How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

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**"

GitHub Username: Sirchip

Gitdo

Description

Github is a powerful place for collaboration for both professionals and hobbyists. Users can manage code, issues, contribution from other developers and track assigned work.

Problem:
When a user is assigned to a pull request or issue they are only notified when the issue is created, commented, or updated.There is no great way to keep up or track pull requests or issues on the go.It is easy to forget an issue or for that issue to become stale because the user has no easy way to keep up with iit.

Proposed Solution:
Design and build an app that allows a user to add repositories, stay up to date on issues and pull requests. View and track issues and pull requests per repository, and see the comment history for an issue or pull request.

Intended User

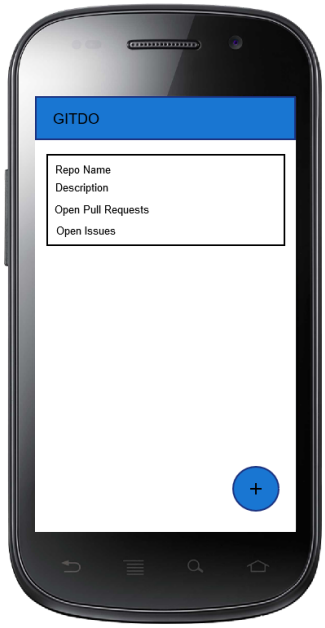
The intended audience are developers that use github for tracking pull requests and issues on a daily basis.

Features

- Add repositories for tracking
    - Search for repositories
- Show a count of pull requests and issues
- Allow user to read comments
- Notification when updated
- Connect and use github api

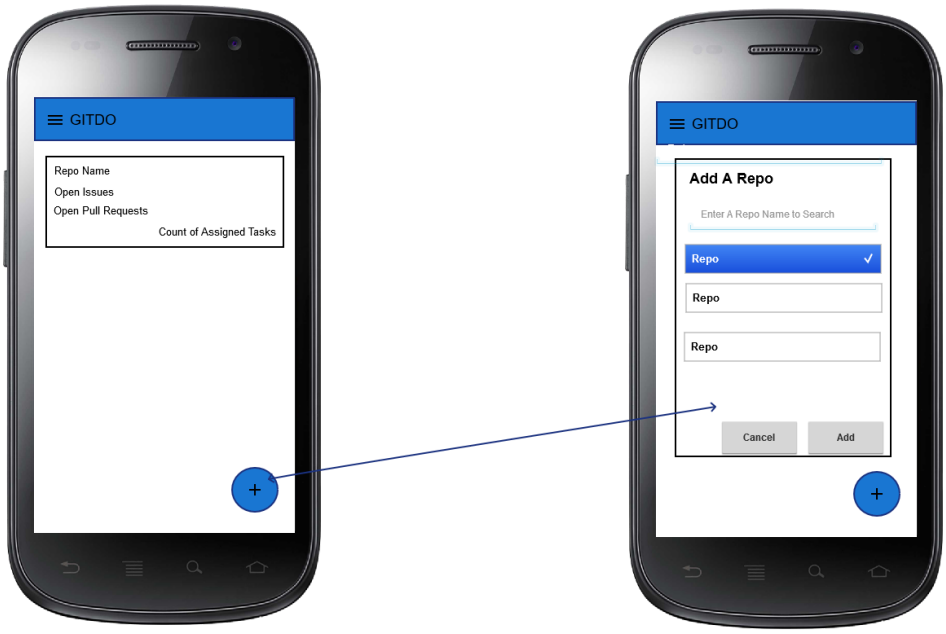
User Interface Mocks

Screen 1 - Landing Screen



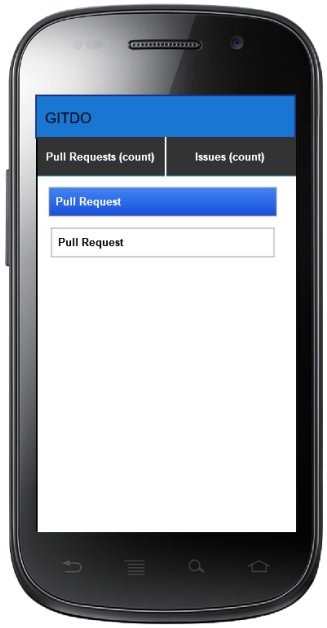
Landing page for the app, shows the list of tracked repos including count of open issues and pull requests.

Screen 2 - Find and Add Repository



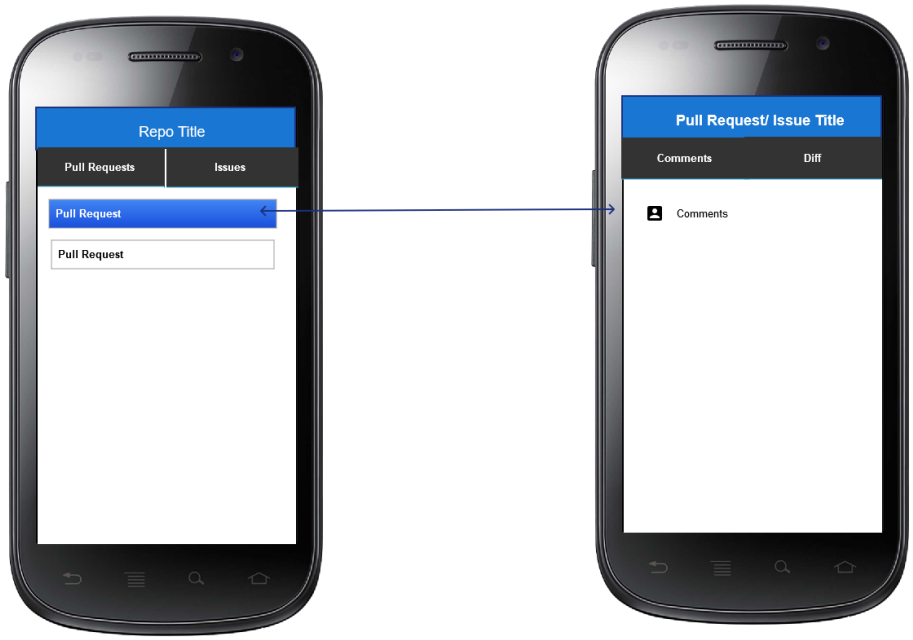
Clicking the FAB will open to an edit text to execute a search of repositories from github. Will be supported by lazy loading and pagination. Can also support advanced search by date, user, size, and language.

Screen 3 - Selected Repo Page - Pull Requests



Selecting a repository will open a tab view with options to select pull requests and issues by default Pull requests are open. This can be expanded to show other information like branches or an about tab.

Screen 4 - Selected PR/ Issue View



Selecting a pull request or an issue from the list will open a view with tabs. One tab will show the comment history for the selected item and it will include the picture of the user that left it. For a pull request the other tab is a webpage with the diff of the pull request.

Key Considerations

How will your app handle data persistence?

Local sqllite database updated with a sync adapter

Describe any corner cases in the UX.

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

- Retrofit to make network calls and parsing data simple and manageable.
- Butterknife to reduce the boilerplate view logic setup
- Glide for loading user avatars
- An ORM library to make queries simpler and easier to read.
- Espresso for view testing
- Junit/Mockito for plain java testing

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

- Get api key for github

Task 2: Build github API wrapper library

- Build a Library around the github api that includes the models and rest endpoints.
    - Make open source

    - Start with just the primary apis needed for the app
- Tests for library
- This can be a separate project from the app code

Task 3: Implement UI for Each Activity and Fragment

- Implement Landing page
    - Recyclerview
    - FAB to open “Add Repo” dialog
- Add Repo Dialog
    - Searchable edit text
    - Recyclerview of results
    - Add Button
        - Adds to list
    - Cancel Button
        - Close dialog without adding
- Repo Detail View
    - Tab View
        - RecyclerView for PRs
        - RecyclerView for Issues
    - Click on PR Tab
        - Show list of PRs
    - Click on Isssue Tab
        - Show list of isses
- PR/ Issue View
    - Tab View
        - RecyclerView comment history
            - Load comment
            - Load image
- Testing
    - Espresso with mock data

Task 4: Local Database and Update

- Create support for a local database
- Build Sync Adapter for uploading
- Notify user when a repo has been updated.

Task 5: Make screens functionable

- Landing Screen
    - Load lists from database
- Add Repo Dialog

        - Connect repo search to api wrapper
        - Add Button
            - Add repo to database and update DB
            - Notify activity of update
- Clicking Repo
    - GoTo: Repository Activity
- Repository Activity
    - Load repository PRs and Issues from database
- PR Fragment
    - Load comment history
- Issue Fragment
    - Load comment history
- Testing
    - Junit tests for utils and presenters

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**"