

SERACHJOB

AMAN SAHAI

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 you're 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

Task 3: Your Next Task

Task 4: Your Next Task

Task 5: Your Next Task

GitHub Username: aman-sahai

LinkedIn Profile: <https://www.linkedin.com/in/aman-sahai-05b160149/>

Jobs Info:

Description:

Information related to the job posts for the fresher's or the one who are presently looking for a job. User can know the details of the company, location, How to apply for a job, description of a job. For more details he can view the company profile from the application.

Intended Users:

Anyone who is in search of a job can use the app. the user can search the jobs available and know the details of a job like desired skills, procedure for how to apply, location, and details of a company.

Features:

- Fetches the data of the jobs from (jobs.github.com website) and display the details of the jobs.
- Widget for revisiting the app and read the last viewed job details.
- Authenticate the User using Firebase authentication.
- Main Activity will display the Ads using Ad mobs.
- the data of the Home Screen Widget will be updated via intent-service.
- User can add the job details to the the saved jobs using room database.

App Design Specification:

- App is written in JAVA Programming language for android
- App's Strings are stored in strings.xml
- the dimensions will be stored at dimens.xml
- Images are not stored in the app, but I use Picasso (an image loading and caching Library) which will take care of the images in the project.

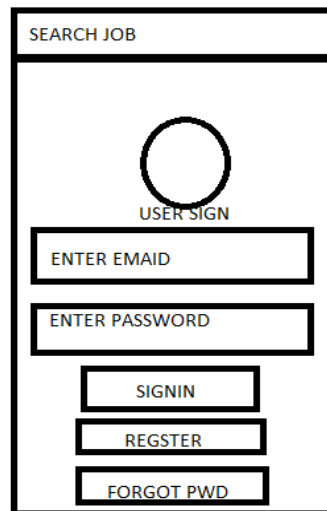
1/17/2019 CapStone_1 - Google Docs

https://docs.google.com/document/d/1YdHKucOYisGoKvaAu-LHXEnjMx2OM8ZyZHi_D66K8Lg/edit 3/9

- Colors used in the app can be seen at colors.xml
- Themes can be seen in styles.xml - Standard theme is used for the Application.

User Interface Mocks:

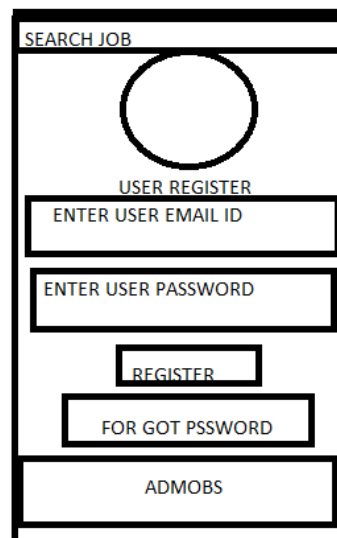
Screen1:-



A vertical rectangular mockup for a user login screen. At the top is a header bar labeled "SEARCH JOB". Below it is a large circle representing a profile picture, with the text "USER SIGN" centered underneath. This is followed by two input fields: "ENTER EMAID" and "ENTER PASSWORD". At the bottom are three buttons stacked vertically: "SIGNIN", "REGISTER", and "FORGOT PWD".

This is the main activity of the app where you have to login.

Screen 2 :-



A vertical rectangular mockup for a user registration screen. At the top is a header bar labeled "SEARCH JOB". Below it is a large circle representing a profile picture, with the text "USER REGISTER" centered underneath. This is followed by two input fields: "ENTER USER EMAIL ID" and "ENTER USER PASSWORD". Below these are three buttons stacked vertically: "REGISTER", "FOR GOT PSSWORD", and "ADMOBS".

This Screen Appears to user at the time to click the Registration.

Screen3:-

SEARCH JOB

↓

FOR GOT PASSWORD

ENTER REGISTER EMAIL ID

FORGOT PWD

This screen is a mobile app interface for the 'Forgot Password' feature. It has a header bar with the text 'SEARCH JOB'. Below the header is a large downward-pointing arrow. Under the arrow is the text 'FOR GOT PASSWORD'. Below that is a rectangular input field with the placeholder text 'ENTER REGISTER EMAIL ID'. At the bottom of the screen is a button labeled 'FORGOT PWD'.

This Screen appears when you click on Forgot password

Screen4:

SEARCH JOB FAV,SIGNIN

LOGO COMPANY LOGO

○

○

○

This screen is a mobile app interface for user authentication. It has a header bar with the text 'SEARCH JOB' and 'FAV,SIGNIN' on the right, followed by a hamburger menu icon. Below the header is a container with two items: a circular 'LOGO' and a rectangular 'COMPANY LOGO'. Below this container are three identical rows, each consisting of a circular input field and a rectangular input field.

This activity provides the list of movies in Grid View which are selected from settings men

Screen 5:

SEARCH JOB DETAILS	
COMPANY IMAGE	
TITLE	
COMPANY NAME	
DESIGNATION	
HOW TO APPLY	
LOCATION AT	
CREATED AT	
DESCRIPTION	
COMPANY URL	

This activity provides the details of the selected movie and displays a icon to add in Favourites.

Screen 6:



The user can view the Last viewed Job details in the Home screen widget. The home screen widget displays the Job title and name of the company.

Key Considerations:

How will your app handle data persistence?

This app will use Volley for fetching the data from the server. And uses Ad mobs for displaying ads.

Describe any edge or corner cases in the UX.

App will have a parental navigation button on the top left corner of the screen. From that the user can come back to parent activity. Display alert dialogs to the user if there is no internet connectivity.

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

- Picasso: to handle the loading and caching of images – Version 2.3.2,
- Volley for networking: Version 1.0.1.
- GitHub jobs Api : displays the jobs posted in that website.
- Firebase-Auth: To authenticate the user -Version 16.0.1.
- Android Studio Version - 3.1
- Gradle Version - 3.2.1
- Alexander Like button for favourites- 0.2.3

- Room Version: “1.1.1”

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

Firebase Authentication to authenticate the user.

Ad Mobs for the free and paid users.

Next Steps:

Task 1 : Project Setup

- creating a project and naming it as a FindMyJob.
- Configure the libraries, dependency for the app functionality.

Task 2 : Implementing UI for Each Activity.

- Design a layout for User Authentication.
- Design a layout which shows the details of the jobs posted in the website.
- Design a layout showing the details of a Selected Job details. And the layout contains favourite button. Which we can add the job to saved jobs.
- Design a layout showing the Saved jobs of the user and details of a jobs posted.

Task 3: Authentication

- Implement authentication in order to use the application.

Task 4: Data Fetching

- fetching the data from network.

Task 5: Room database

- storing the job details into saved jobs using Room database using View model live data.

Task 6: write the UI Espresso tests to the application.

Task 7: Sign the app in release mode and submit the application.