GithubMetrics : Udacity Android Nanodegree Capstone Project

GitHub Username: geniushkg

GithubMetrics

Description

Evaluating developers github profile manually by surfing each repository is tiresome task. GithubMetrics comes to rescue , you just have to enter username and general profile over will be generated.

Intended User

Recruiters working in technical field , more specific , those who recruit software engineers.

Features

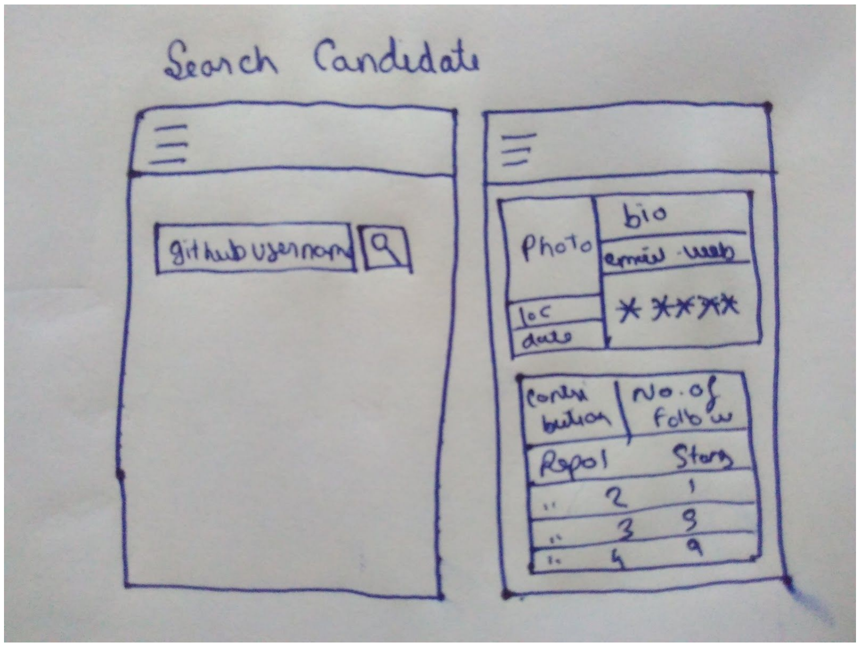
- Displays users streak graph

- Number of followers
- Contribution in year
- Saves history in database
- Most Important , rates developer hirablity on scale of 1 to 5.

User Interface Mocks

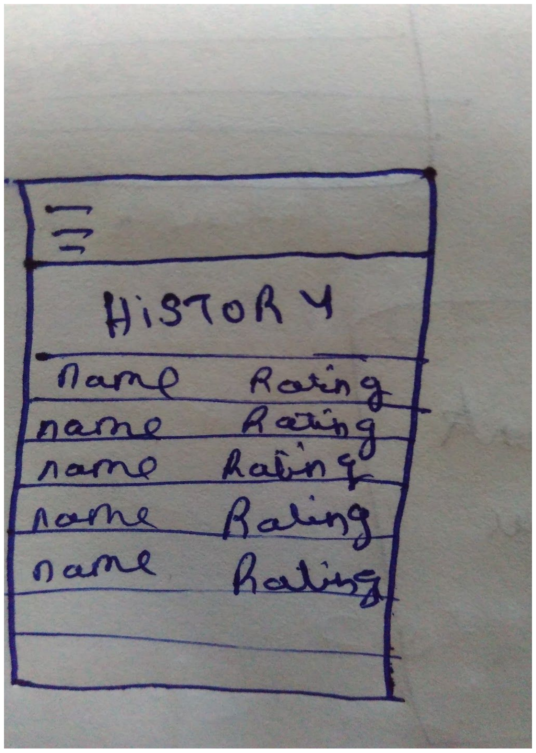
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1 : search empty and searched



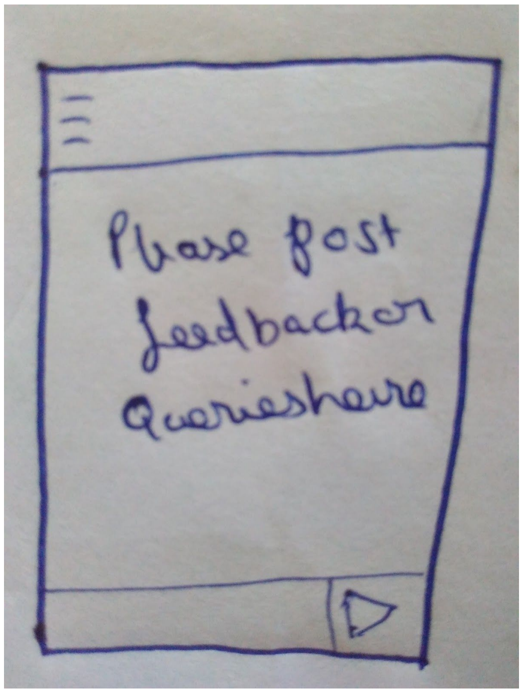
This is image of search fragment , that will be shown by default
Another image is of same search fragment will data is loaded

Screen 2 : History



History will provide username of candidates searched also , it will provide ratings of that candidate

Screen 3 : In-App Feedback



This will allow user to share feedback with developer of app.

Key Considerations

How will your app handle data persistence?

Use of sharedpreference to store access token and miscelleneous informations
Built content provider to store searched candidate history

Describe any corner cases in the UX.

Tapping of rating will display user metrics in details , and how the score is calculated.

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

Github-oauth - For github Oauth Authentication
Picasso - Image loading caching
Retrofit - Github Api consumption over network

Describe how you will implement Google Play Services.

Firebase authentication - via github-oauth library , to login user and get access token.
Firebase Analytics to see which feature is used maximum , and if possible get some ui feedback

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 google-serivce json file from firebase to app
- Update gradle dependencies for libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity
- Build UI for HomeActivity
- Build Fragment for candidate search
- Build UI for displaying searched candidate history
- Build UI for in-app feedback

Task 3 : get design assets

- Get icons
- Scale or crop images to be used,etc

Task 4: Integrate Github Login and store token to Firebase auth

- Integrate oauthgitlib to fetch user access token and store in shared preference
- Exchange Oauth token with firebase auth and get User data
- Implement Flags and make sure user login is never asked again.

Task 5: HomeActivity

- Implement navigation drawer with menus
- Fragment manages and handle all cases

Task 6: Search Fragment

- Implement cardviews and other ui element
- Implement AsyncTask to fetch data from network

Task 7: History & Feedback

- Implement recyclerview and display history
- Implement in-app messaging for getting feedback

Task 8: Loaders

- Implement POJO classes , Loadermanager and callbacks
- Overirde Acvitivy lifecycle to use loaders to update ui

Task 9 : Widget

- Implement widget to show last queried candidates.