GitHub Username: erikcox

Daily Programmer Challenges

Description

Daily Programmer Challenges is an Android app that lets you browse programming challenges and see solutions from the dailyprogrammer subreddit.

Intended User

This is an app for programmers who want to challenge themselves.

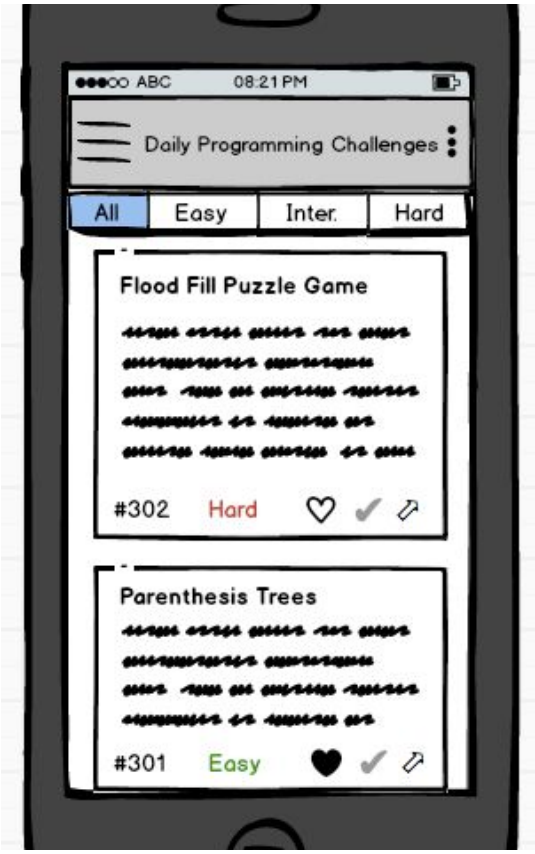
Features

- Pulls data from the Reddit api
- Formats that data in easy to read challenges and shows solutions
- Allows the user to mark challenges as favorites
- Allows the user to mark challenges as completed so they no longer show up in the feed

- Ability to sort challenges by newest / oldest
- Ability to sort challenges by difficulty
- Ability to share challenges through shareIntents

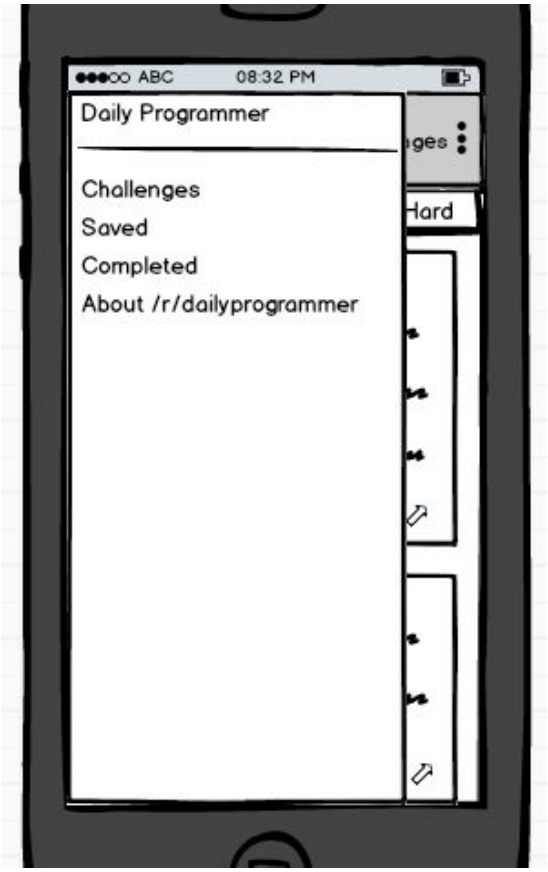
User Interface Mocks

Main screen



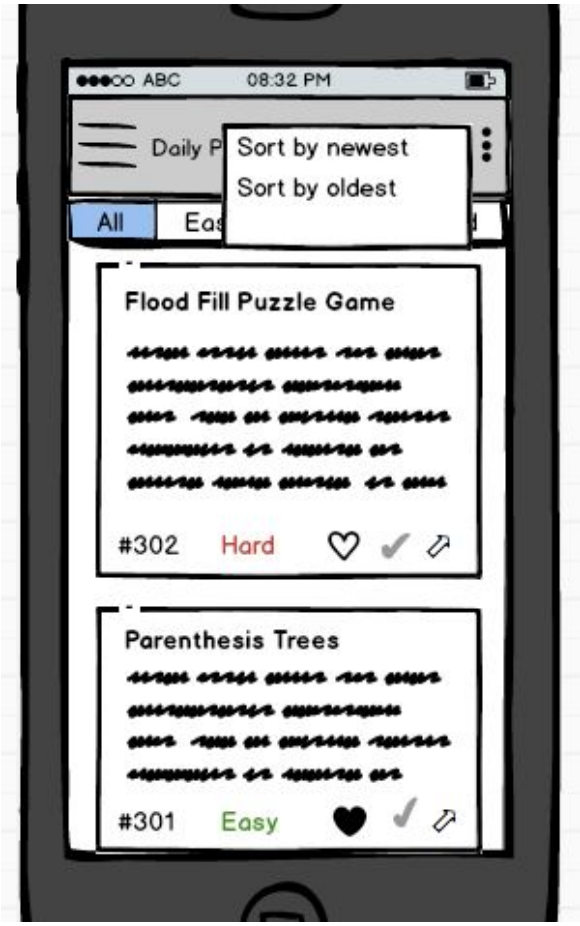
Main screen that shows a list of all the challenges with the ability to filter by difficulty

Navigation drawer



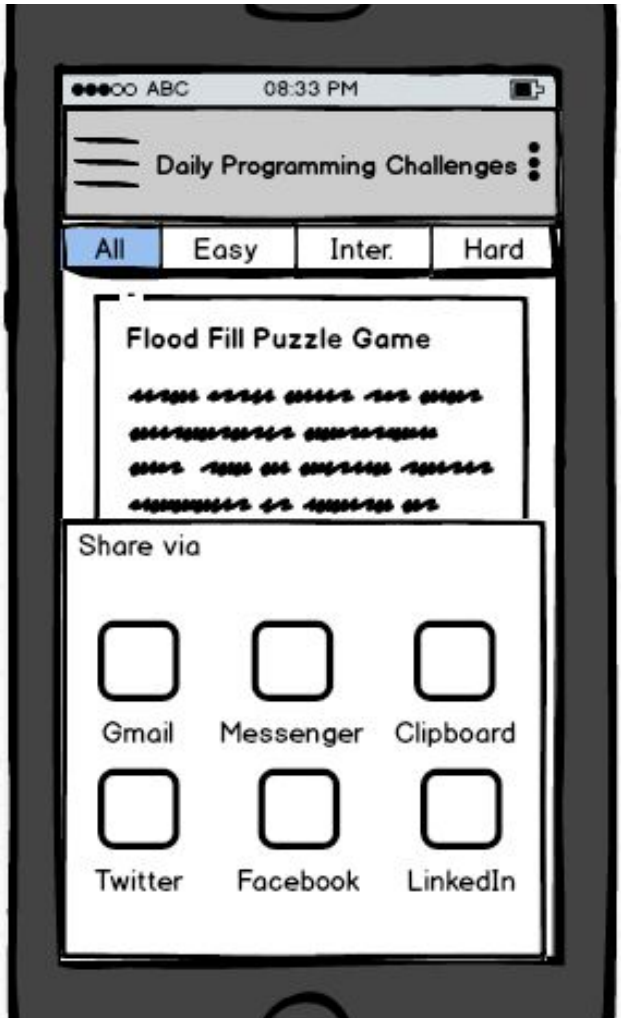
Android navigation drawer that lets you switch between all challenges, saved challenges, completes challenges, and an about page that gives more information about the programming challenge subreddit.

Sort options



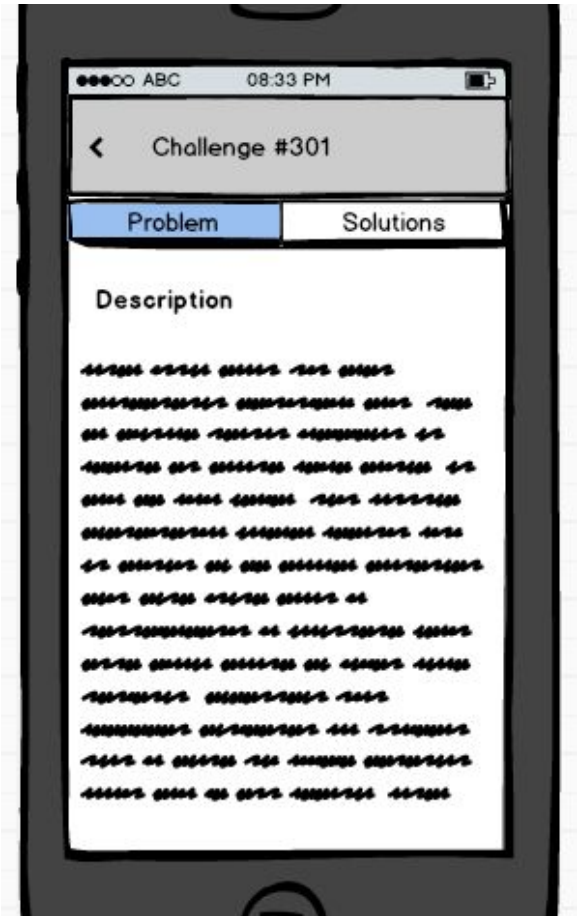
The settings menu allows you to sort the current screen by newest or oldest.

Sharing challenges



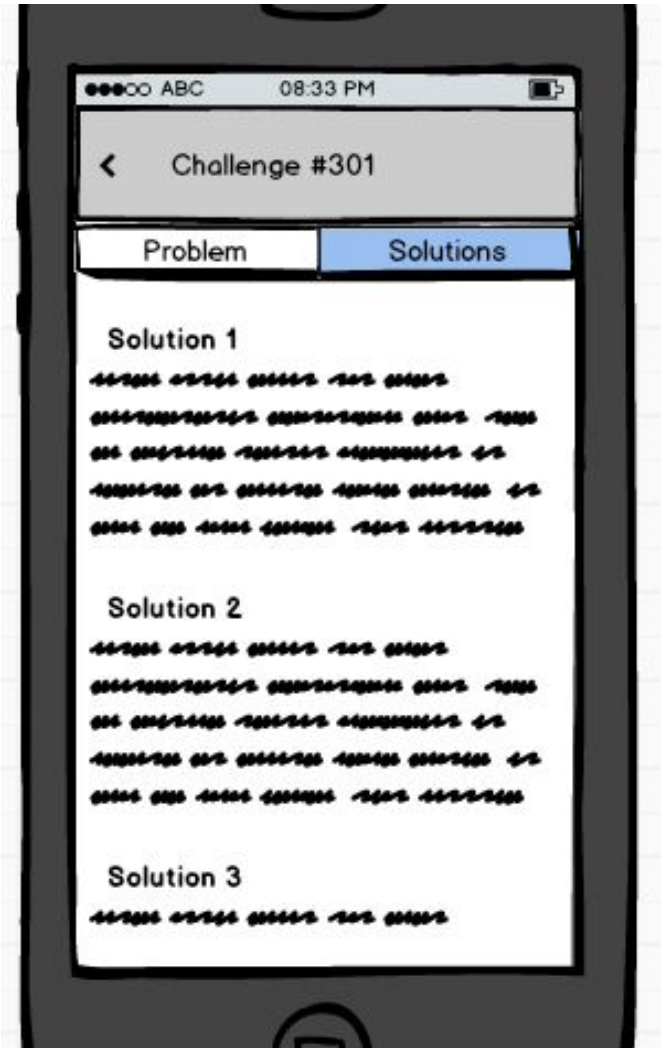
Each challenge has a share option that allows you to share a link to the current programming challenge via Android's sharedIntents to other applications, such as Gmail.

Challenge details



The main details page outlining the challenge and it's rubrik.

Challenge Solutions



The solutions screen shows the user solutions to the given challenge.

Saved challenges

This view shows you all of the challenges that the user has marked as saved.

Completed Challenges

This view shows you the challenges that the user has marked as completed. Those marked as completed will not be shown in the main challenges views.

About /r/dailyprogrammer

This view shows the user the description and instructions about the dailyprogrammer subreddit and the programming challenges.

Key Considerations

How will your app handle data persistence?

The app will handle data persistence via a content provider.

Describe any corner cases in the UX.

The sub-views will all have back arrows to return to their calling view, which will also work with the back button and the navigation drawer.

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

- JRAW for interfacing with the reddit api
- Retrofit for handling networking
- Timber for logging
- Firebase / Crashlytics for analytics
- Android support library
- Android design library

Describe how you will implement Google Play Services.

Google Analytics for tracking
Google Drive for settings backup
Google Cloud messaging for notifications

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 - getting the data

Connect to the reddit api and grab data from the dailyprogrammer subreddit

- Clean and filter the data
- Store the data

Task 2: Implement UI for Each Activity and Fragment

Create a theme for the app and implement the following views:

- Build UI for MainActivity
- Build UI for each fragment in MainActivity (All, Easy, Intermediate, Hard)
- Build UI for Navigation drawer
- Build UI for sorting menu
- Build UI for challenge details
- Build UI for challenge solutions
- Build UI for for saved challenges

- Build UI for completed challenges
- Build UI for About view
- Build UI for master > detail flow

Task 3: Implement content provider for persisting data

Build a content provider to persist the data

- Have a way to update the when new challenges are released
- Update the when something is marked as favorite or completed

Task 4: Add tests

Create JUnit tests along the way to verify data

Task 5: Add sorting functionality

Build functionality to update views based on newest, oldest, by difficulty, and by saved and completed.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"