TheMuseAndMe

- 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.
  - Describe how you will implement Google Play Services.
- Next Steps: Required Tasks
  - Task 1: Project Setup
  - Task 2: Implement UI for Each Activity and Fragment
  - Task 3: Your Next Task
  - Task 4: Your Next Task
  - Task 5: Your Next Task

GitHub Username: xiaobo1992

TheMuseAndMe

Description

TheMuseAndMe is a client Android app for theMuse(www.themuse.com). Finding the interesting information about jobs, companies, career coaches and advice on the app.

Intended User

The intended user will be people who are doing career search. This app will provide information about jobs, company and career advice to help the user find interesting jobs easily.

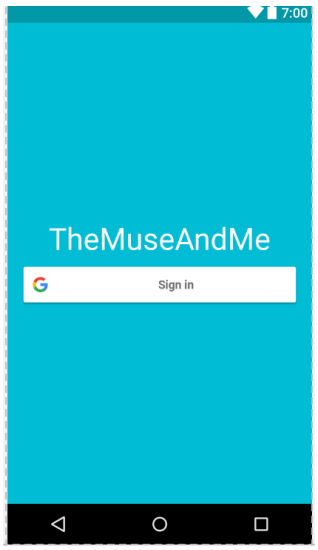
Features

- Viewing listed jobs, companies, coaches and advice

- Viewing details of the jobs, companies, coaches, and advice
- Storing interested jobs, companies, coaches, and advice
- Recording company application history
- Login the main pages of the app

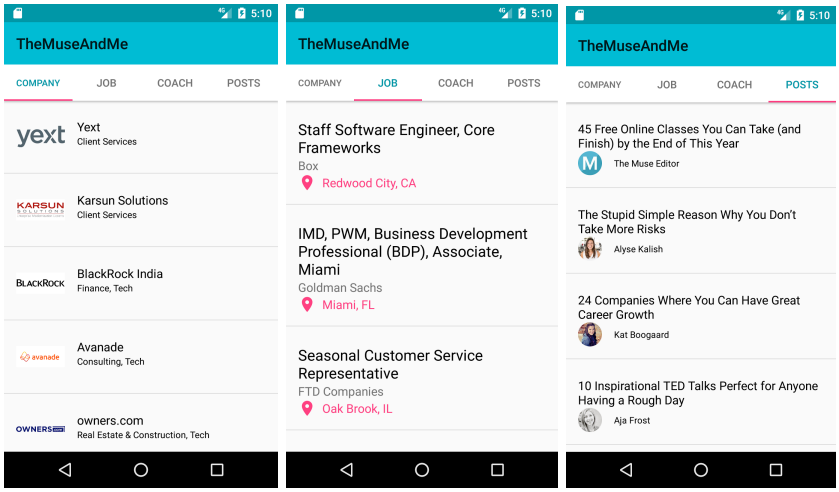
User Interface Mocks

Screen 1



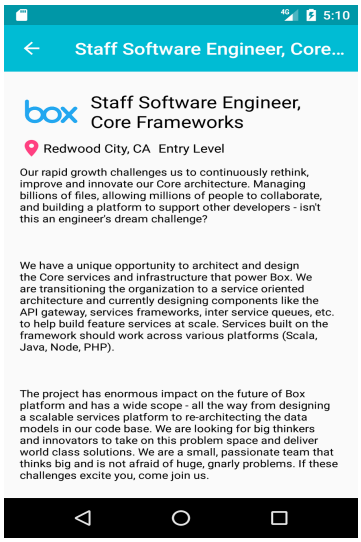
Login page for beginning

Screen 2



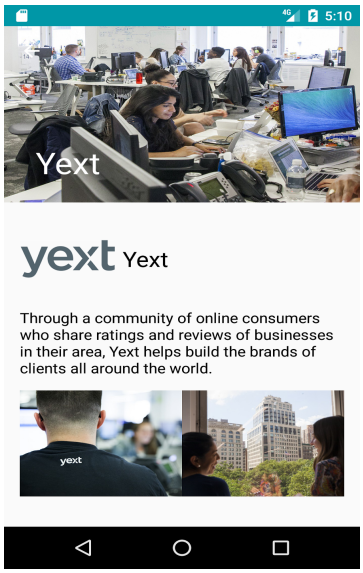
After login, the main page for list company, job, coach, and posts, this page will be similar for saved page

Screen 3



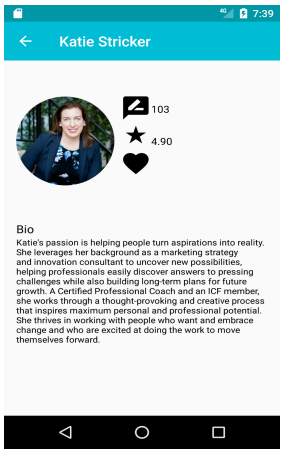
Job detail after click item in job pager

Screen 4



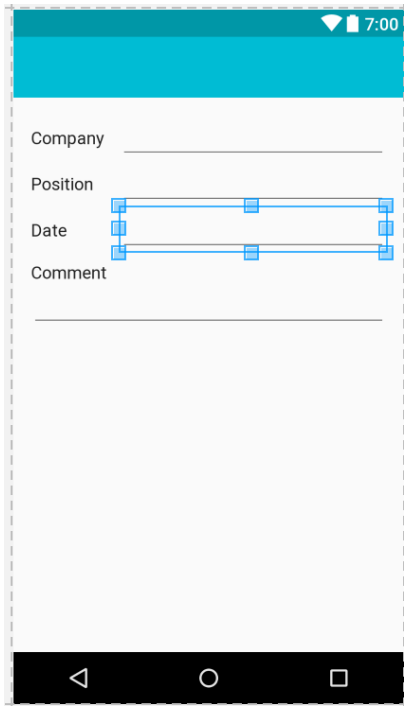
Company detail after click company event

Screen 5



coach profile after click item in coach pager

Screen 6



Save job application page

Screen 7

Widget for present company

Key Considerations

How will your app handle data persistence?

The app will use ContentProvider and MySQLite(or Firebase) for data persistence.

Describe any edge or corner cases in the UX.

- Saving the app state when people doing the rotation on the phone
- Stopping the infinite loading when the all the data received from the request

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

- okHttp: make HTTP rest request
- ButterKnife: binding the layout component
- fresco: loading imaging at the background thread
- Gson: data transmit between activities

Describe how you will implement Google Play Services or other external services.

- Google sign in: use for third-party login
- Google map: use for locating company and job location

Next Steps: Required Tasks

Task 1: Project Setup

- Setup dependency
- Setup theme of the app

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity - Activity
- Build ViewPager for each list (Job, Company, Coach, Post)
    - Job list - Fragment
    - Company list - Fragment
    - Coach list - Fragment
    - Post list - Fragment
- Build ViewPager for each favourite job, company, coach, post
    - favourite job list - Fragment
    - favourite company list - Fragment
    - favourite coach list - Fragment
    - favourite post list - Fragment
- Build UI for application history - Activity
    - job application history list
- Build UI for widget to present company information

Task 3: Build TheMuse library

- Create connection between app and TheMuse API
- Create relevant request method for get jobs, companies, coach, and advice according to parameter

Task 4: Build the database

- create contract, databasehelper, content provider for saved job, coach, company and advice table
- create contract, databasehelper, content provider for job application history

Task 5: Build relevant function

- login function
- list job, company, coach, post function
- save favorite jobs, company, coach, post function
- store job relevant application function
- create AsyncTask to load data from theMuse API
- create loader to load favourite from the contentProvider

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"