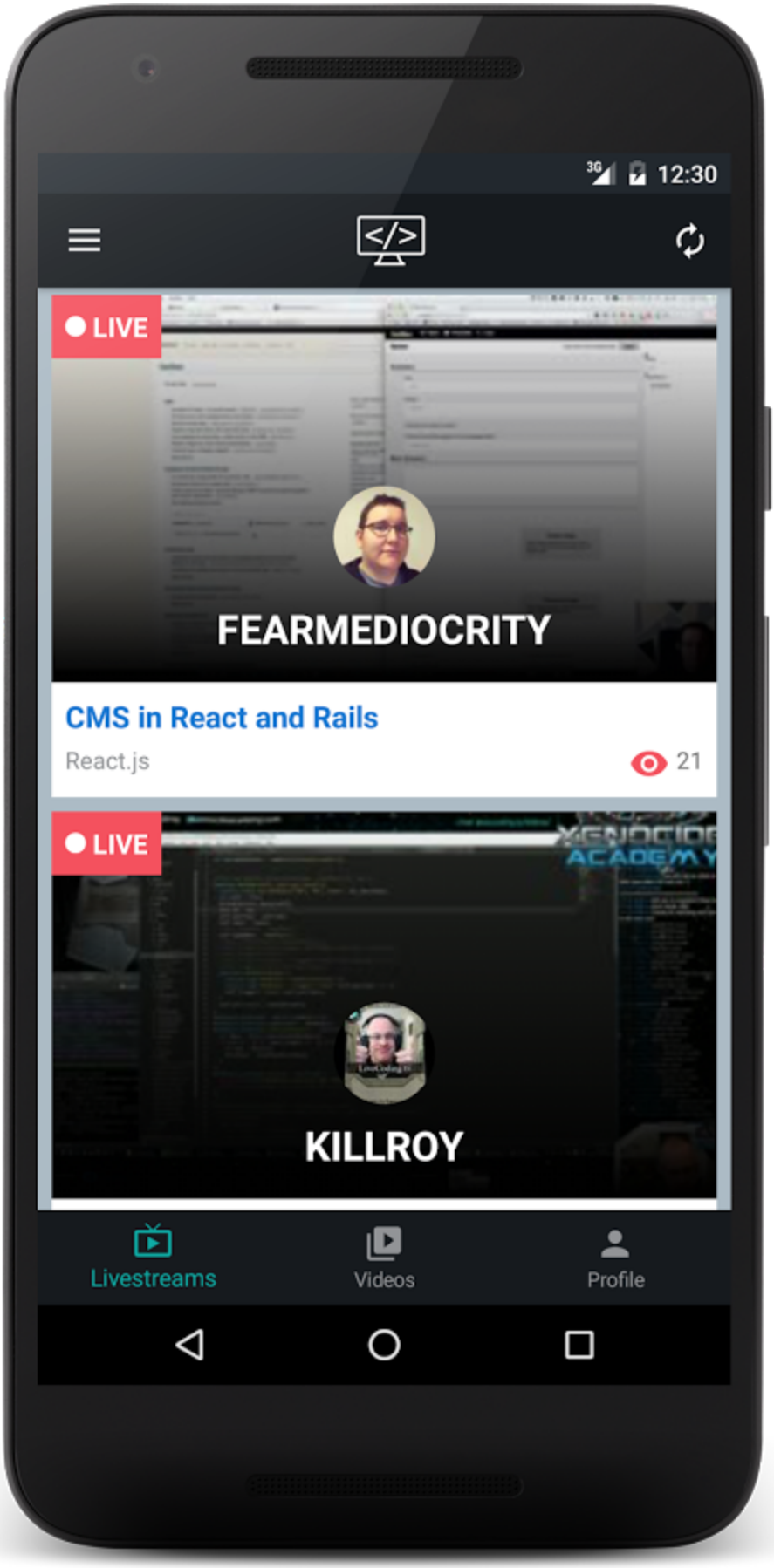
Features

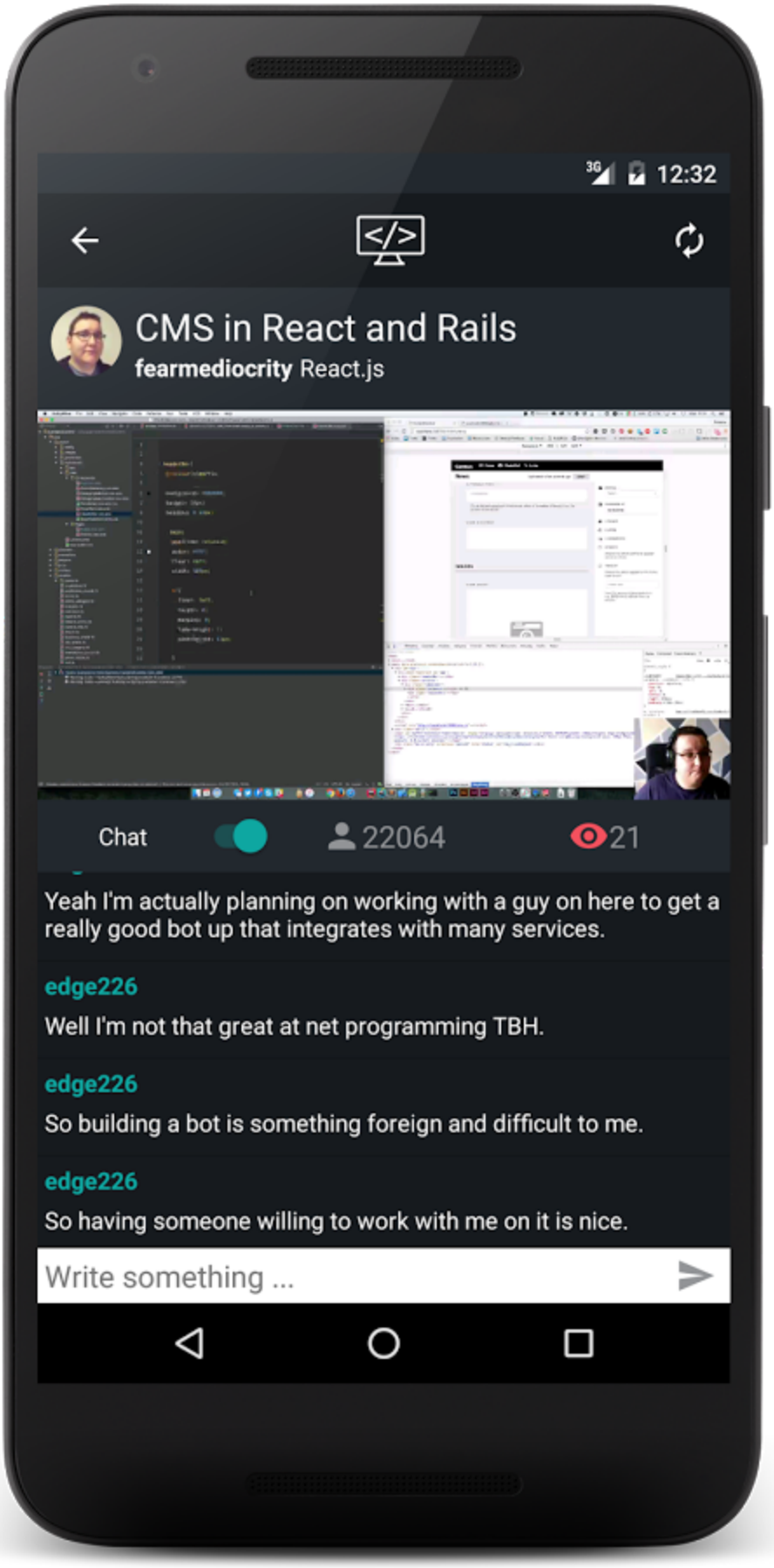
List the main features of your app. For example:

- Many programming languages to choose from.
- Chat with other coders.
- Read profile info of the coder you are watching.
- Choose language (English, French, German)

Screen 1.

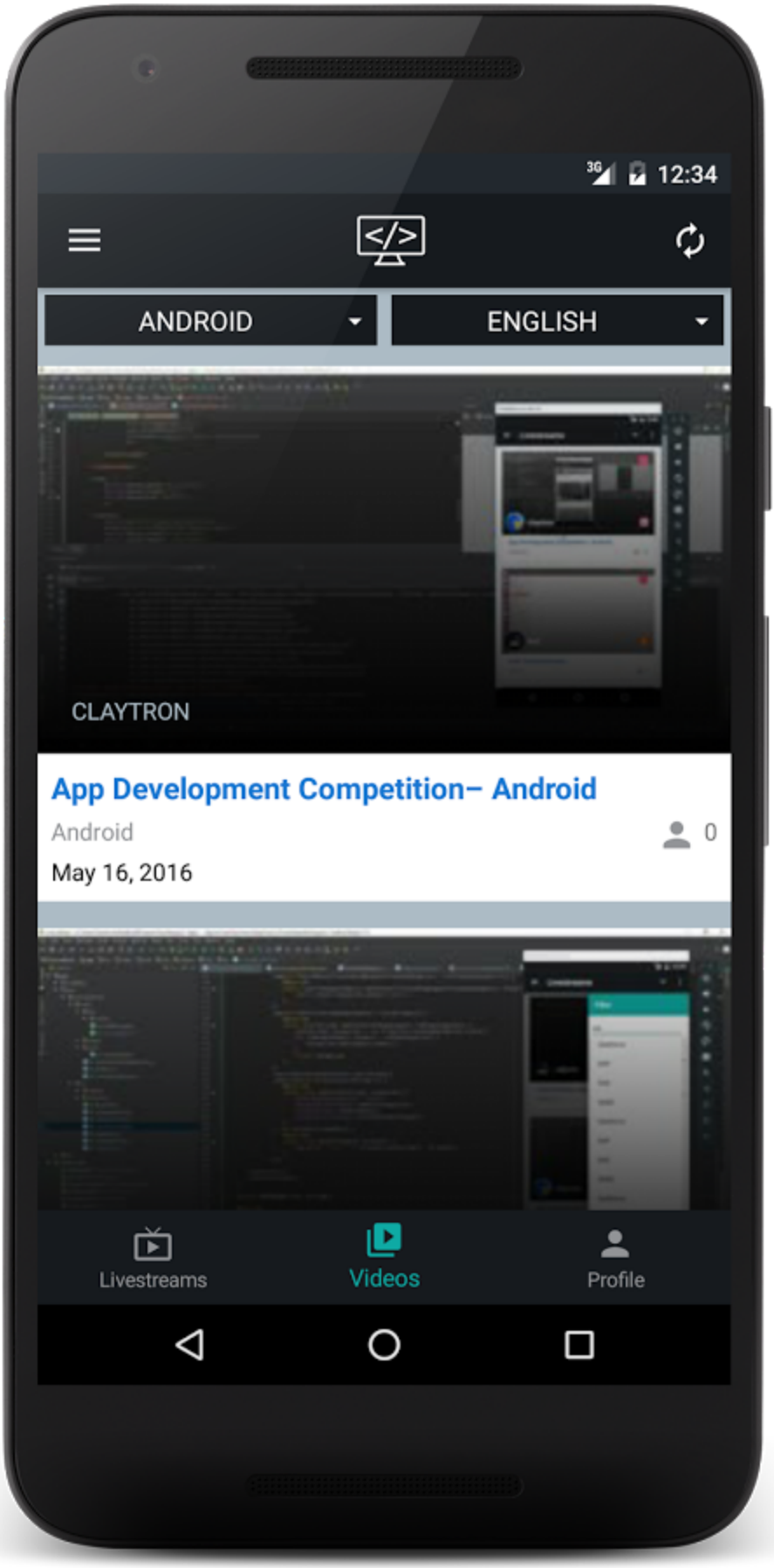
A gridview with livestreams.


12:30
LIVE
FEARMEDIOCRITY
CMS in React and Rails
React.js
21
LIVE
KILLROY
Livestreams
Videos
Profile


12:32
CMS in React and Rails
fearmediocrity React.js
Chat
22064
21
Yeah I'm actually planning on working with a guy on here to get a really good bot up that integrates with many services.
edge226
Well I'm not that great at net programming TBH.
edge226
So building a bot is something foreign and difficult to me.
edge226
So having someone willing to work with me on it is nice.
Write something ...

Screen 2.

Screen 2 has the actual coding session and a chat function.


12:34
ANDROID
ENGLISH
CLAYTRON
App Development Competition– Android
Android
0
May 16, 2016
Livestreams
Videos
Profile

Screen 3

Screen 3 has a video search function for programming languages and languages.

Key Considerations

How will your app handle data persistence?

Will use Firebase.

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

Picasso and / or Glide to handle the loading and caching of images. Design support library for the edit text field. Retrofit for the API calls. Because i've used them in the other projects.

Describe how you will implement Google Play Services.

Google Analytics to track usage data.
Google Location and Context (Awareness API) to enable my app
To react to the user's current situation.

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

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

You may want to list the subtasks. For example:

- Add library dependencies to gradle
- Create a github repo
- Create the project in Android Studio
- Clone the repo

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Add three Java class activities.
- Add three related XML layout files.

Task 3: Your Next Task

- Create Retrofit interfaces for the endpoints.
- Implement a custom content provider to provide
- Custom search suggestions

Task 4: Your Next Task

- Create a POJO class which references the JSON items I want to parse.
- It will implement the Parcelable class.
- Use remote views API to create a widget.

Task 5: Your Next Task

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

- Create a free debug variant.
- The app will have a standard Android content provider backed by a sqlite database.
- AsyncTaskLoader with a search interface.