Description
Intended User
Features
User Interface Mocks
    Screen 1
    Screen 2
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.
Next Steps: Required Tasks
    Task 1: Project Setup
    Task 2: Implement UI for Each Activity and Fragment
    Task 3: Implement Backend
    Task 4: Add Google Services
    Task 5: Widget

GitHub Username: s-udacity

Weedroid

Description

Weedroid is a client for the Weechat IRC client. While IRC bouncers allow multiple clients to connect under a single username, they do not provide persistence or work well with spotty mobile connections, and most setups provide only ~100 lines of scrollback. Reformatting a persistent Weechat client for mobile brings a contemporary mobile messaging workflow to the venerated IRC protocol.

Intended User

IRC users who could benefit from staying in touch via IRC on their phones

Features

- Syncs with remote Weechat instance
- Notifies user of highlights and PMs

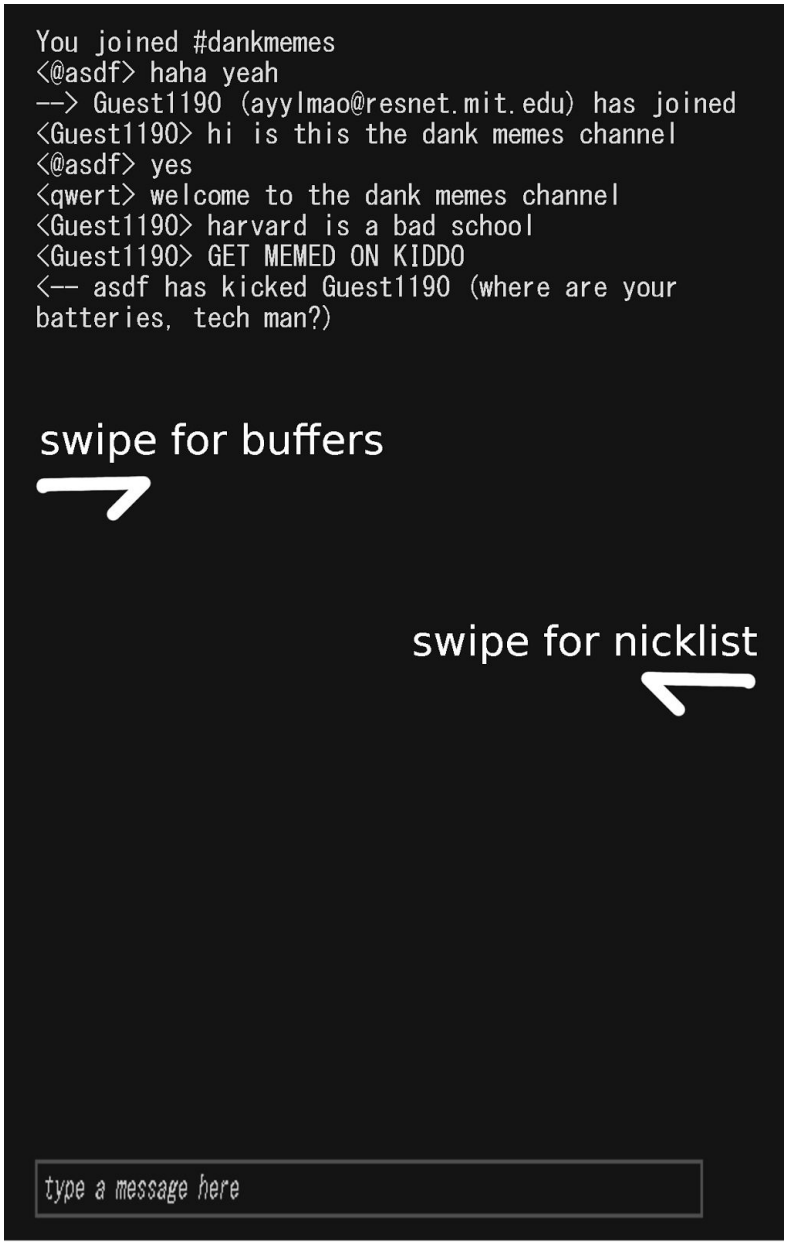
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1

Tablet layout

Screen 2



Phone layout

Key Considerations

How will your app handle data persistence?

The app will use a simple ContentProvider to manage the user's preferences (such as the servers they prefer to connect to).

Describe any corner cases in the UX.

The user may not be able to tell the difference between when the app is running in the background or when it is not running at all. An optional persistent notification icon can notify the user when there is a connection prese

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

The weechat-relay library provides an abstraction layer on top of Weechat's relay (remote control) protocol. Using a well-maintained third party library ensures my app doesn't break if Weechat updates their API.

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

Add the weechat-relay dependency and ensure it compiles properly and can by imported.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for the buffer (chat) activity
- Add buffer-switch sidebar
- Build settings UI

Task 3: Implement Backend

- Populate buffer with data from weechat-relay with a SyncAdapter
- Create ContentProvider for settings

Task 4: Add Google Services

- Admob ads (which can be disabled voluntarily in settings)
- Analytics (opt-in in settings)

Task 5: Widget

- Implement "current activity" widget

- Shows which buffers have new activity (as Weechat does)