
GitHub Username: KKdev13

AndroidRefresher

Description

Been in a long journey learning Andorid programming?! Concerned about forgetting all what you learned about intents, content providers, libraries, activities, fragments, and so on and on and on ?! I hear you! AndroidRefresher is the right place for you. It's simply an app that helps you never forget how to create an app! AndroidRefresher tests your knowledge and understanding of Android development in a simple, fun quiz format of a total 30 questions. Hang in there; it's just a 1.1 release! More questions will be added in future releases and updates, so stay tuned! The app is built using the Android Studio platform and Java language.

Intended User

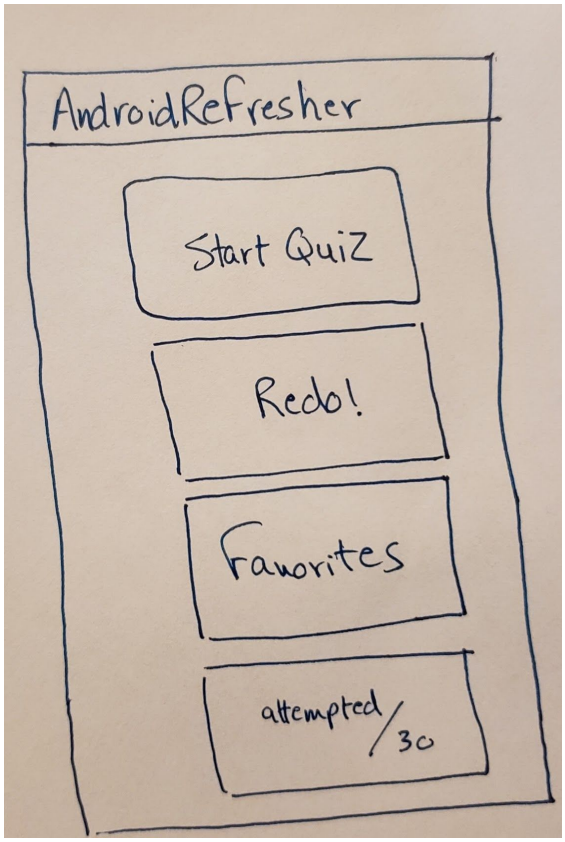
The app is intended to the Android and Java-aware users and developers who want to brush upon and test their android knowledge, and improve their skills.

Features

- Works offline
- RTL layout support
- Free and paid versions
- Includes a widget for easy access
- Total of 30 scored questions
- Repeat the incorrectly answered question
- Mark your favorite questions

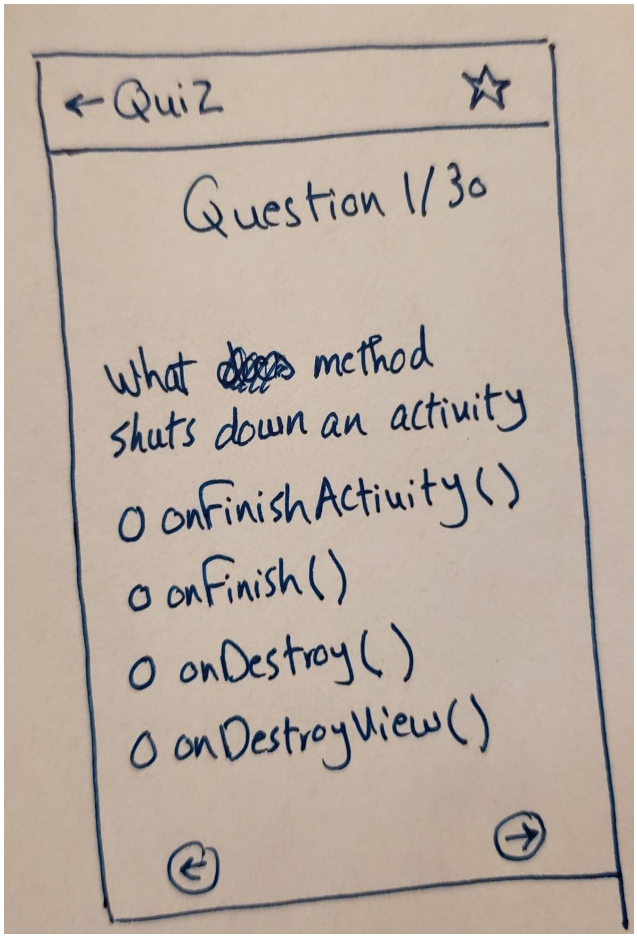
User Interface Mocks

Screen 1

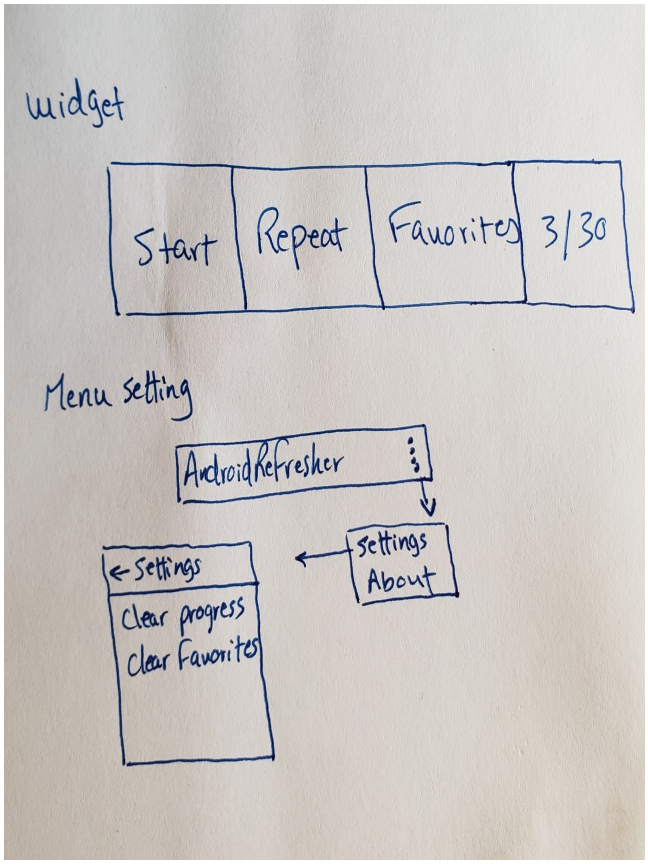


Main activity quiz UI

Screen 2



Quiz's questions preliminary layout



A widget and menu settings items

Key Considerations

How will your app handle data persistence?

The app will be implemented using a simple SqLite database and content provider, keeping track of questions, incorrect answers, and favorites. It will use recycler views, loaders, and adapters to display and update user views. Images and string constants are properly maintained in their corresponding folder structure in Android Studio.

Describe any edge or corner cases in the UX.

No need to answer all questions at once; the app saves your correct answers and continues form there! For example, if you only answer four quiz questions correctly in your first trial, next time you run the app, it will show your correct attempts as 4/30, so you can only try the new and incorrectly answered questions. The app will make sure the user answers a question first before navigating to next or previous questions.

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

- The app will use stable release versions of Gradle and Android Studio!
- ButterKnife 8.1 (for data binding)
- SqlLite (for storing quiz questions and other database operations)
- Google analytics and ads 10.2 (for ads and analytics!)
- AppComapt 26.1 (for older versions support)
- RecyclerView (for fetching and viewing data)
- Espresso 3.1 (for unit testing)

Describe how you will implement Google Play Services or other external services.

- Google Analytics (to provide more insight about how users are interacting with the app)
- Google Ads (for providing free and paid versions)

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Create project AndroidRefresher in Android studio and folder structure.
- Add necessary permissions and dependencies in the manifest file.
- Configure necessary libraries, if any.
- Make Github repository and submit project progress.

Task 2: Implement UI for Each Activity and Fragment

- Create layout files for each activity/fragment, design a proper UI that serves app purpose.
- Bind each layout to its appropriate Java file.

Task 3: Database

- Create a proper database file with required tables.
- Add a contract class and implement a content provider.

Task 4: Verify database operations

- Add proper CRUD methods and test their functionalities.
- Keep track of questions, correct and incorrect answers.

Task 5: Adding features and feedback

- Adding a user friendly navigation features.
- Providing user feedback of the right and wrong answers
- Implementing an add favorite functionality, for the user to save his or her preferred questions.

Task 6: Google play services

- Implement Google Analytics.
- Implement Google Ads.

Task 6: Adding widget

- Adding a widget to the app using an Intent service.