
GitHub Username: mrubin888

Quick Prog

Description

One of the most effective ways for people to jumpstart their coding education is by working through projects. Projects allow coders of all levels to expand their skill range and comfort level by getting hands-on experience working through real world problems. Quick Prog is a peer-mentorship coding app that allows users to post project prompts, as well as to complete the prompts of others. This allows for accelerated learning without heavy reliance on 1 on 1 coaching or formal education. Users learn what they need to know to solve real world problems.

Intended User

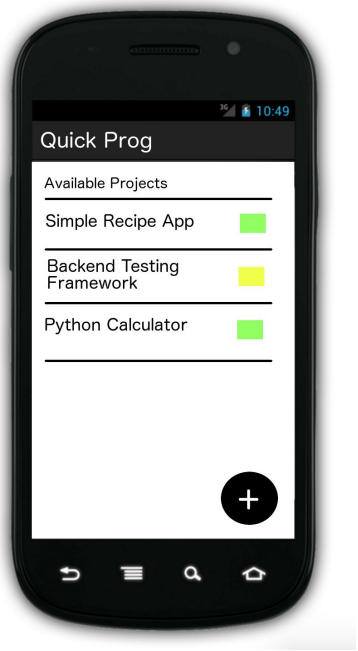
This is an app for computer programmers of all levels, but especially for beginners looking for some early direction.

Features

- Allows users to post project prompts
- Allows users to view project prompts
- Allows users to track their progress on projects by marking them as in progress and complete

User Interface Mocks

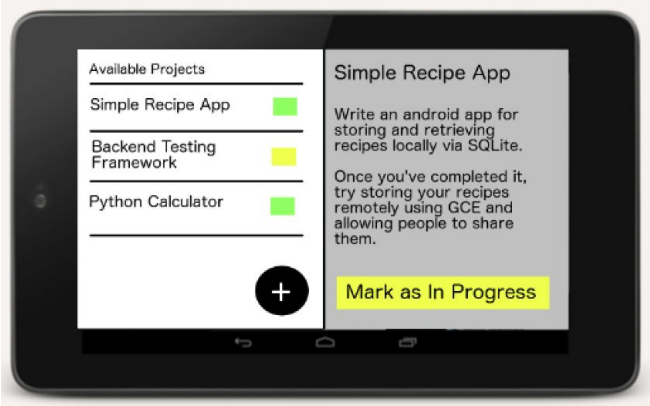
List Screen



Detail Screen

Create Project Screen

Tablet Master-Detail Screen



Key Considerations

How will your app handle data persistence?

It will pull on app start from a Google Cloud Endpoint with a SQL-based server (MySQL probably), and cache results on a local sqlite database with a content provider.

Describe any corner cases in the UX.

There are none.

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

There are none required by my initial design, but it may make sense to utilize some as I build out my functionality.

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: Setup Backend

- Setup a MySQL database with a table for projects (Either use an ORM or manually save the schema)
- Setup a Google Cloud Endpoint with basic create and read routes
- Connect the routes to the database

Task 2: Setup UI

- Build UI for main activity
    - Build UI for list and detail fragments
- Build UI for new project activity

Task 3: Setup Local SQLite DB

- Setup contract for local SQLite DB
- Setup content provider for local SQLite DB
- Setup DBHelper for local SQLite DB

Task 4: Make Frontend Functional

- Connect new project activity to POST route of GCE
- Build async task for fetching list of projects from GCE and loading them into local SQLite
- Setup adapter to populate list fragment from results on async task complete (this should populate from local SQLite database)

Task 5: Generate Some Basic Content

- Through the app, add a few basic projects for proof of concept.