Table of Contents

GitHub Username: ciscorucinski

Resume App

Description

Resume App is a supplement to the traditional paper resume when you are searching for jobs. Typically, a person will have different resumes for different types of jobs and companies as a person wants to maximize the chances of getting hired when a company could be looking over hundreds of applications. This app allows a user to see a predefined resume about a potential candidate employee, and has methods to get in contact with said candidate. It is a great way to display a person's Android programming skills during an interview so they can explain to employers all the hidden parts they are not seeing.

Intended User

For the first release, the main users of the Resume App are myself and individuals or companies that will be hiring me. There will be specialized apps for each individual / company for a more personal touch. Also, that will allow me to display specific information to specific individuals and companies. The app for myself will be a Hub that will allow me to view what each individual or company can see in their specialized apps.

Future releases of the app, could all others to somehow upload their data to the app to allow companies they are applying for to view their resume in the app. This is uncertain, as this app is being built to show my programming skills with Android and knowledge of Android Studio and the Gradle build system. Therefore it is a good complement to a paper resume; however, other people cannot say the same thing.

Features

Individual / Company Apps

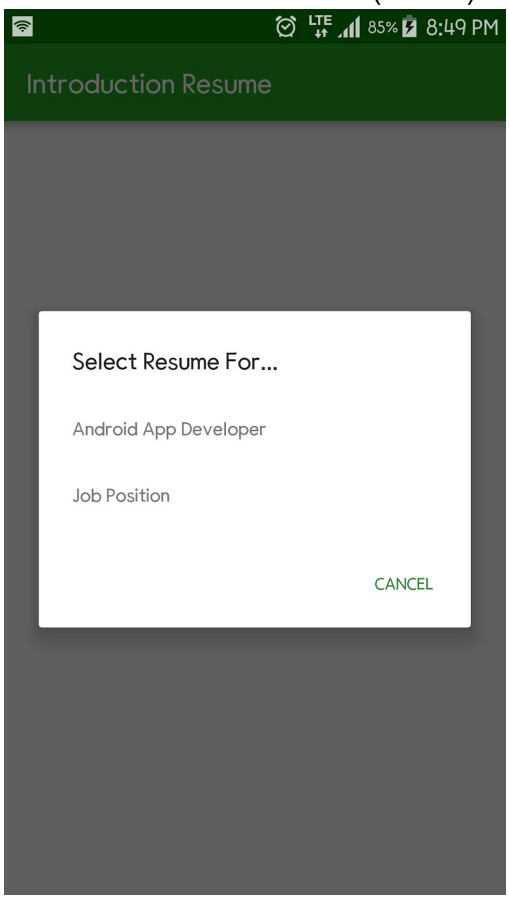
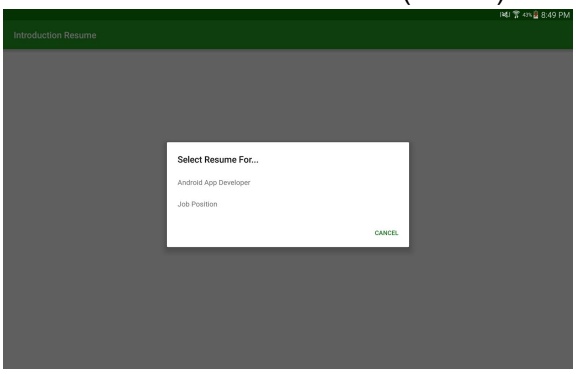
- Viewing a typical resume (specialized resume for the company) via sections such as Education, Work Experience, Projects, Contributions, and other personal information such as contact information and objective.
- Ability to view LinkedIn Profile
- Ability to view GitHub Profile
- Ability to get in contact with candidate employee through email and phone

Hub App

- All the features of the Individual / Company Apps for purposes of knowing what each app will look like and for testing / verifying components will work for the individual / company.

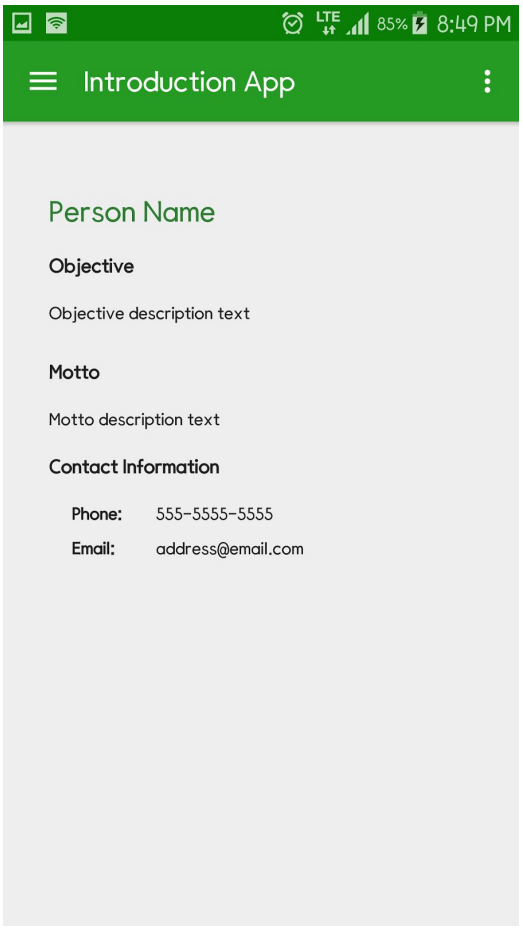
- Ability to select all available resumes that I have (Software Engineer, English Teacher, etc)

User Interface Mocks

Hub Companion Activity - Phone	Hub Companion Activity - Tablet
Screen 1 Phone - both orientations (similar)  Resume Selection Activity This is the first screen that is displayed to Hub App users. It allows people select a specific resume to view. That is its only purpose.	Screen 1 Tablet - both orientations (similar)  Resume Selection Activity Same as the phone description

Resume Activity - Phone - Basic

Screen 2
Phone - Portrait



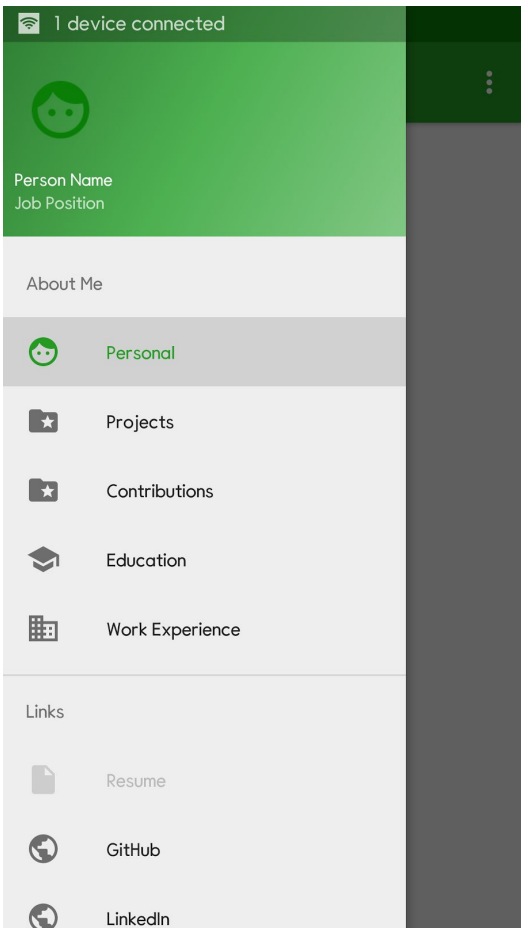
Resume Viewing Activity

This is the first screen that is displayed to users of the Individual / Company App. It is the second screen that is displayed to users of the Hub App.

It will display basic information about the person this resume is for such as the Objective, their motto, and basic contact information. This is considered the Personal information (see next picture)

Resume Activity - Phone - Navigation

Screen 2
Phone - Portrait

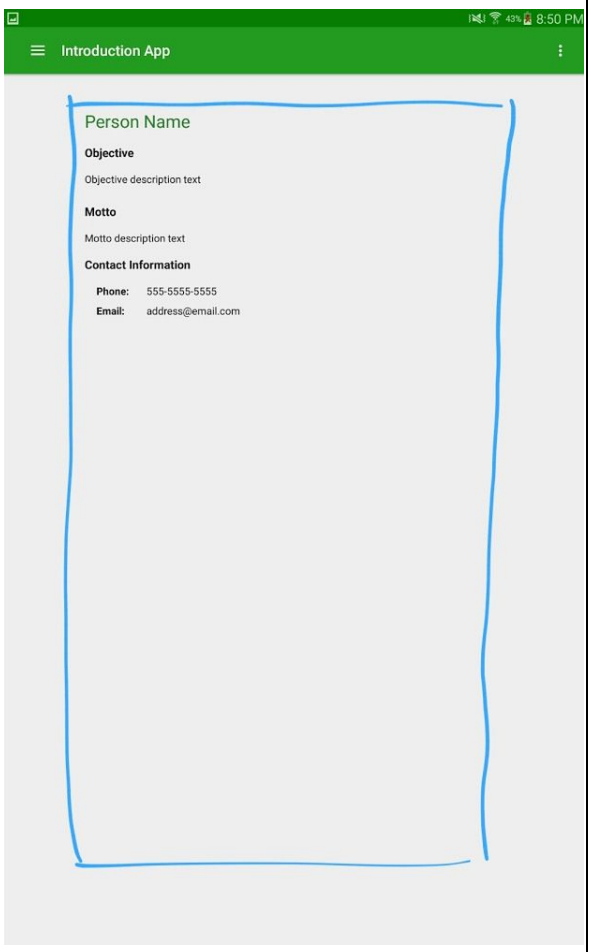


Resume Viewing Activity

This is what is displayed when the user opens the Navigation Drawer. The Navigation Header will display the Name of the person this resume belongs to, and their position they are seeking.The current displayed information is highlighted (Personal); however, they can choose from Projects, Contributions, Education, and Work Experience. Farther down the drawer, a user can see the person's GitHub and LinkedIn profiles. While it is hidden, there will be direct links to email and open up the dialer to call the person.

Resume Activity - Tablet Portrait - Basic

Screen 2
Tablet - Portrait

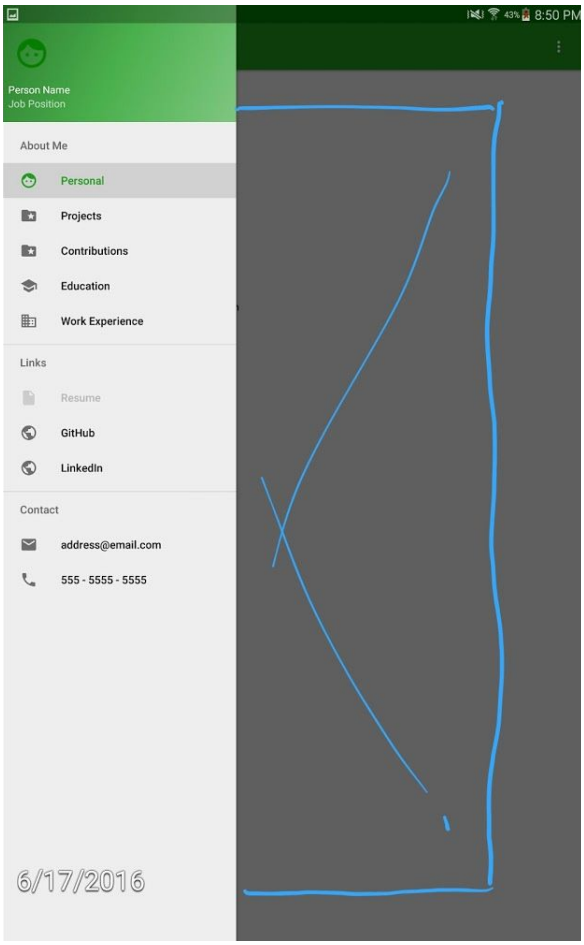


Resume Viewing Activity

Same as the phone description; however, extra padding is used.

Resume Activity - Tablet Portrait - Navigation

Screen 2
Tablet - Portrait

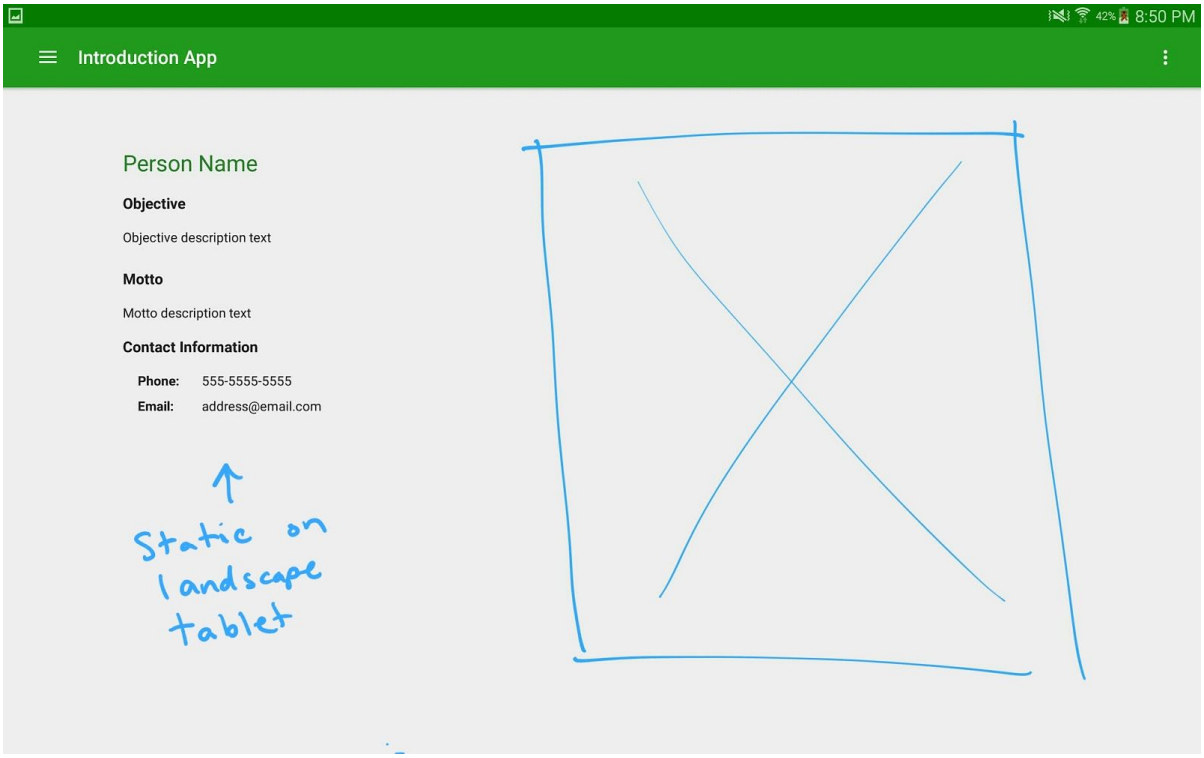


Resume Viewing Activity

Same as the phone description; however, the full Navigation Drawer is visible.

Resume Activity - Tablet Landscape - Basic

Screen 2
Tablet - Landscape

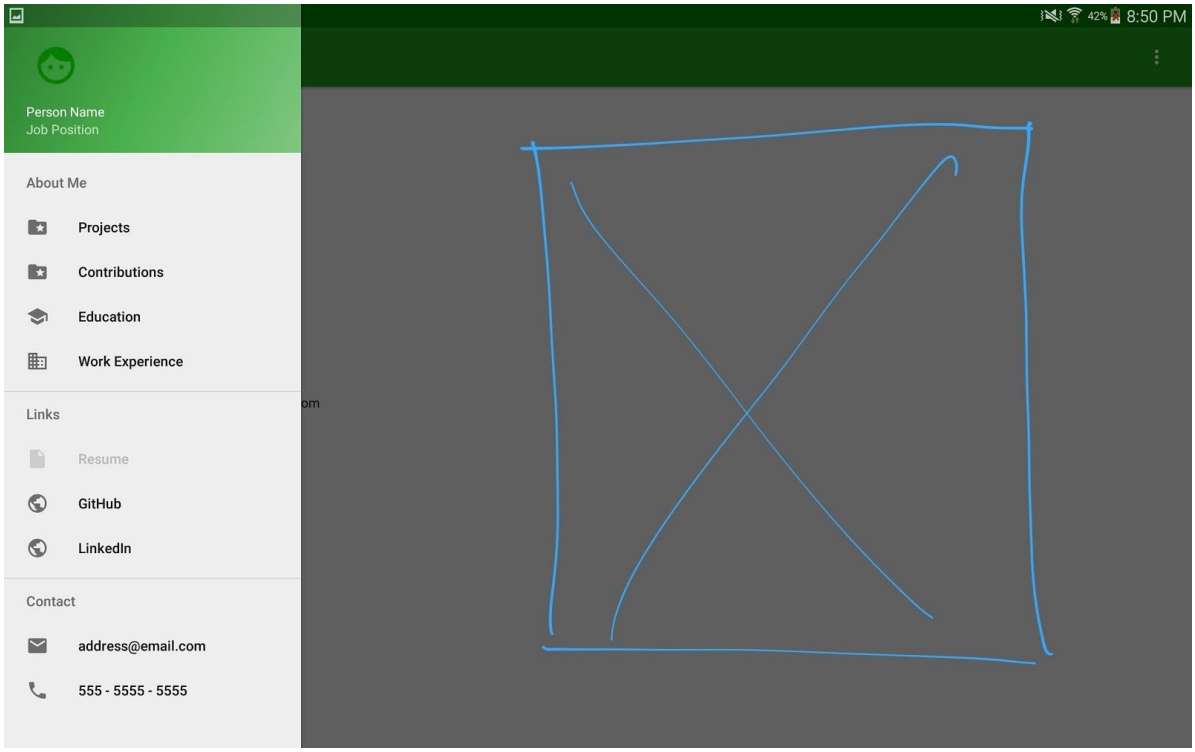


Resume Viewing Activity

Same as the phone description; however, the larger display area means that the Personal information will always remain static on the screen. When the user selects a new resume section to view (i.e. Projects, Contributions, Education, or Work Experience) then that information will be displayed where the BLUE box is. The BLUE box information will be scrollable if needed; however, the Personal information will remain locked in place.

Resume Activity - Tablet Landscape - Navigation

Screen 2
Tablet - Landscape



Resume Viewing Activity

Same as the phone description. The BLUE box behavior is the same as described by the "**Resume Activity - Tablet Landscape - Basic**". As you can also see. The Personal Navigation menu item will not be available to users. That is because that information is statically displayed on the screen as seen and described in "**Resume Activity - Tablet Landscape - Basic**"

Key Considerations

How will your app handle data persistence?

Data persistence will be handled with an SQLite Database and backed with a Content Provider to access the data. SQL will be generated via SQL Delight that is being developed by Square Inc. Initial data will be added to the database on first app startup. The first version of the app will not have an ability to insert new data into the database.

Future versions of the app could allow for the candidate employee to upload new resumes, update current resumes, and allow uploading a PDF version of their resume to the app for viewing.

UX Corner cases.

In the first iteration of the app, there won't be too many corner cases as there will be a very specialized interaction with little form of modifying things.

Future versions will add abilities for the candidate employee to add and modify the underlying data after an employer has installed the app. This will be handled via the new Firebase platform. This will add a whole series of corner cases; however, Firebase should be able to handle them with grace. Again, this will be in a future update.

1. **On tablets when an employer rotates from portrait to landscape when they have selected the Personal Information Section**

    a. Since the landscape tablet will allow a larger amount of information to be displayed, tablets in landscape will always display the Personal Information Section to users. The tablet landscape layout will not allow the selection of the Personal Information Section; it will just be displayed automatically in the UI.

    b. Therefore, a new default section will not be displayed; instead, the part of the UI that updates the new sections will remain blank.

    c. If the user rotates back to Portrait then the expected behavior of displaying the Personal Information Section will occur.

    d. If a user decided to select a new section to display in the landscape view and then rotates to portrait, then the new section would be displayed as expected.

2. **The user does not have a device capable of making calls via a SIM card (i.e. tablets)**

    a. The app will open up a dialer from other apps through VoIP such as Skype

Libraries to be used.

- Support (Design) Libraries : Simplify the coding and Materialization of my app.
- Support Custom Tabs: Allow viewing of webpages while staying in the app.
- SQL Delight: Creates Java model classes from real SQL statements. Also creates other raw SQL statements that can be used via a Content Provider.
- Auto-Value: Needed for SQL Delight
- Schematic: To create a content provider; however, the generated Content Provider will be copied and modified to allow the user of the raw SQL statements from SQL Delight
- Stetho: Viewing persistent storage on a phone via Google Chrome.
- Timber : Advanced Logging capabilities to make debugging easier.

Next Steps: Required Tasks

Task 1: Project Setup

- Create project
- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI's for…
    - Main Activity
    - Resume Selection Activity
- Create the XML resources for all the UI's

Task 3: Implement Custom UI

- Build Custom Views and layouts for...
    - Personal Information
    - Education
    - Work Experience
    - Projects
    - Contributions
- Hook up the custom UI's with the current fragments and activites

Task 4: Develop Build Variants

- Create Build Variants for Debug and Release.
- Create Flavor Variants for each specific company
- Implement special Logging capabilities in the debug build only
- Add Flavor specific resources

Task 5: Create Data Access Layer

- Create real SQL statements for all the different tables. Also create SQL to insert and select data based on parameters, and to insert predefined resume data for 1st app startup.
- Have SQL Delight generate my Model classes that as the raw SQL statements.
- Use Schematic to generate Database and Content Provider java classes.
- Copy-paste the generated Content Provider and modify it to use the Model classes raw SQL statements.
- Modify the Schematic Database class to use the Model's SQL statements to insert predefined data on apps 1st startup.
- Use Stetho to analyze the inserted data in the database.

Task 7: Implement Loaders

- Develop one Loader that does all performs all the needed functionality.
- Abstract out the parts that need changing to an abstract class that will handle all the common functionality.
- Develop the remaining Loaders using the Abstract class.
- Develop interface that links loaded data from the Loaders to Fragments

Task 7: Implement Data Gathering Ability for all the Fragments

- Use the Loaders and the interface to pass data between each other.
- Finish creating the data-dependent UI aspects. Double check they work with real data.