Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: **Nicholas McCormack**
Student ID: **X15007588**
Supervisor: **Padraig DeBurca**

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: Nicholas McCormack ______________________ Date 13/05.2018

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks

and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced

- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

National College of Ireland

BSc in Computing

2017/2018

Nicholas McCormack

X15007588

x15007588@student.ncirl.ie

Rolette Mobile Application

Technical Report

National College of Ireland

Table of Contents

Executive Summary

The main objective of this project is to create an effortless android application that provides a simple environment for job recruitment or job searching. Simplification is the key to the application, as the target customer base will be any person within the current workforce. The concept is quite straightforward, why search for jobs when you can have an app that can do it for you? There will be two groups of users that will benefit from this simple system, Employees and Employers. Employees will use the app to find for relevant roles that match their criteria. Employers will use the system to find applicants, to fill the posted roles within their organization.

The main purpose of the app is to create a user-friendly environment for job hunting where even the most technical novice will be able to navigate and operate the application. I want to remove the mundaneness of the typical job search, rather than you searching for the job let the app search for you. The job postings will be pushed to employee users in same format as the tinder dating app pushes matches to their users. With a simple swipe left or swipe right action to reject or register interest.

The App will be developed within the Android studio IDE using its XML files to create the UI layout. Google's Firebase will be used to offer a real time database to retire store and process data within the application.

1 Introduction

From the beginning of the Internet one of the main aspects of the championed aspects was the system to share information. Sharing and retrieving information are two different things. How many hours have you as a person sat down and looked through a job recruitment site such as Linkedin, Monster or Jobs.ie? How many job recruitment apps have you installed on your phone at this moment in time. In my opinion they all offer the same mundane service as each other, they all offer the same recruitment process that has been around since the days of the industrial age. We now live in a fast paced connected world, a world of fast food, speed dating , why should we not increase the speed at which the recruitment process works? On average a person who is waiting on a job application result to return could be waiting up to weeks for a response and that's just if you have been successful, the time for an unsuccessful applicant may be weeks to months later. My application Rolette is here to offer the alternative to the fast moving times we live in.

1.1 Background

I decided on the creation of this application from my own personal experiences with using job recruitment apps and websites. I felt the typical process of what is available is too time consuming, with little return on the time invested in job applications. I felt some jobs that are advertised on sites and apps over inflate requirements around experience or qualifications required to apply. The ability to apply is still available but it can be very discouraging in further searching for jobs. The main reason I had decided to return to college was to reskill myself and increase my stature in the realms of the work force. It is my hope this application will offer a platform to users a new way of applying for jobs, in a way where the invest less time and emotion. This in turn I hope will remove the dis heartening effect the current system gives users.

1.2 Aims

The aim of the android application project is to develop a extremely user friendly environment for job seekers and employers to come together and get results faster than the old process currently used within the work force. The application should offer to the employees a throwaway type job application system where they can apply for many jobs that suit their relevant expertise without having to invest huge amount of time searching for roles or applying for roles in which they may need to restructure application forms. I want to bring the jobs to them. With regards employers I want to offer them a database of potential employees where they can fill roles in a much faster period than what is currently on offer at the moment. Most big companies use recruitment agencies that may not necessary have the complete interests of the company as their main priority. The reason that companies use recruitment agencies is purely down the company simply does not have the time or resources to review every applicant for every role that applies for a position within their company. Rolette will remove majority of this time consuming process by moving through recruitment stages in a much faster process.

2 Technologies

In order to complete the android application project from the beginning to the final stage the development will done using the Android Studio IDE. The Android Studio IDE will house the majority of the tools needed to complete the project. Additional tools needed to complete my project I will need to use Google’s Firebase (Non SQL database), Android device sensors for location services, and API and libraries to pull job postings from other job recruitment sites.

2.1 Structure

The System, I will look at the requirements within the system, functional and non-functional. The process’s that are involved in each requirement will be documented within this chapter. We will also look at the user requirements, environmental requirements and data requirements.

Within this chapter we will also discuss some of the design and architecture within the system. These will be provided through diagrams I have created to help better understand what we are trying to accomplish. I will also show some of the GUI through mockup wireframes.

Testing, I will discuss the series of intended tests I would be planning to carry out and what questions they will hope to answer.

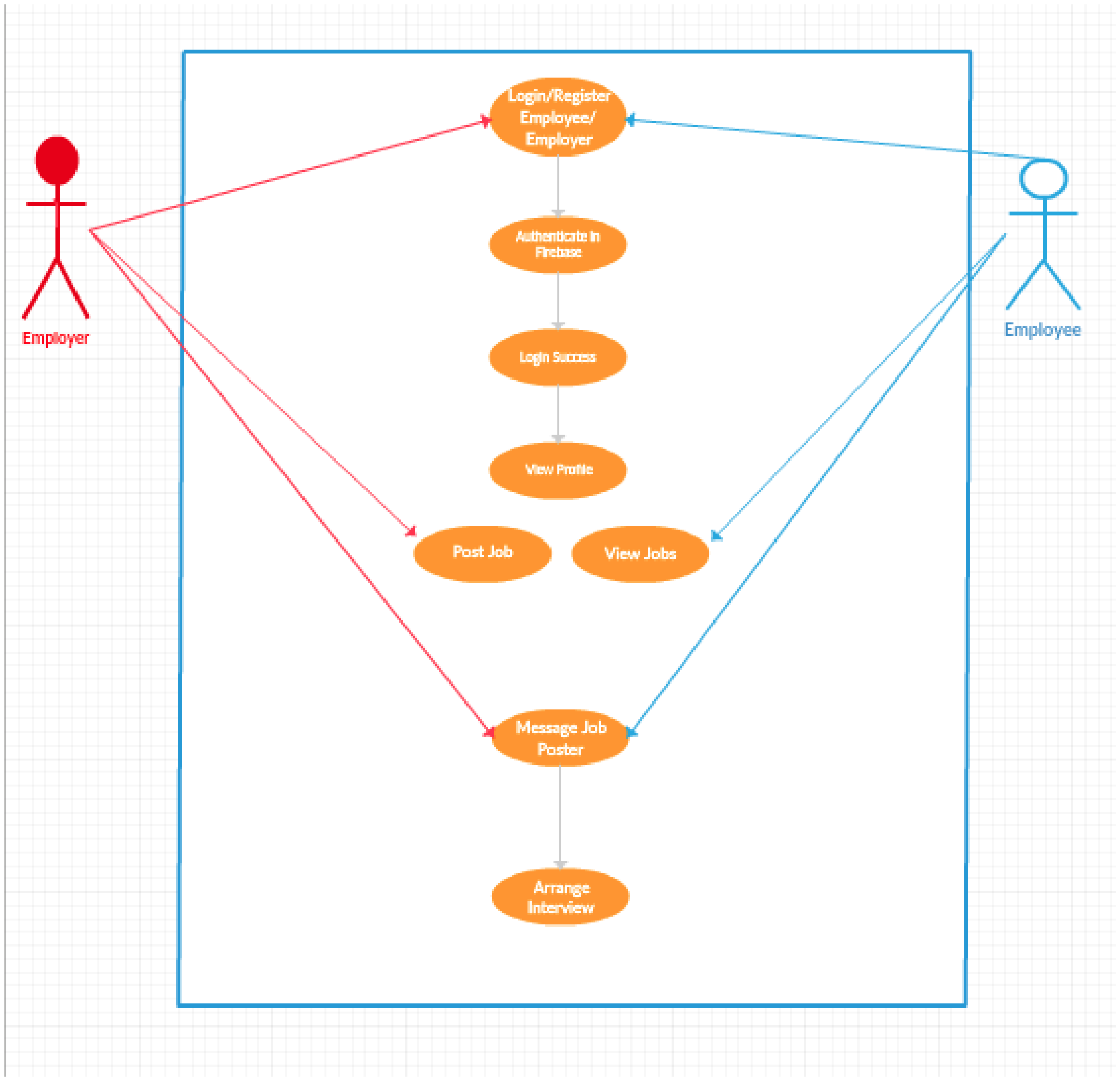
Conclusions, here I hope to help better understand what we have done right or wrong while creating this application. We will also discuss the opportunities and the constraints within the application.

Further Developments, I will discuss where we could go to further expand this application. Whether it is to different platforms of mobile technology or to increase the current functionality or add new functions to the existing product.

3 System

3.1 Requirements

3.1.1 Functional requirements



3.1.2 Requirement 1 <Employee Job Search>

3.1.2.1 Description and Priority

This feature allows the employee user to login and search the available jobs that are specific to his location and field of expertise. User can decide whether to reject or register their interest in these roles. Priority High

3.1.2.2 Use Case

1 Employee user will be able to login to their profile with the correct credentials.
2 Employee will be displayed the relevant roles available to them.
3 Roles will be specific to their field of expertise and location parameters.
4 User will be given the option to register interest or reject the roles on offer.

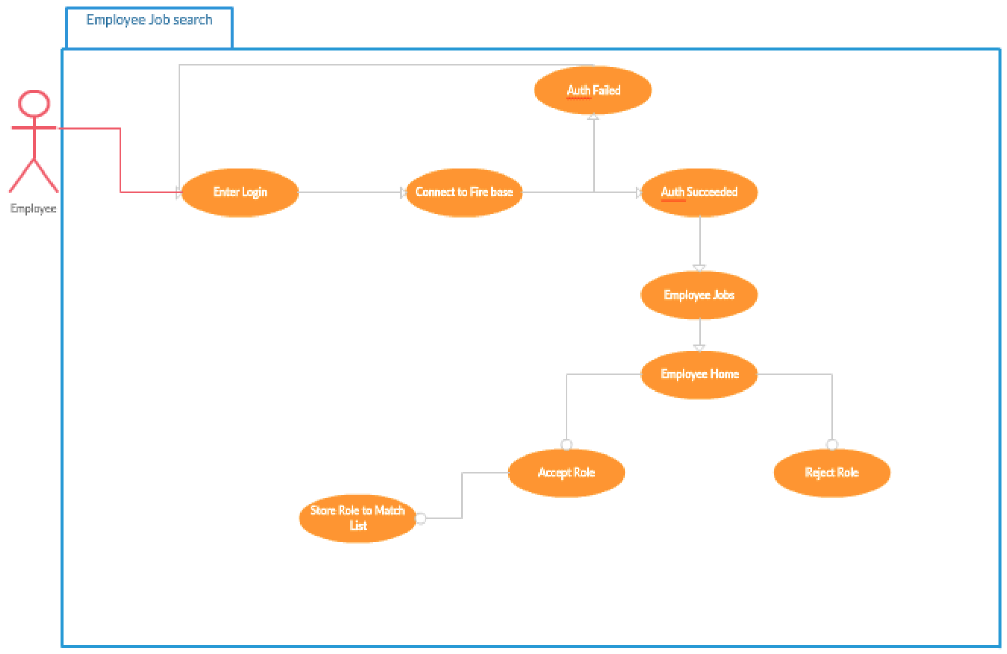
Scope

The Employee user will have the options to view the roles available to them and be given the ability to accept or reject the roles on view. Roles acceptance or rejection will be done by simple swiping feature.

Description

This use case describes the actions that take place when the Employee user rejects or accepts a role.

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

This use case starts when an Employer logs in to the application

Main flow

5 User launches the app on device app.
6 The user successfully logs in to the application.
7 The application retrieves all the profile information for user.
8 The information is rendered and displayed for the user to interact with.
9 User can reject or accept the role on display.
10 Accept role, User can view role in more detail and make direct contact with the prospective employer.

Alternate flow

A1 : <Failed log in>

1 User launches app with a stable Internet connection.
2 The User enters the credentials to access profile.
3 Credentials do not match those stored in Firebase.
4 User is returned to the login page to renter their credentials.

3.1.3 Requirement 2 <Employee Profile View>

3.1.3.1 Description and Priority

This feature will allow the Employer user to view the profiles they have received from employees that have registered an interest in the job they have posted. Action the Employer user can take is to either reject the employee, or accept the employee to then start a dialogue with them. Priority High

3.1.3.2 Use Case

1 The employer profile should have their own unique login to access the app.
2 Employers will be able to view the roles they have posted.
3 Employers will be able to view the profiles of employees interested in their roles.
4 Employers will be able to continue dialogue with users they have accepted to open communications with.

Scope

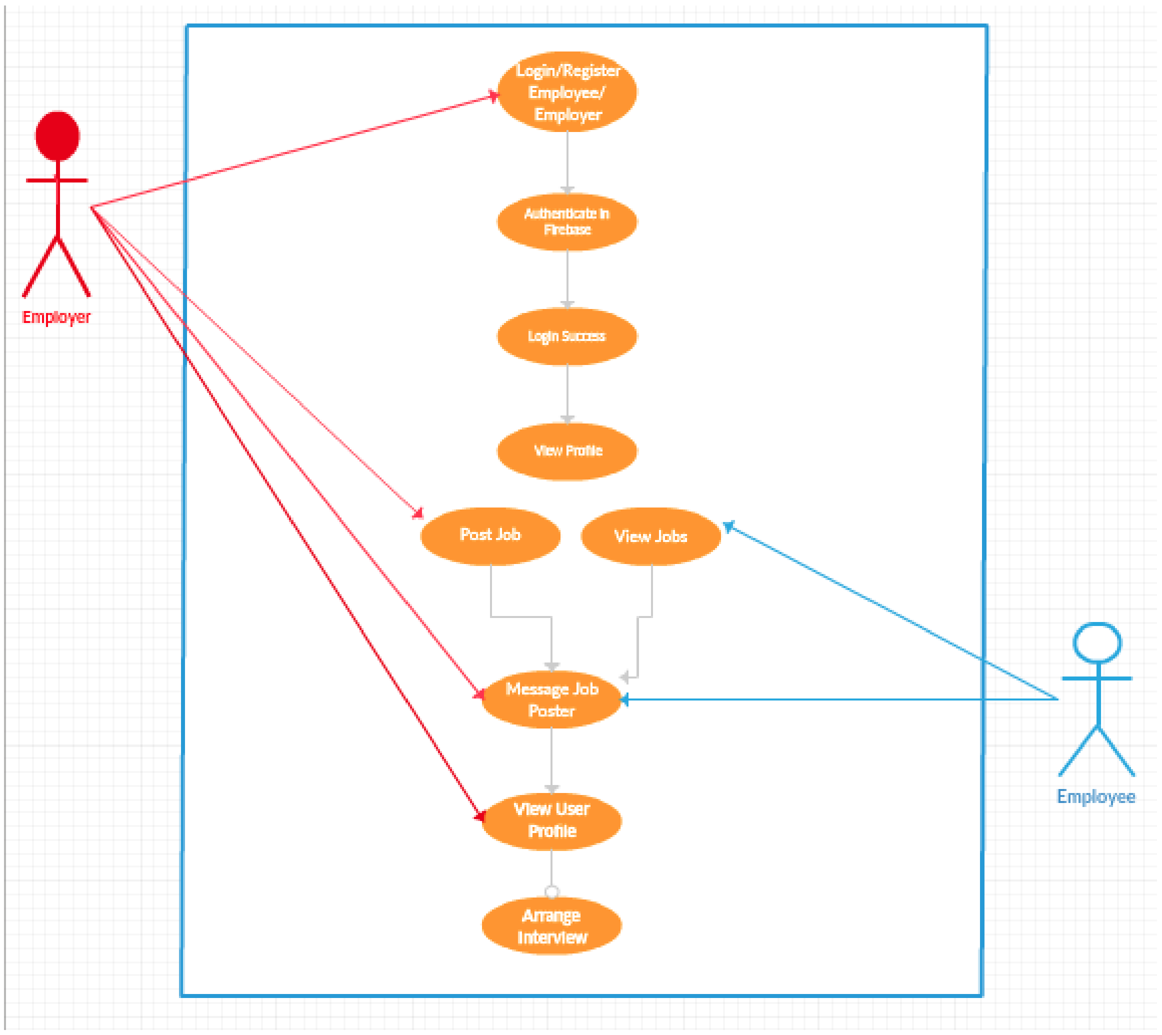
The scope of this feature is to give the employer user the ability to view their persons of interest profiles in regards to a role they have posted.

They can then view the employee profiles that have registered their interest and reject or accept the employee for further communications.

Description

This use case describe the actions that take place when an Employer user logs in to the application to view the employee profile with regards a role they have posted. Also the subsequent actions they can undertake once viewing the interested employee profile.

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

This use case starts when an Employer logs in to the application

Main flow

1 User launches the app on device app.
2 The user successfully logs in to the application.
3 The application retrieves all the profile information for user.
4 The information is rendered and displayed for the user to interact with.
5 Within the accepted list the employer can open dialogue.

Alternate flow

A1 : <Failed log in>

1 User launches app with a stable Internet connection.
2 The User enters the credentials to access profile.
3 Credentials do not match those stored in Firebase.
4 User is returned to the login page to renter their credentials.

3.1.4 Requirement 3 <Employer Job Post>

3.1.4.1 Description & Priority

This feature is one of the main components of the app. The job posting should consist of only relevant information regarding the position. No contact information should be included in the post. Job posts should be clear and concise and relevant. Priority : High

3.1.4.2 Use Case

1 Employers will have their own company login to access their account/profile to access features.
2 Employers can create new postings.
3 Employers can edit existing posts.
4 Employers can delete existing posts.
5 Employers will include relevant information about post.

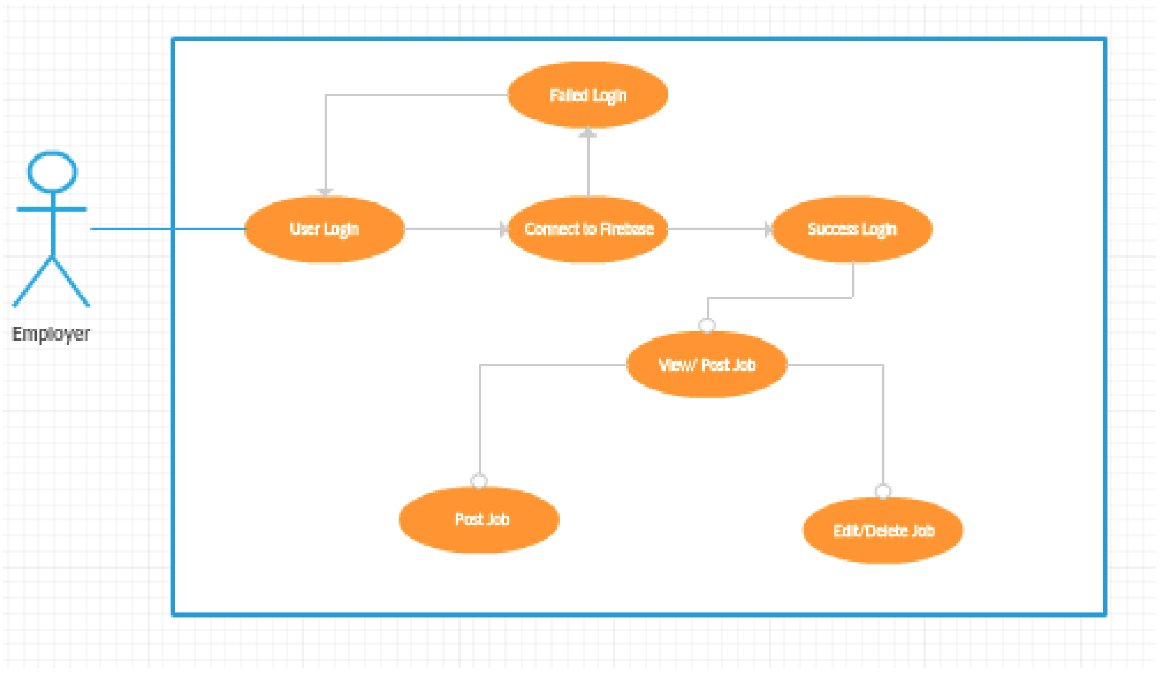
Scope

To give the employer profile the ability to post positions available within their business. The feature aims to making posts clear and concise, only relative info regarding the job should be included. This is what will be viewed by potential employers and must not consume too much of their time.

Description

This use case describes the path of the job post feature within the employer profile.

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device.

Activation

This use case starts when an Employer logs in successfully to the application.

Main flow

1. The system identifies the user credentials against those stored in the firebase database.
2. The <Employer> can view their main dashboard. Job posts function is available to them here.
3. The Employer can Create new post, Edit existing post, Delete existing post.

Alternate flow

A1 : <Failed Log In>

1. Employer credentials do not match those that are stored in the firebase database.
2. User is returned to the login form.
3. User cannot proceed without the correct login details.

Termination

Application closes when user exits the application with automatic logout.

Post condition

Application automatically logs out.

3.1.5 Requirement 4 <Message Employer>

3.1.5.1 Description & Priority

This feature will allow users to login and contact employers regarding jobs they have posted on the application. Messages are carried to and fort by using JSON. Users can interact wit a post and contact the employer directly.

Priority: High

3.1.5.2 Use Case

1. Users should be able to login.
2. Users should be able to view jobs posted by employers.
3. Users should have the ability to contact the employer who posted the job.
4. Users can then have a private conversation over the posted job.

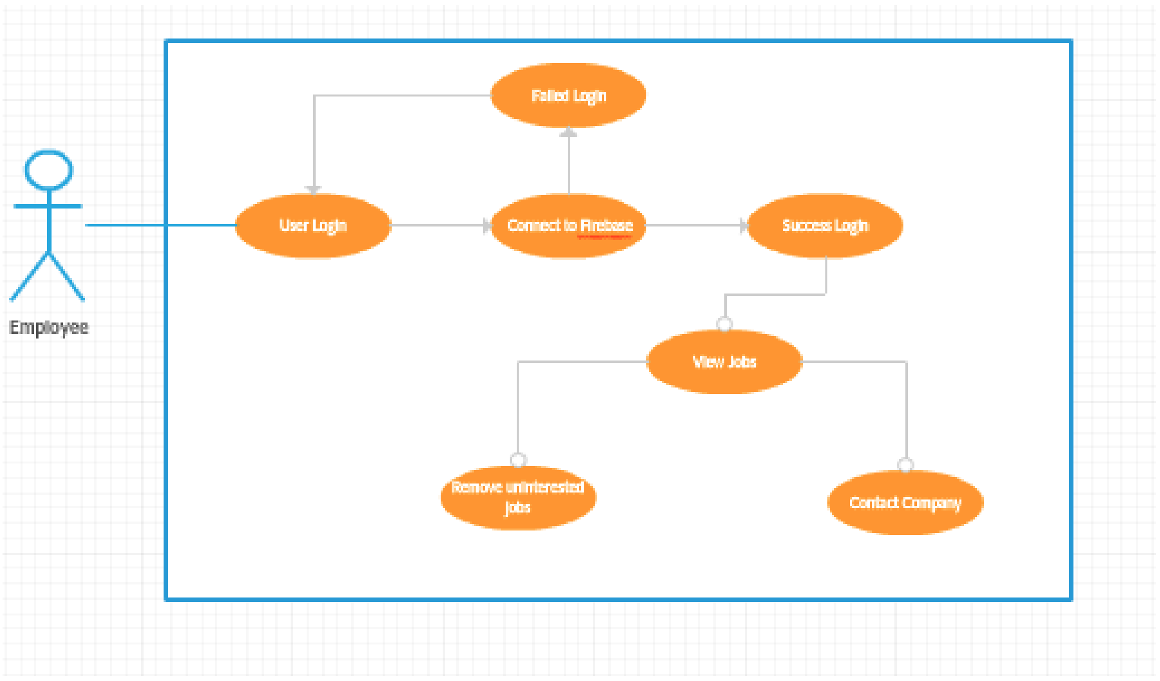
Scope

Login of an employee user and the contacting of an employer of a job they have posted in the application.

Description

This use case describes the path of the registered Employee user making contact with an Employer user about the posting of a job they have advertised on the application.

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

The use case starts when the Employee opens the application.

Main flow

1. The Employee logins are checked against those stores within the firebase database.
2. The Employee is displayed their own profile dashboard.

Alternate flow

A1 : <Failed Login>

1. Employee login credentials checked against the credentials saved against those stored in Firebase.
2. Authentication fails returned to the login page,
3. Employee has to enter correct details to proceed.

Termination

Application closes when user exits the application with automatic logout.

Post condition

Application automatically logs out.

3.1.6 Requirement 5 <Edit Profile>

3.1.6.1 Description & Priority

This feature will allow users to edit their own profiles with the option to upload a profile picture. The point of this feature is to make the user an attractive prospect to potential employers.

Priority : Medium

3.1.6.2 Use Case

1. Employees should have a dashboard to view their profile
2. Employees should have an ability view of all jobs within the application
3. Will have the option to contact employers of the jobs they are interested in.
4. Employees will have the option to edit their existing information.
5. Employees will have the option to upload a profile picture.
6. Employees will have the option to view their messages.

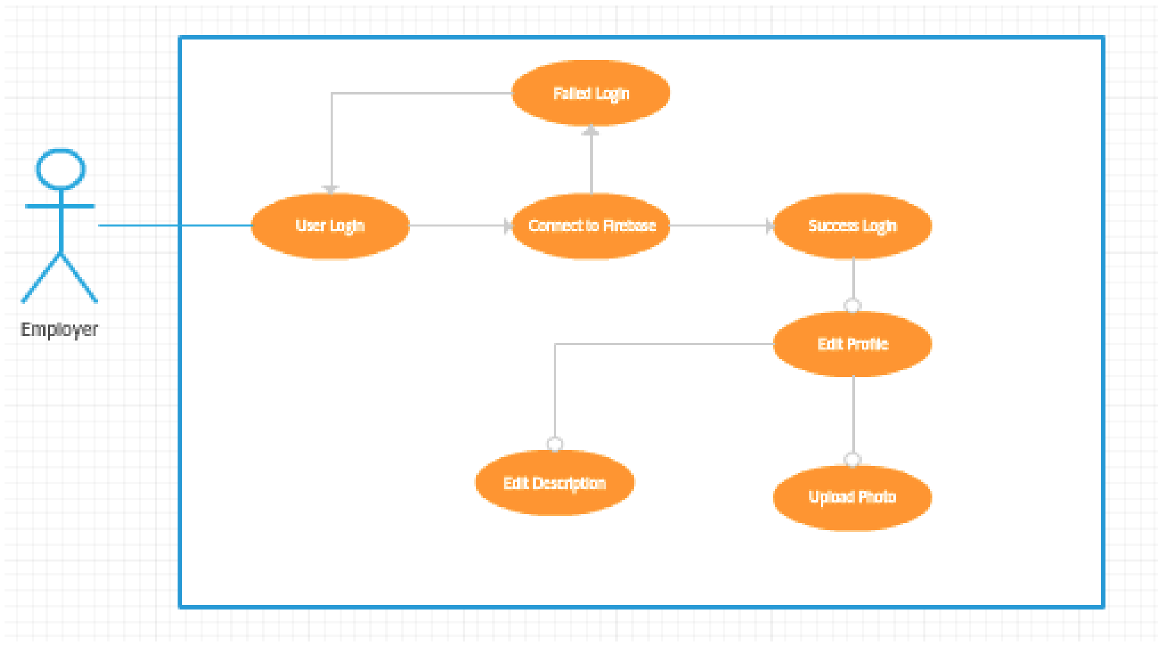
Scope

Edit or Enter Profile information for the employee or employer user. This is to help either the employee describe the relevant experience they possess as to help land them a job with a company. With regards the employer this is to help an Employer see the companies location, area of expertise, companies standards or expectation's,

Description

This use case describes the path of the creation of a new profile and editing or uploading new information to the profile.

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

The use case starts when the Employee opens the application on a compatible android device with a stable Internet connection.

Main flow

1. Employer opens application on a compatible device.
2. Employer enters credentials to access employer/employee profile dashboard.

3. Credentials are checked against those stored within Firebase.
4. Employer gains access to the Employer dashboard.
5. Option to Post/Edit Job or Edit Profile.
6. Employer edits profile details fed back to firebase to be stored.

Alternate flow

A1 : <Failed Employer/Employee Login>

1. Employer opens application on a compatible device.
2. Employer enters credentials to access employer profile dashboard.
3. Credentials are checked against those stored within Firebase.
4. Credentials don't match those stored within Firebase.
5. Employer is returned to the login screen to renter credentials.

Exceptional flow

E1 : <Upload Photo>

1. Employer opens application on a compatible device.
2. Employer enters credentials to access employer profile dashboard.
3. Credentials are checked against those stored within Firebase.
4. Employer gains access to the Employer dashboard.
5. Options to Post/Edit Job or Edit Profile.
6. Employer edits profile uploads photo.

Termination

Application closes when user exits the application with automatic logout.

Post condition

Application automatically logs out.

3.1.7 Requirement 6 <Log in Employer>

3.1.7.1 Description & Priority

This feature allows employers/recruiters to maintain their profiles for either themselves or on a company's behalf. Employer profiles will have information regarding the companies included in the description, including their name, location, number of employees, sector, a brief description of the company's culture and briefs. The login feature is the most important feature form a security standpoint. Users are required to provide a valid email address and a strong password consisting of one uppercase letter and one numerical value at least.

Priority : High

3.1.7.2 Use Case

1. Employers shall be able to log on to new profile.
2. Employers will be able to access their profiles and edit/update their information.
3. Employers will be able to choose their system preferences such as notifications.
4. Employers will be able to post new roles/positions to be made available for the viewing of employees.
5. Employers will be able to remove roles update.
6. Employers will be able to view potential matched candidates and make contact by opening dialogue.
7. Employers must have a company email address or Google account.

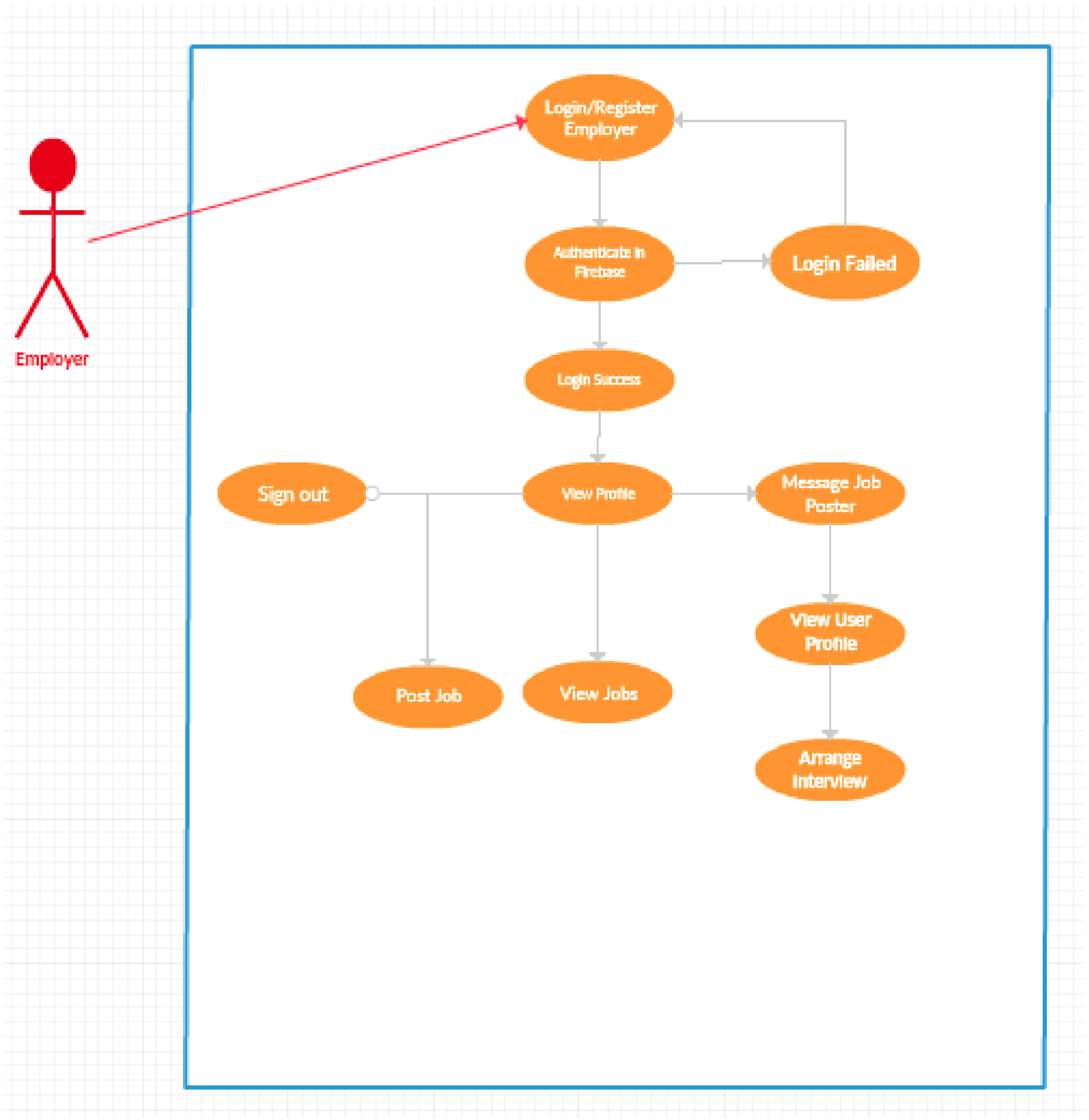
Scope

Create a profile for the employer side of the app with information about the company. This should also give employers extra features to their profile such as to add, remove job. Employers should be able to see the messages they have received from employees regarding jobs they have posted.

Description

This use case describes the login of a new employee profile

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

The application is open and running on a compatible android device. The user has a stable Internet connection.

Main flow

1. Employer opens application on a compatible device.
2. Employer enters credentials to access employer profile dashboard.
3. Credentials are checked against those stored within Firebase.
4. Employer gains access to the Employer dashboard.

Alternate flow

A1 : <Failed Log In>

1. Employer opens application on a compatible device.
2. Employer enters credentials to access employer profile dashboard.
3. Credentials are checked against those stored within Firebase.
4. Credentials don't match those stored within Firebase.
5. Employer is returned to the login screen to renter credentials.

Termination

Application closes when user exits the application with automatic logout.

Post condition

Application automatically logs out.

3.1.8 Requirement 7 <Log in Employee>

3.1.8.1 Description & Priority

This feature allows employees to maintain their profiles for to remain attractive to potential employers or to view the jobs posted by potential employers. Employee profiles will contain information regarding the current age their, field of expertise and should include a brief description about themselves.

Priority : High

3.1.8.2 Use Case

8. Employees shall be able to log on to new profile.
9. Employees will be able to access their profiles and edit/update their information.
10. Employees will be able to view new roles/positions to be made available for the viewing by employers.
11. Employees will be able to register their interest in the posted jobs.
12. Employers will be able to view the profiles of potentially interested candidates and make contact after dialogue has been opened.
13. Employees must have a valid email address or Google account.

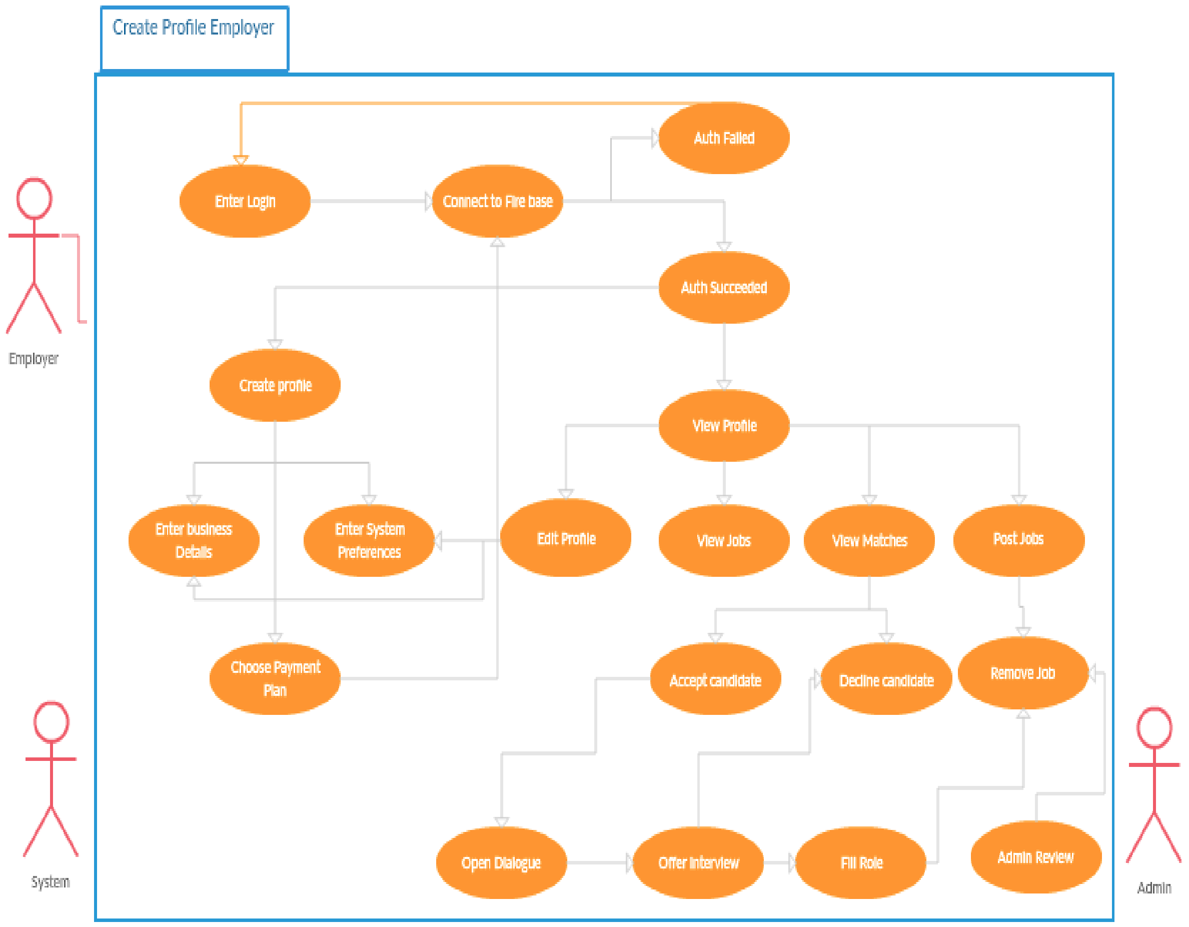
Scope

Create a profile for the employee side of the app with information about themselves and their experience and their field of expertise. This should also give employees the ability to view jobs posted on the application. Employees will also be able to make contact with employers regarding a job posted within the application.

Description

This use case describes the login of a new employee profile

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

The application is open and running on a compatible android device. The user has a stable Internet connection.

Main flow

5. Employee opens application on a compatible device.
6. Employee enters credentials to access employee profile dashboard.
7. Credentials are checked against those stored within Firebase.
8. Employee gains access to the Employee dashboard.

Alternate flow

A1 : <Failed Log In>

6. Employee opens application on a compatible device.
7. Employee enters credentials to access employer profile dashboard.
8. Credentials are checked against those stored within Firebase.
9. Credentials don't match those stored within Firebase.
10. Employee is returned to the login screen to renter credentials.

Termination

Application closes when user exits the application with automatic logout.

Post condition

Application automatically logs out.

3.2 Requirements Not Implemented

This is where I will include the requirements I did not implement from the mid point presentation and discuss why I felt they where not included in the final project.

3.2.1 Requirement 3 <Admin Login>

3.2.1.1 Description & Priority

This feature will allow master administrators to login and give the system administrator the full control over the application. The main admin will have the same abilities as the sub admin but will also have the power to create new admins and remove admins from the database.

Priority : High

3.2.1.2 Use Case

5. Admins should have a dashboard to view all requests such as pending requests to remove a post.
6. Admins Should have a view of all profiles within the app.

7. Will have a log report of all cases reported to the app about breaches of the terms of use,
8. Will be able to suspend and unsuspend accounts.
9. Edit payment information of an employer account.
10. All chat dialogue between employers and potential employees is available
11. Creation and deletion of sub admin accounts.

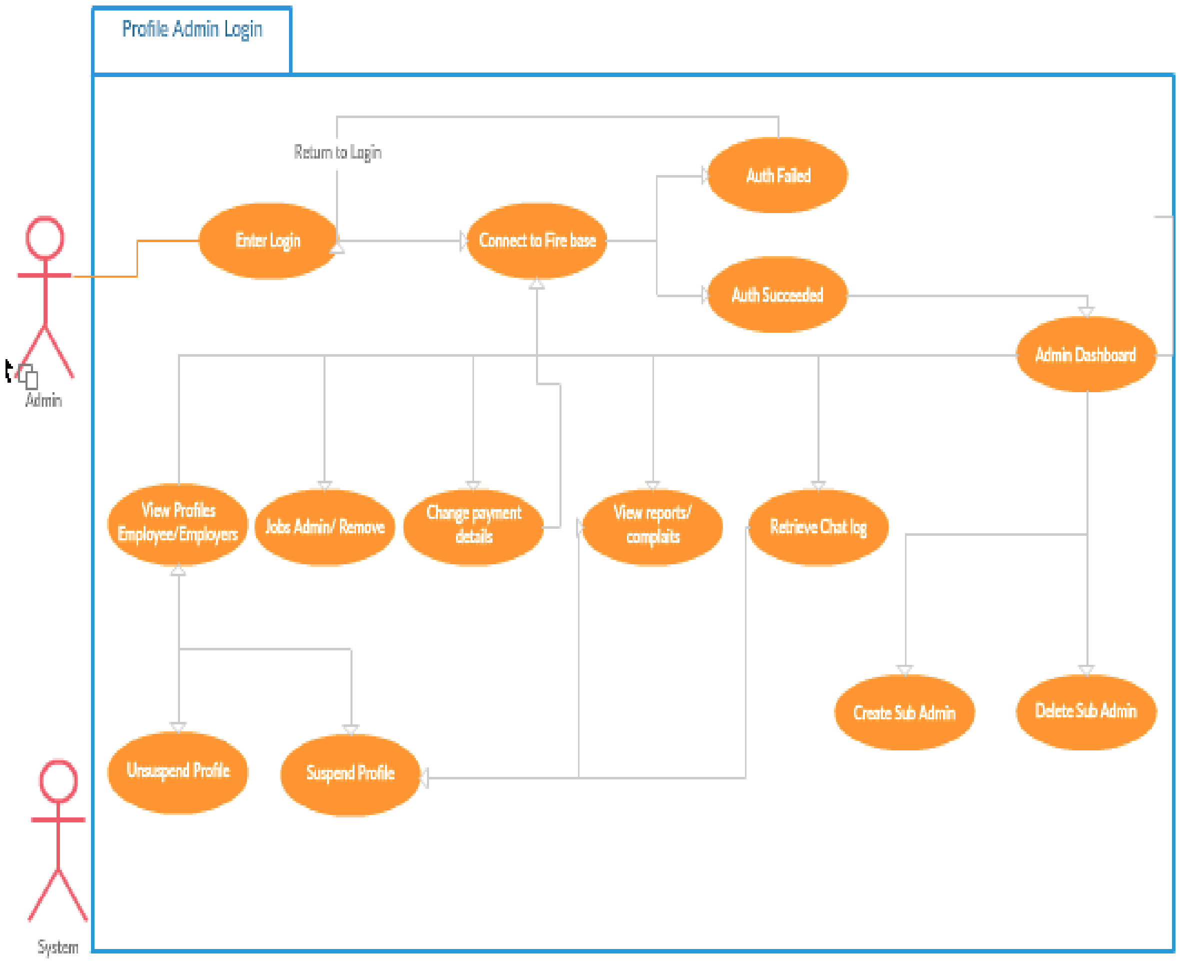
Scope

Login of Admin and rights of an admin on the application. To securely login and have major administrative rights over the app.

Description

This use case describes the path of the creation of an admin profile.

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

The use case starts when the administrator opens the application.

Main flow

3. The administrator logins are checked against those stores within the firebase database.
4. The administrator is displayed the administrator dashboard.

Alternate flow

A1 : <Failed Login>

4. Admin login credentials checked against the credentials saved against those stored in Firebase.
5. Authentication fails returned to the login page,
6. Admin has option to reset the password via reset link.

Termination

Application closes when user exits the application with automatic logout.

Post condition

Application automatically logs out.

3.2.2 Requirement 4 <Sub Admin Creation>

3.2.2.1 Description & Priority

This feature will allow master administrators to create new sub administrator profiles, which will give all most administrative rights to the new profile user. Actions the new admin will be able to control will be to review all profiles, approve or remove job posts, suspend and unsuspend accounts, view/delete/update payment details on accounts. Review the matches between an employer and employee of a job post. Request the dialogue between two parties for review of cases of reported behavior or suspicious job post removal by employers.

Priority : Medium

3.2.2.2 Use Case

7. Admins should have a dashboard to view all requests such as pending requests to remove a post.
8. Admins Should have an ability view of all profiles within
9. Will have a log report of all cases reported to the app about breaches of the terms of use,
10. Will be able to suspend and unsuspend accounts.
11. Edit payment information of an employer account.

12. All chat dialogue between employers and potential employees is available

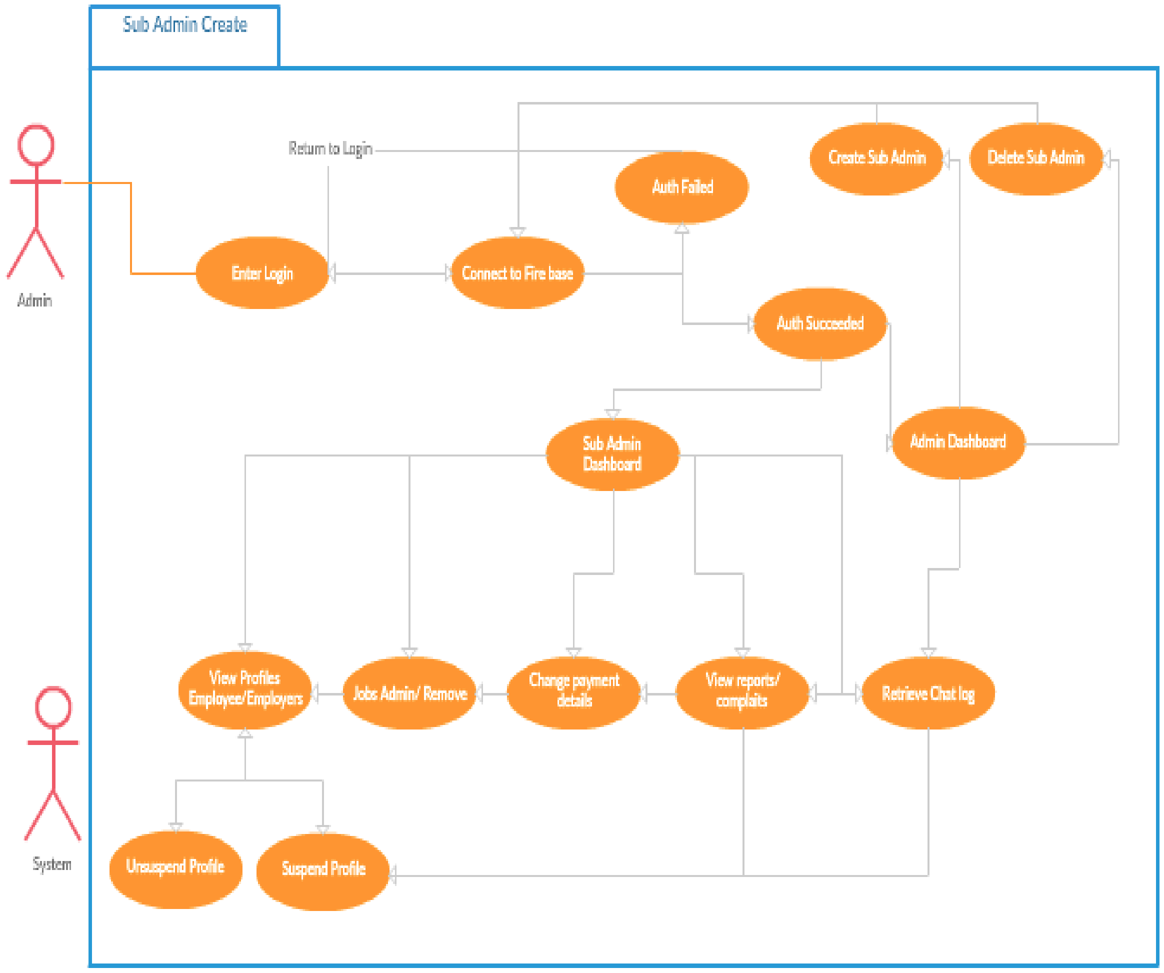
.*Scope*

Create a profile for the sub admin.

Description

This use case describes the path of the creation of a sub admin profile.

Use Case Diagram



Flow Description

Precondition

The application is open and running on a compatible android device. The user has a stable Internet connection.

Activation

The use case starts when the administrator opens the application on a compatible android device with a stable Internet connection.

Main flow

7. Main admin opens application on a compatible device.
8. Main admin enters credentials to access admin profile dashboard.
9. Credentials are checked against those stored within Firebase.
10. Admin gains access to the Admin dashboard.
11. Option to create new sub admin.
12. Creates new sub admin, details fed back to firebase to be stored.

Alternate flow

A1 : <Failed Admin Login>

6. Main admin opens application on a compatible device.
7. Main admin enters credentials to access admin profile dashboard.
8. Credentials are checked against those stored within Firebase.
9. Credentials don't match those stored within Firebase.
10. Admin is returned to the login screen to renter credentials.

Exceptional flow

E1 : <Delete Sub Admin>

7. Main admin opens application on a compatible device.
8. Main admin enters credentials to access admin profile dashboard.
9. Credentials are checked against those stored within Firebase.
10. Admin gains access to the Admin dashboard.
11. Options to delete sub Admin.
12. Sub admin deletion fed back to firebase for removal.

Termination

Application closes when user exits the application with automatic logout.

Post condition

Application automatically logs out.

Reason for Exclusions

Changes made to the application was primarily the exclusion of an administrator, I felt this was unnecessary as it was hard enough to create to different types of logins for both employers and employees. The admin and sub admin creation was going to cause larger complications and time restraints on an already restricted time project. In the end I had to settle to have the two main users. In future there is room to add an administrator for the application but as this is just a college project at the moment there is no need for the application and its content to be monitored.

3.2.3 Data requirements

In this section we will discuss the data requirements that are required to implement the key functionalities of the application

- Users require an email address or Google account to gain access to the application, as it is a android application a Google account is required to access the Play Store.
- User location as the application will store the users current location and displays jobs in the parameters the user sets.
- Records the desired field the user wishes to apply for jobs in, i.e. Medical, H.R, Tech, Finance.
- All the information will be stored within Google's Firebase NoSQL back end. Information stored will include User profiles, Job posts, Matches, Messages.
- Google firebase is joined to the main application using dependencies that are coded in the configuration files.

3.2.4 User requirements

- The purpose of Rolette is to give users an easy to use job search application. The application is portable and lightweight in terms of storage.
- The user is required to be the owner of an android device in order to install the application.
- The user is required to create a Google account to download the application from the Play Store.
- The Android device must be running at least version 4.1 Jellybean in order to run the application.
- The user is required to have a stable Internet connection in order to run the application.
- It is not necessary but to improve service the user may need to turn location services to find jobs in their surrounding area.
- User will be required to select from drop down menus the area of which the wish to apply for jobs.

3.2.5 Environmental requirements

Android device, running android version 4.1 Jellybean to test the running application on. Although Android Studio offers a test service on multiple devices, I feel the nature of the application field-testing would be useful.

Android Studio, this IDE will be used in the development and creation of this application and as android is a java language the application will be programmed in Java. Android studio also provides tools for debugging and testing as stated can offer testing on a variety of virtual Android devices for testing.

Firebase, this will be the real time database used to store our information gathered in the application. I will also use Firebase for hosting our application.

3.2.6 Non functional requirements

3.2.7 Performance/Response time requirement

Performance and response time are paramount in the development of this app, we will be using a real-time database powered by firebase, and this will help in the storing and retrieval of information when requested. We will also be hosting on Firebase each file uploaded is cached on SSDs at CDN edges around the world which delivers rapid returns of information. With regards response time approximately 1 second is considered to be the standard, whereas any delay of 10 seconds and above will lose the user's interest and attention. So we need the response times optimized as much as possible.

3.2.8 Availability requirement

As with any business, being available to operate is one of the most integral parts of sustaining a successful business. If there were any downtime for maintenance or system faults we would strive to have it a minimum of 3-days/ 72 hours downtime per year this would equate to 99% availability throughout a calendar year.

3.2.9 Recover requirement

AS we are going to be using Firebase as our real time database it offers the function of automated backups and rollback system thus making recoverability a standard process. In the event of disaster all information is handled by Firebase and is easily attainable.

3.2.10 Security requirement

With security in mind we implement a secure platform enforcing all accounts have a either a Google account or at least an email and password to access their accounts, Google firebase will handle recovery of lot information. Security questions, reset link or mobile number can set up Recovery/Retrieval of an account. Users will only be able to access the account to which corresponds with the credentials. Features will only be available to users with the correct corresponding account i.e., admin access only available to admin;s, job posting available to only employers, apply for jobs will only be accessible to employees. We will try to use some sort of encryption for the messaging service, which will keep messages secure from source to end point.

3.2.11 Maintainability requirement

The app will be kept up to date with added features through updates. Customers will be asked to report any bugs that may occur using the application. The report will consist of the version of the app, the version of android the user is operating on along with device. Though Android Studio tests app's on multiple devices not all aspects are infallible. Users will also be given the opportunity to give their feedback on features they may wish to see added to the app.

3.2.12 Portability requirement

The application will be based in android initially and plan to expand over to IOS. The app will run on any android device running 4.0 this will include mobile telephones and tablets.

3.2.13 Extendibility requirement

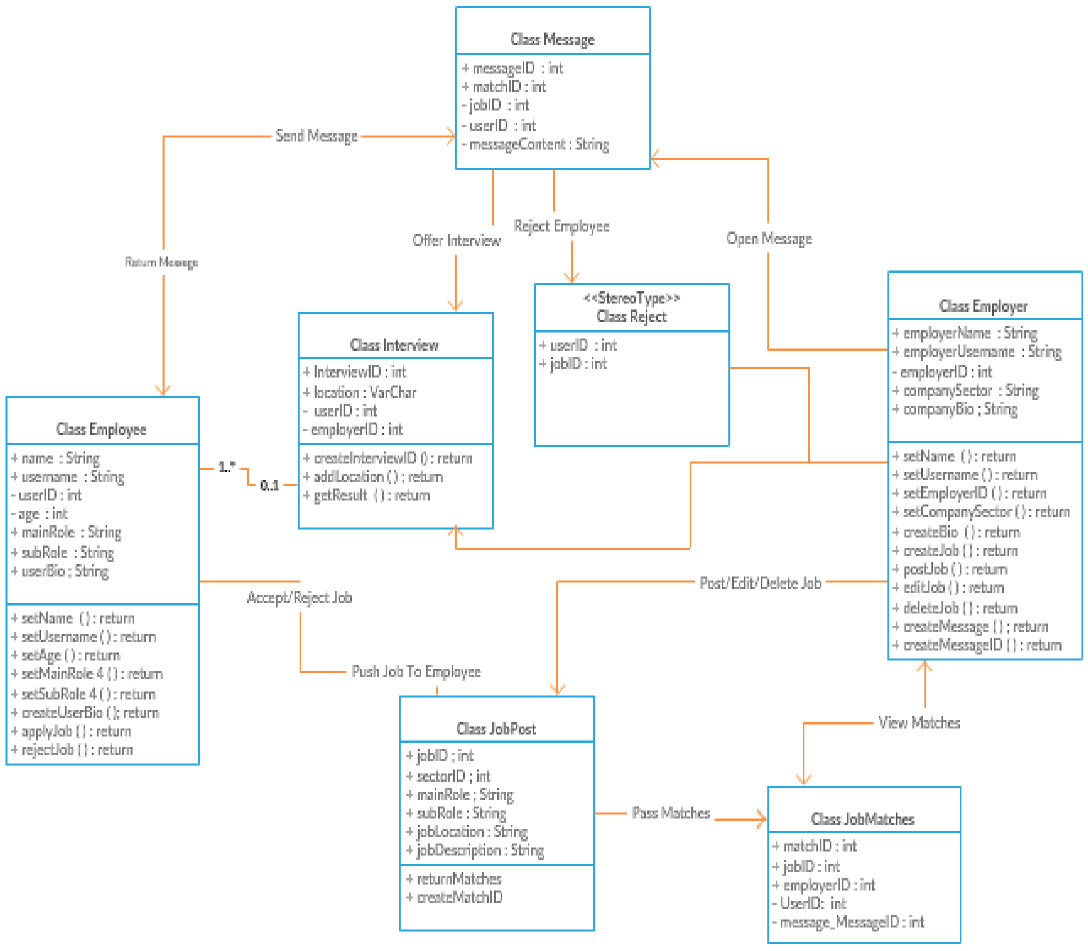
With regards extensibility there is lots of room to extend the app. There are already plans to utilize portable devices facilities such as calendar and location services to improve the service. These added features would be pushed out after the initial software is launched over updates. Again like maintainability we will read the feedback received from our customers to further improve the service.

3.2.14 Reusability requirement

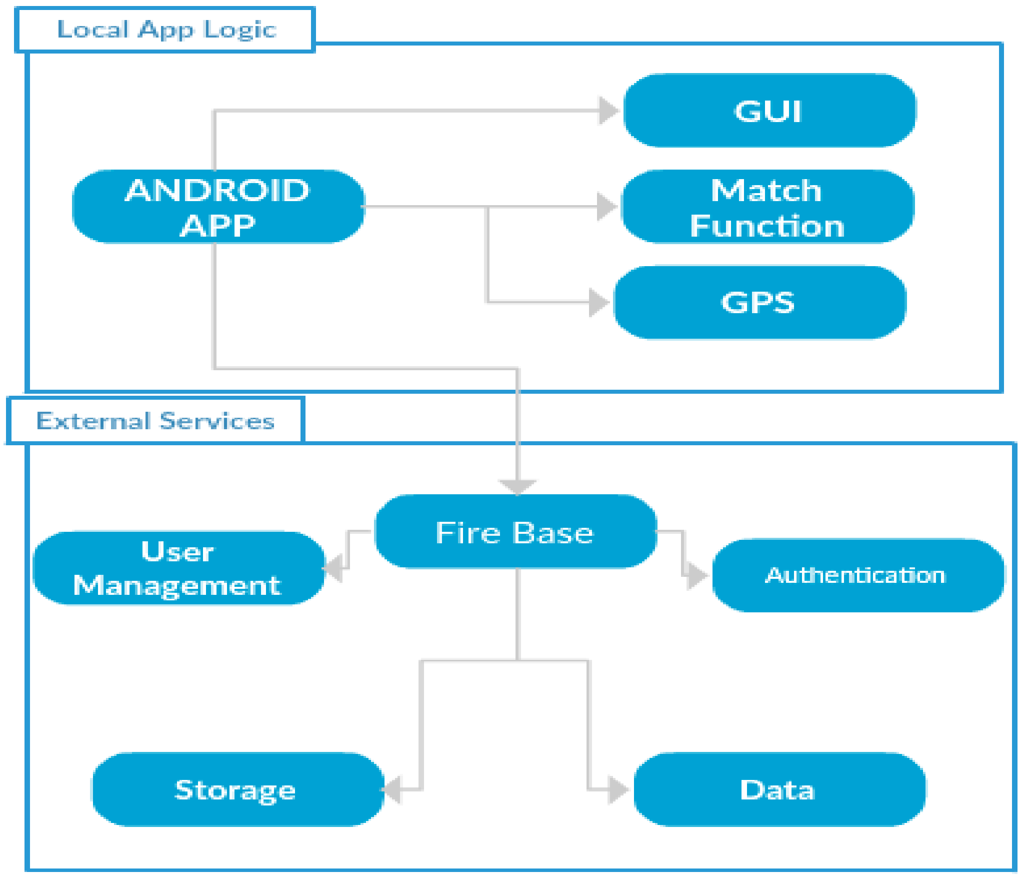
As the code will be done in android studio it will be very reusable over the android platform, but integrating it into other platforms such as IOS maybe difficult in terms of reusability. As there is not a common language between Apple IOS and Android. Also to create a desktop version we would have to start practically from the beginning again.

3.3 Design and Architecture

Class Diagram, in this application the two main classes are the users, which will connect to the services in the other classes such as message class and job post class. This is where the employer and employee classes will interact.



System Architecture Diagram, to help show the on internal logic layer of the application and the external sources that will handle the authentication aspects, data storage and user management.



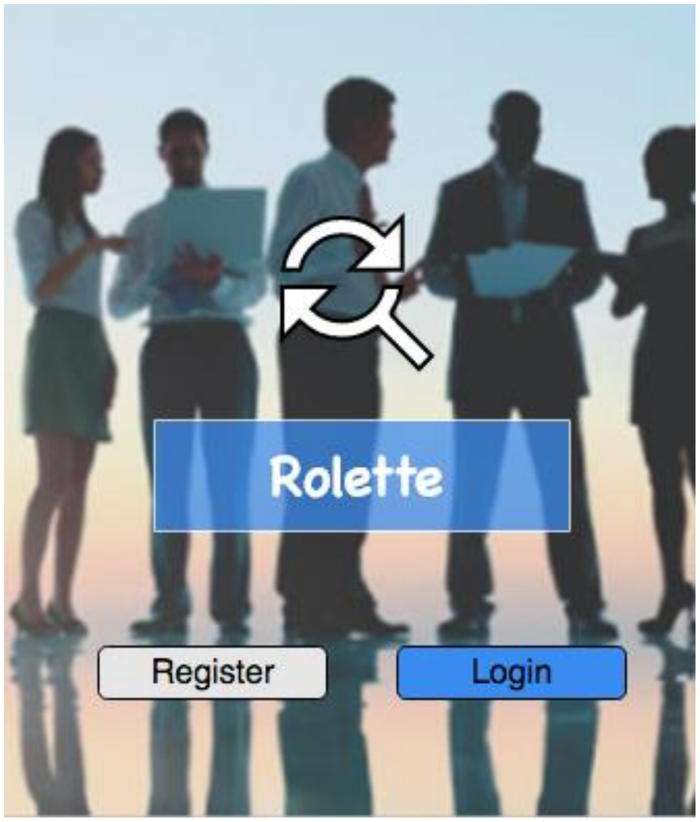
3.4 *Implementation*

As there has not been any application as of yet to implement any algorithms or classes, I do plan to implement the Google firebase as a real time database this will handle the majority of the users information. The service will also offer data analytics and crash reporting. Firebase along with sorting the user information will also handle the email authentication of the logins, it will also offer push notifications and remote configurations.

3.5 Graphical User Interface (GUI) Layout]

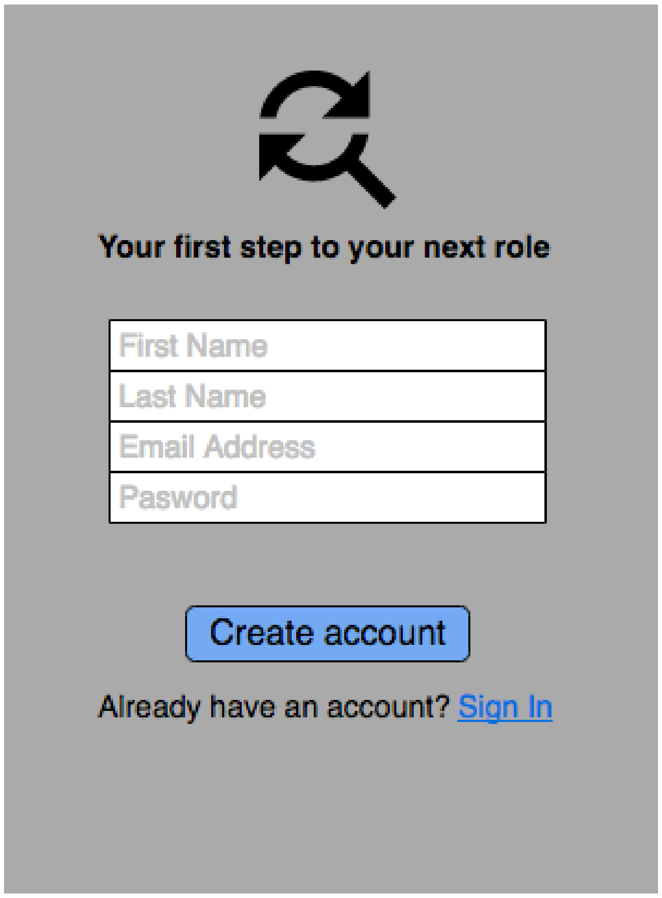
Mock ups

Home Screen



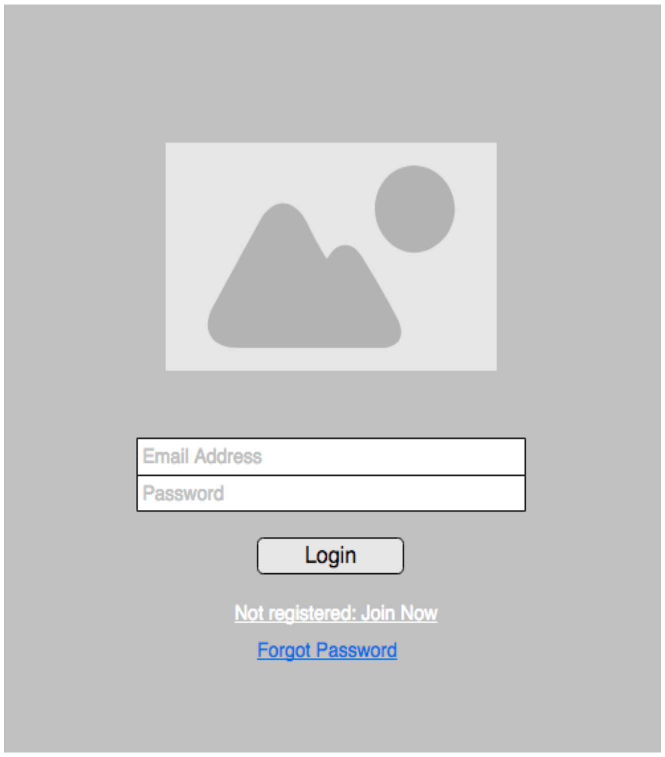
Here is where all users will be brought to when opening the application for the first time. Users can either login or register their details. If they chose to register for the first time they will be brought to the next screen.

Register Screen


Your first step to your next role
First Name
Last Name
Email Address
Pasword
Create account
Already have an account? Sign In

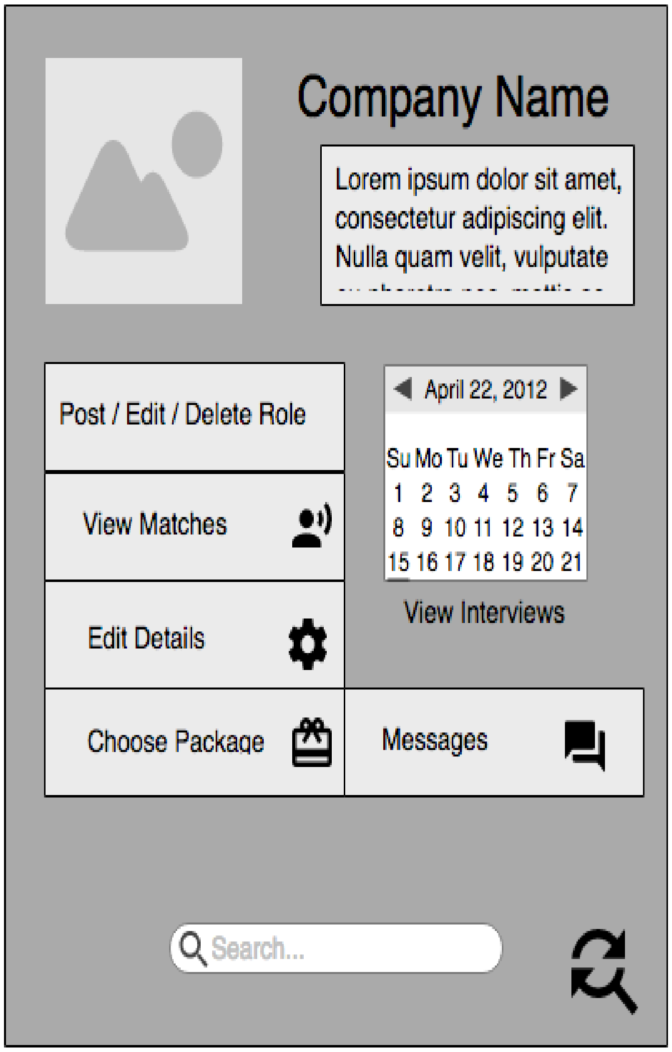
If the user has no account then we will have the create account, user will need to input their Name Email and a password. Though there is no option on the wireframe the user can sign in via a Google account. If the customer already has an existing account they can still return to the following frame, which will be the login page.

Login Screen



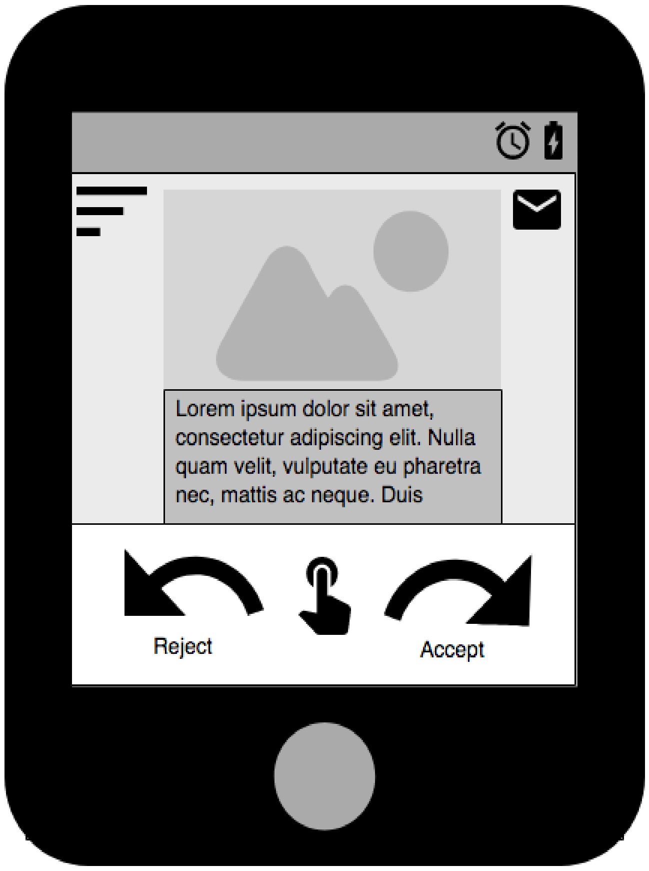
Here we have the main login page, depending if you are an employer or employee you will be brought to two different dashboards in the following screens. Users also have the options to either register it they don't have an account. If they have forgotten their password they can use the "Forgot Password" link to gain access to the account.

Employer Profile



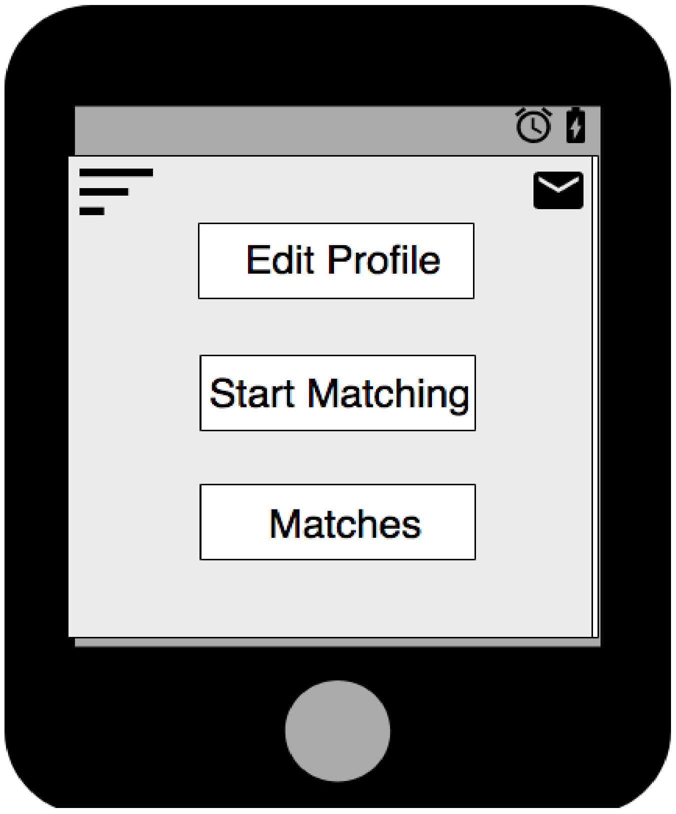
Here is the main dashboard for employers; they will have an array of options to choose from here in the dashboard. They can edit their company Bio; have options on posting new roles or editing and deleting existing roles. Choose a package plan with regards the usage within the app. View their job matches and start messaging with applicant.

Employee Search


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis
Reject
Accept

This is the main home screen layout for the employee user. We will see the job posting for the user being pushed here in a simple GUI. The user will swipe left to reject or swipe right to register their interest. In the role. In the top right hand corner we can see the messaging service. Where on the Top right the user will be able to return to their main dashboard.

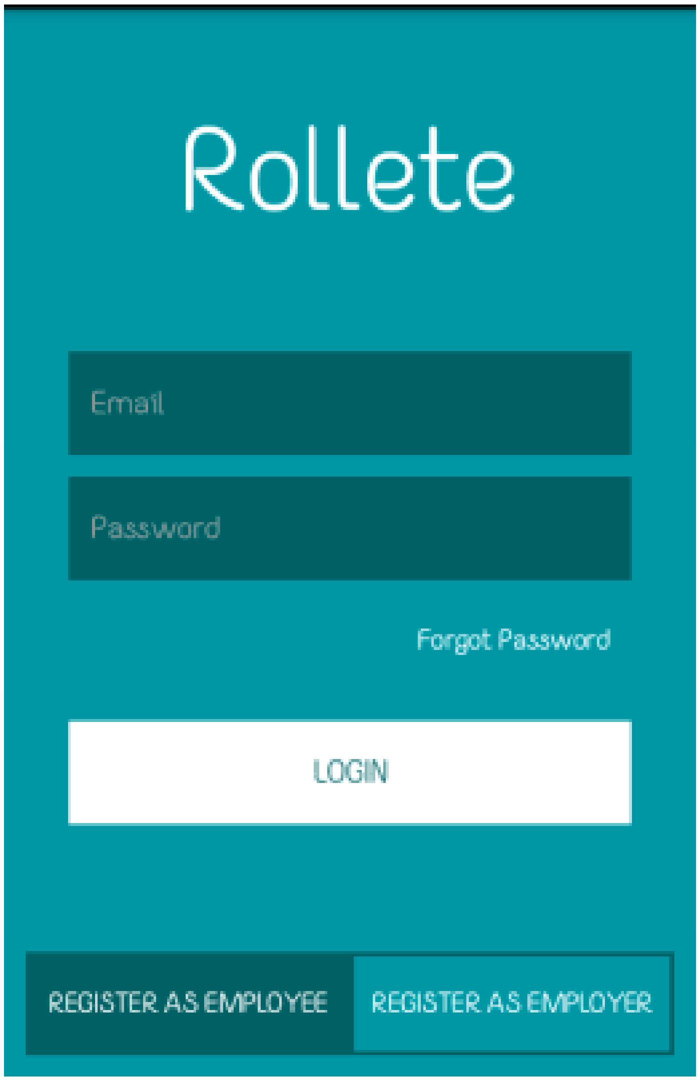
Employee Menu



This is the main Employee user dashboard here they can edit their profile return to the matching screen or view their existing matches. User can also access their messaging service here also.

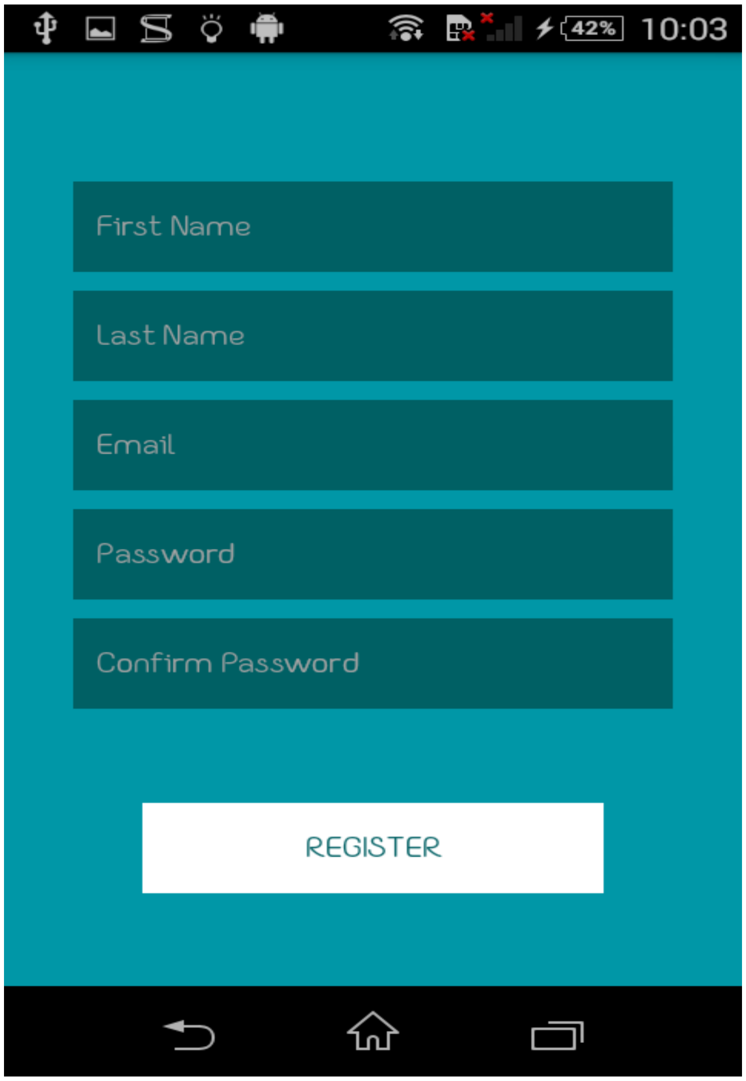
Finished GUI

Login/Register



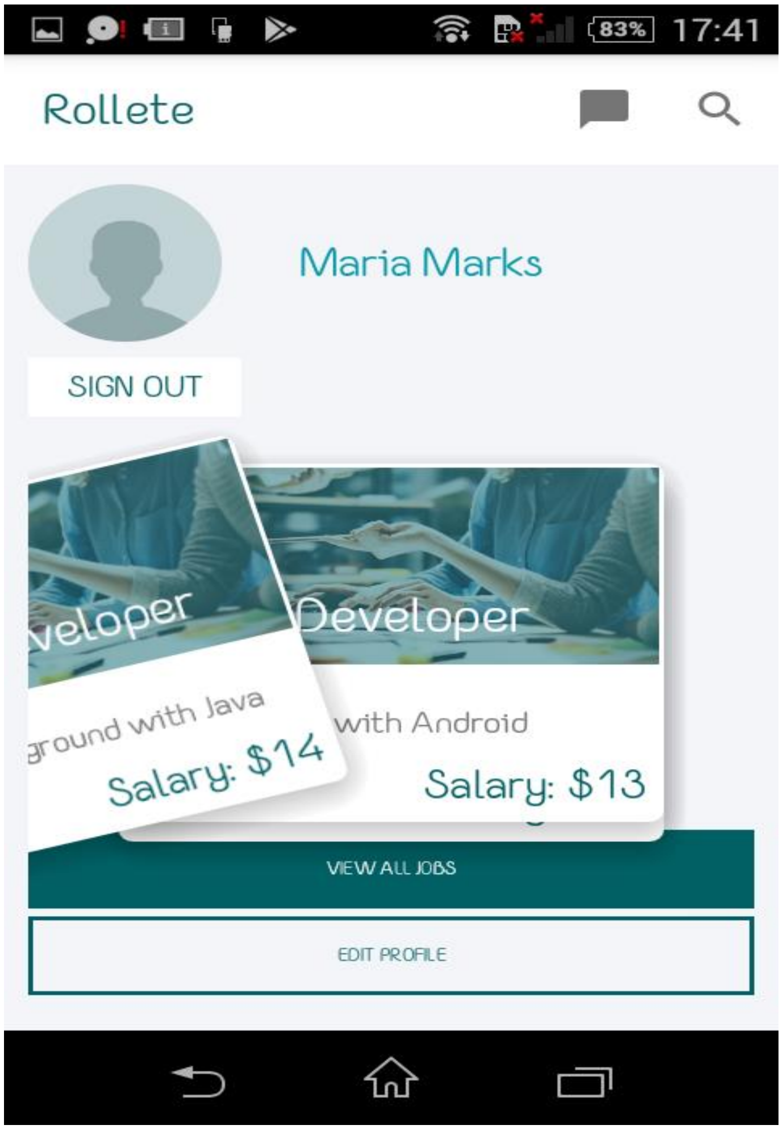
As we can see in this screenshot shot user can either register as an employee or an employer or just login with the credentials they have already registered with, that will be authenticated against those stored in Firebase. If the details match the user will then be brought to either the Employee dashboard or an Employer Dashboard.

Registeration



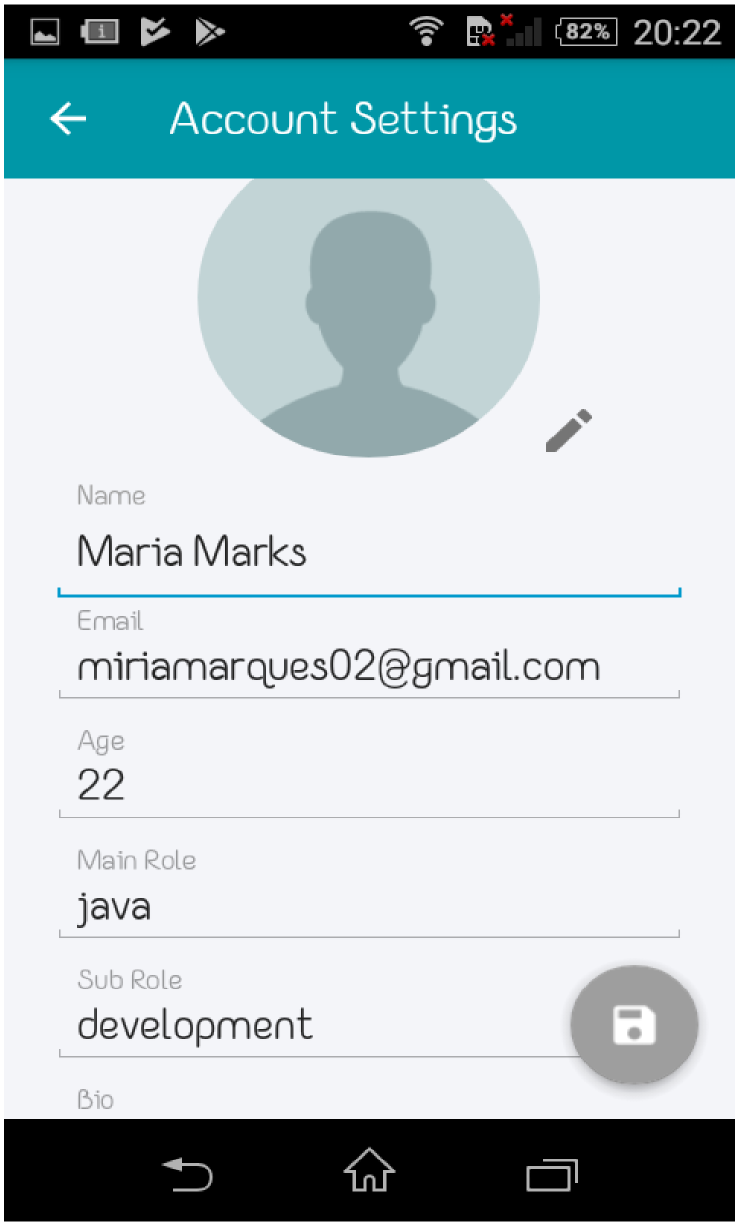
Here we can see the registration page for either an employer or an employee. Once complete the information is stored in the Firebase database where it is given a unique user ID.

Employee Dashboard



Here we have the main employee home where the user can view the jobs that have been posted in by prospective employers. Users can swipe away jobs they do not find interest them. This is the first screen to be found once logging in correctly. From this screen users have the option to edit their profile where they can fill the fields with information on themselves or update the existing information already stored.

Edit Employee Profile



Name
Maria Marks

Email
miriamarques02@gmail.com

Age
22

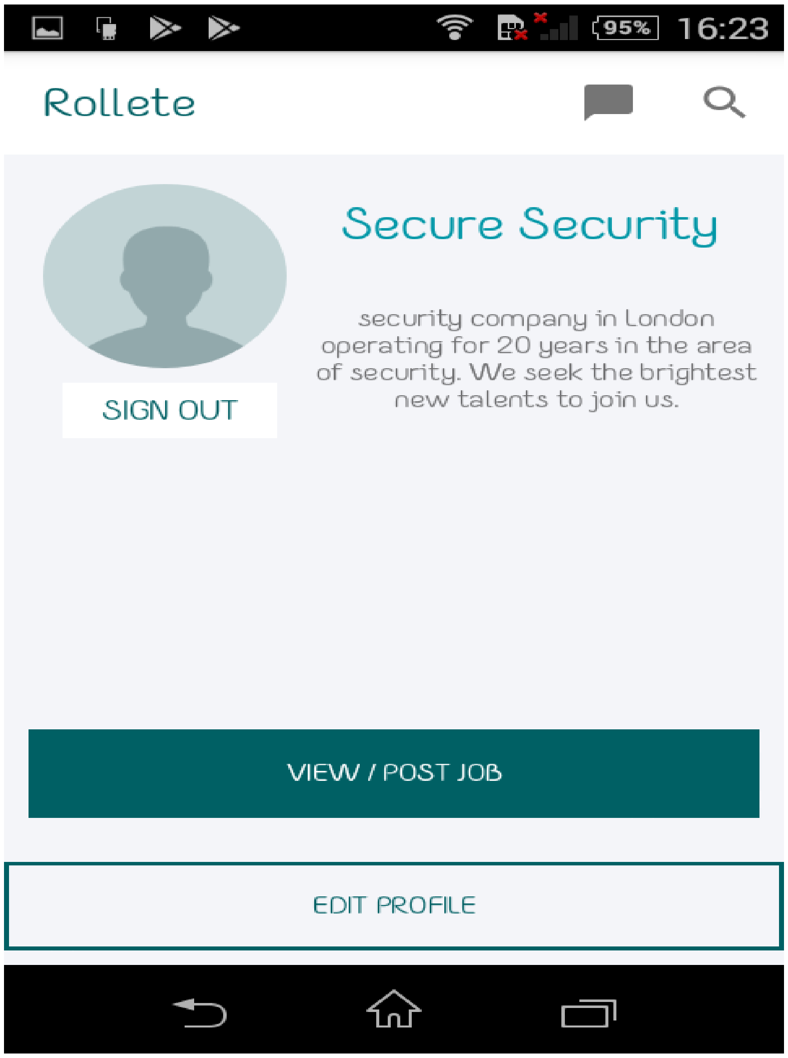
Main Role
java

Sub Role
development

Bio

Here we have the edit profile option for the employee. Employees can either enter their details upon registering or they can update their current details. All of these details are updated in real-time to the Firebase database. Fields to be filled out include Name, Age, Main Role, Sub Role and a Bio on themselves that would have a piece on their experience and background.

Employer Dashboard



Secure Security

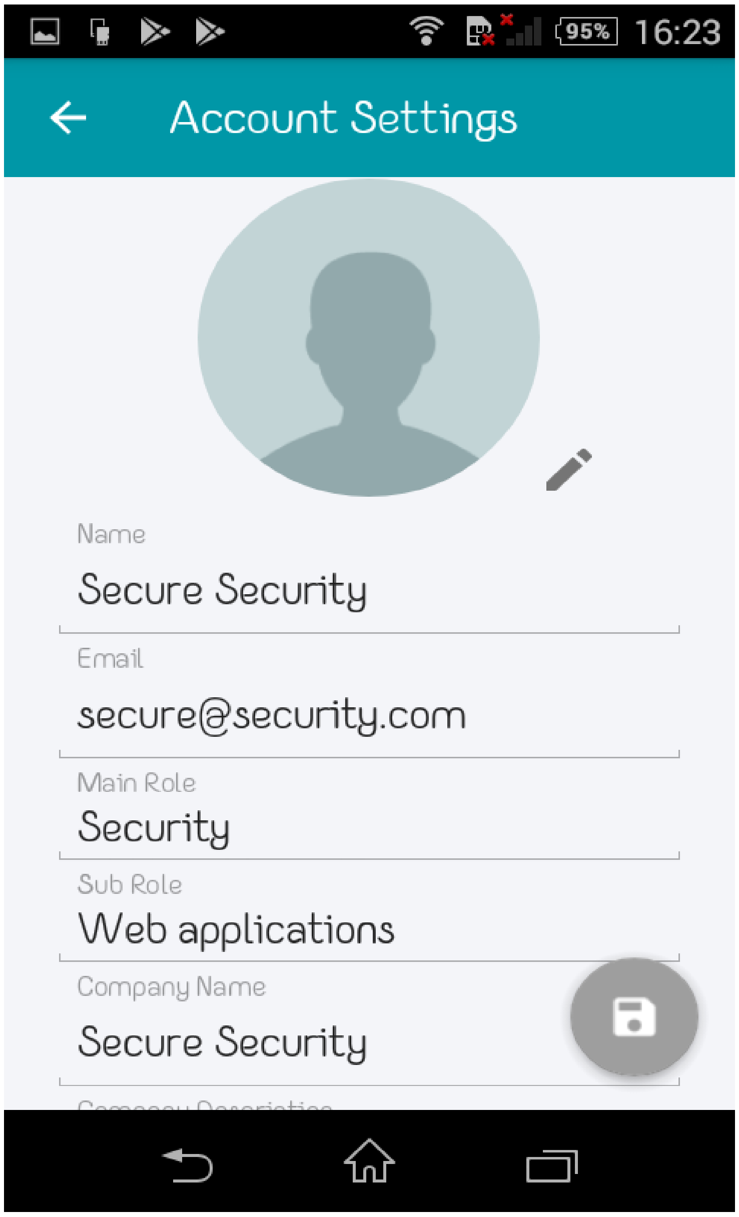
security company in London operating for 20 years in the area of security. We seek the brightest new talents to join us.

After user has been registered or logged in the main menu page will be brought to his screen. The main menu page will contain following options.

- View/Post Job
- Edit Profile

In View/Post Job this is where an employer can post a job to the employee domain to be viewed. In Edit Profile an Employer can either enter their details if recently registered with the application or update their existing details stored within the application. These details are then passed over to be stored in the real time Firebase database.

Edit Employer Profile



Name
Secure Security

Email
secure@security.com

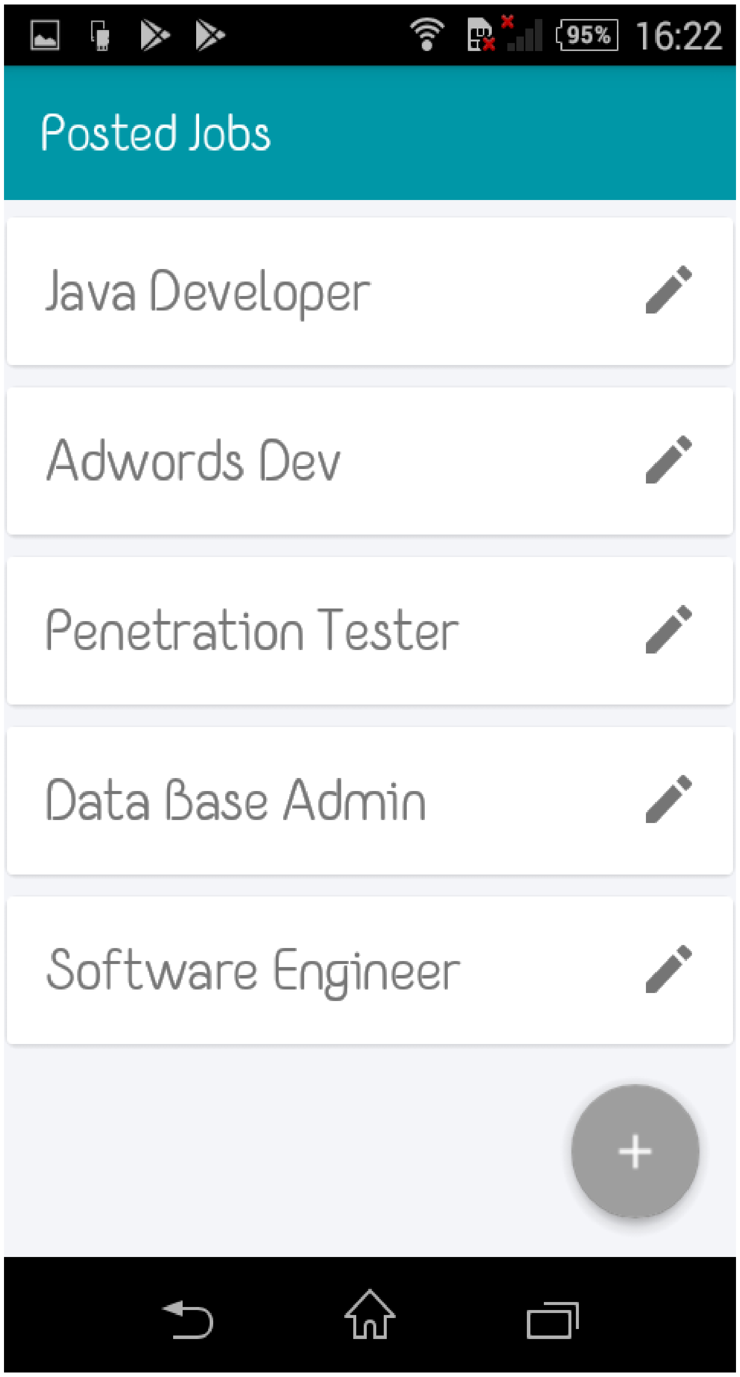
Main Role
Security

Sub Role
Web applications

Company Name
Secure Security

Much the same as the Employee Edit profile the new users can wither enter their details upon registering or they can update their current details. The fields here are the Company name, Company Email, Main role and Sub Roles, Bio on a description on the company's background.

View/Edit Job



Posted Jobs

Java Developer

Adwords Dev

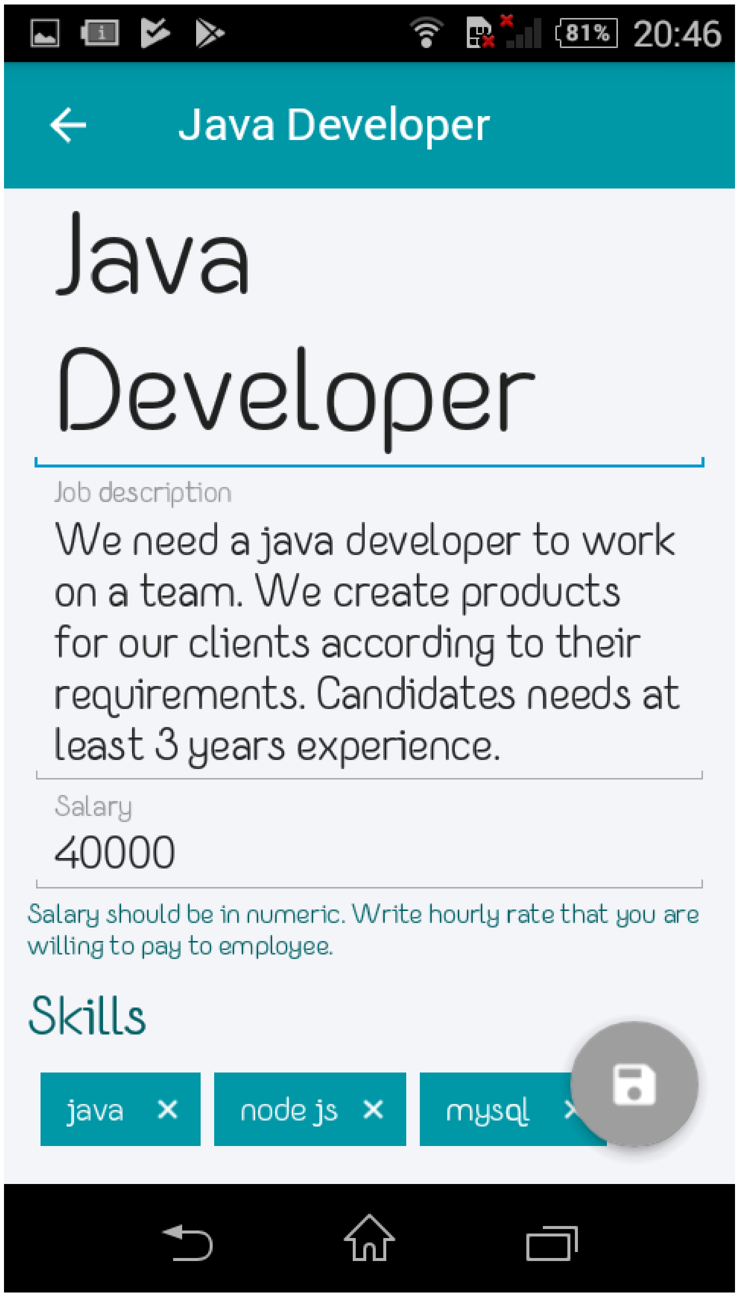
Penetration Tester

Data Base Admin

Software Engineer

Here is the screen where the employers can edit their existing job posts or add new job posts to the application.

Job Post



Java Developer

Job description

We need a java developer to work on a team. We create products for our clients according to their requirements. Candidates needs at least 3 years experience.

Salary

40000

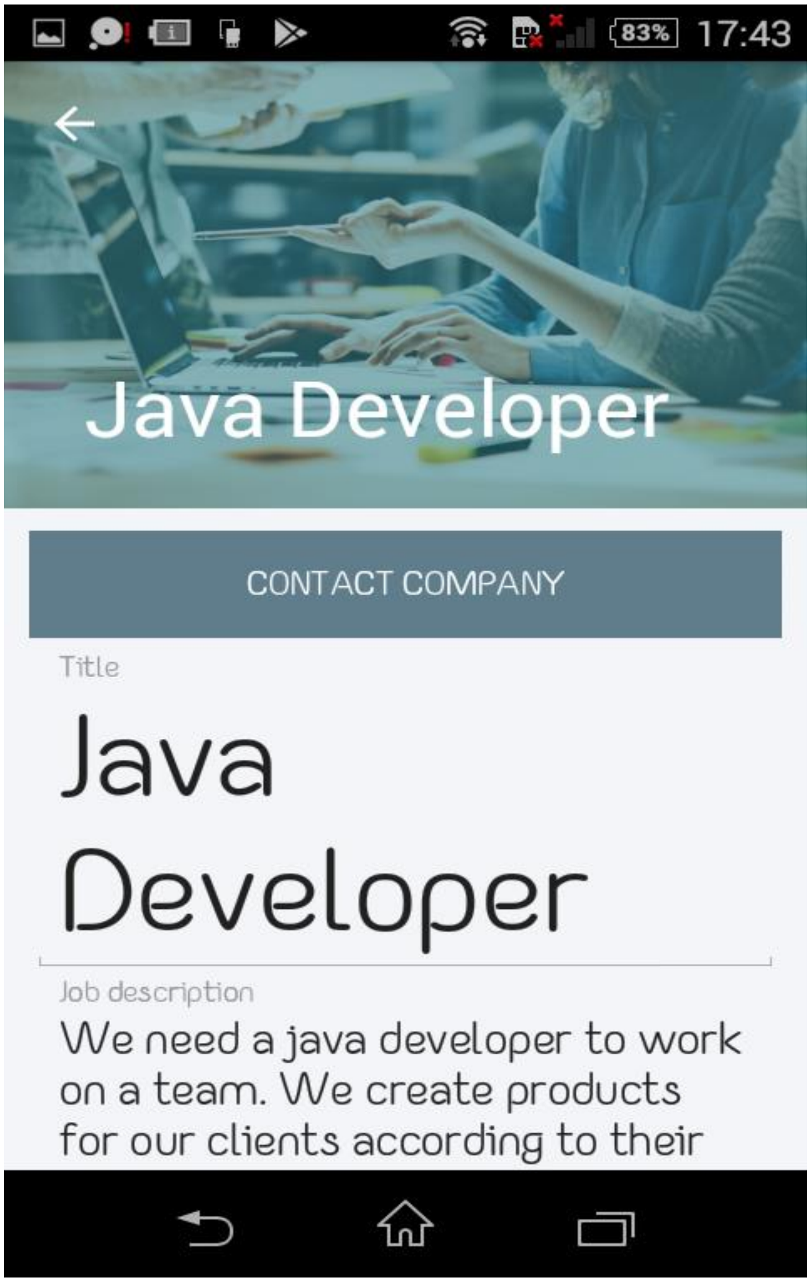
Salary should be in numeric. Write hourly rate that you are willing to pay to employee.

Skills

java ✕ node js ✕ mysql ✕

This is the screen an employer will be guided to when they are posting or editing an existing job. The screen has fields to be completed, Job Title, Job Description, Salary, Skills applicant is required to have. The Skills where set to tags I wanted to use to match the employees and the jobs together.

View Job Post "Employee"



Java Developer

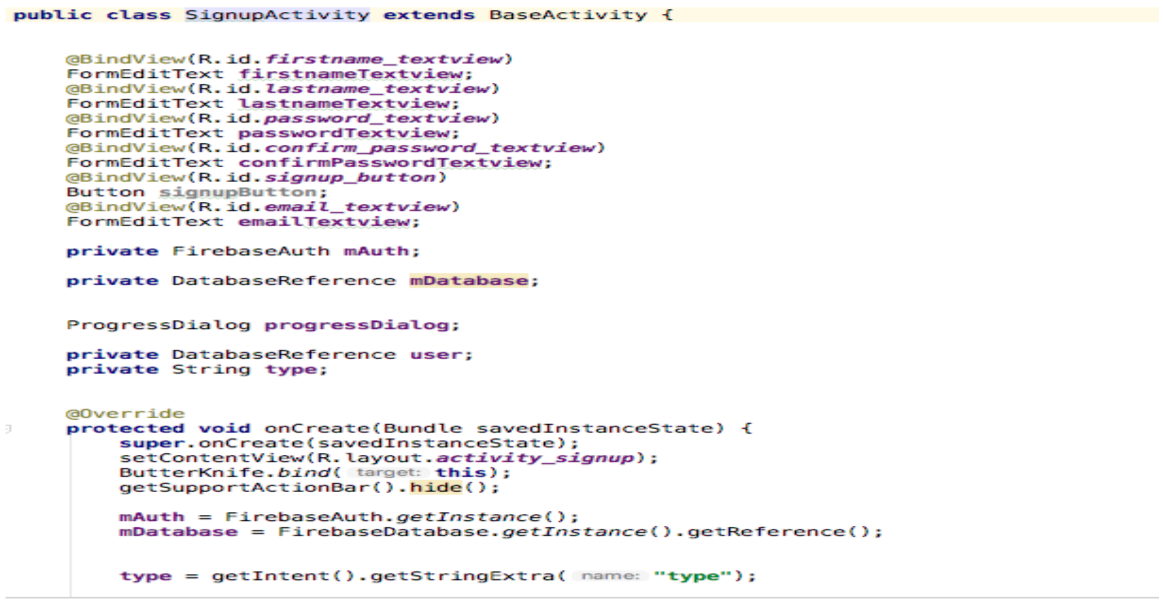
We need a java developer to work on a team. We create products for our clients according to their

When an employee chooses a role they are interested in they can view it in more detail by tapping on the view option rather than swiping it away. When the employee taps on the contact company they can make direct contact with the company that posted the job.

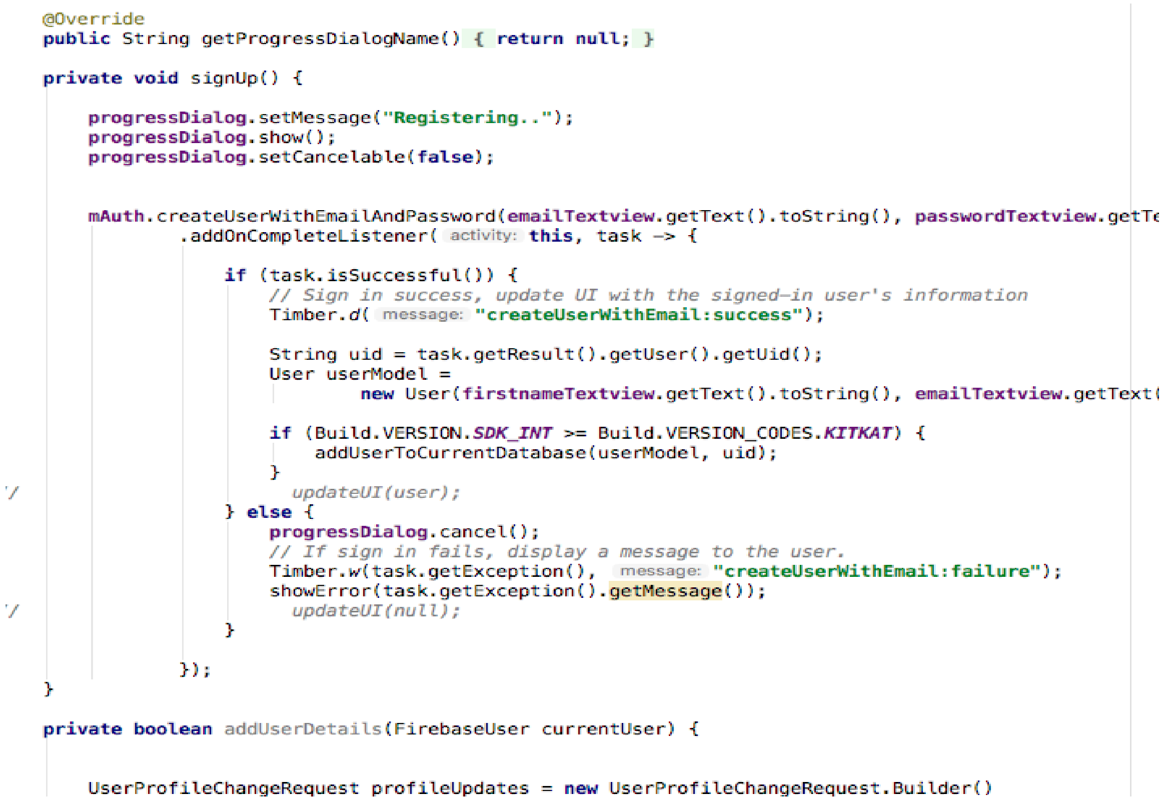
3.6 Application Algorithms and methods

Sign Up

Sign up and authentication for the application will be done using Google Firebase provided methods of authentication and registration

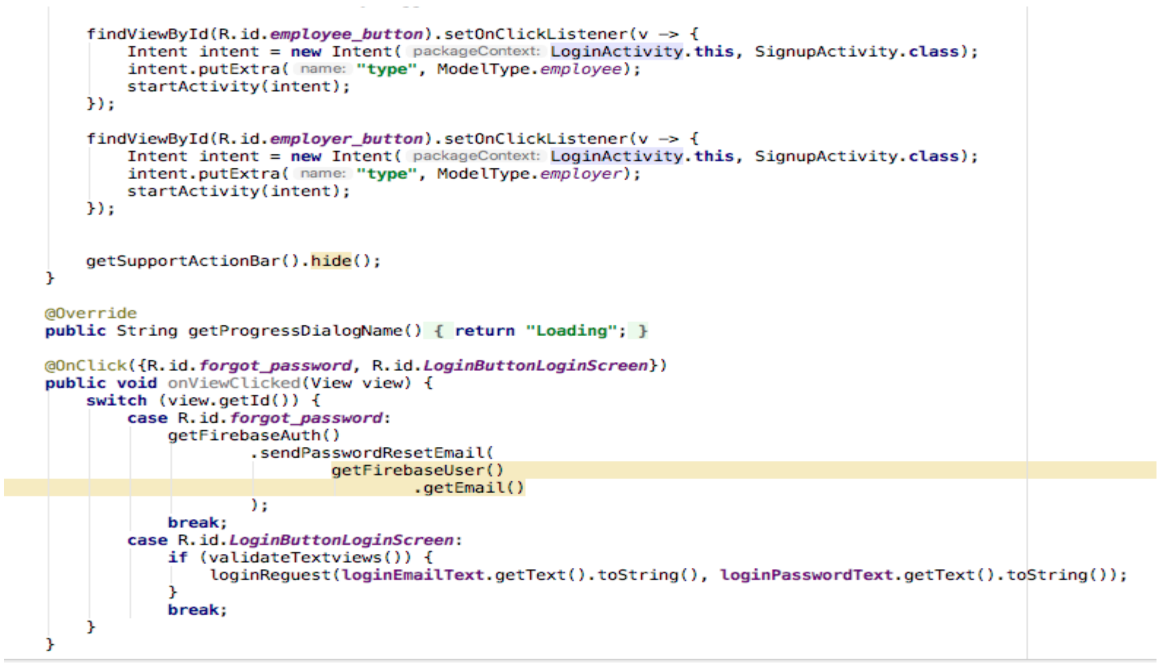


```
public class SignupActivity extends BaseActivity {

    @BindView(R.id.firstname_textview)
    FormEditText firstnameTextview;
    @BindView(R.id.lastname_textview)
    FormEditText lastnameTextview;
    @BindView(R.id.password_textview)
    FormEditText passwordTextview;
    @BindView(R.id.confirm_password_textview)
    FormEditText confirmPasswordTextview;
    @BindView(R.id.signup_button)
    Button signupButton;
    @BindView(R.id.email_textview)
    FormEditText emailTextview;

    private FirebaseAuth mAuth;

    private DatabaseReference mDatabase;


    ProgressDialog progressDialog;

    private DatabaseReference user;
    private String type;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        ButterKnife.bind( target: this);
        getSupportActionBar().hide();

        mAuth = FirebaseAuth.getInstance();
        mDatabase = FirebaseDatabase.getInstance().getReference();


        type = getIntent().getStringExtra( name: "type");
```

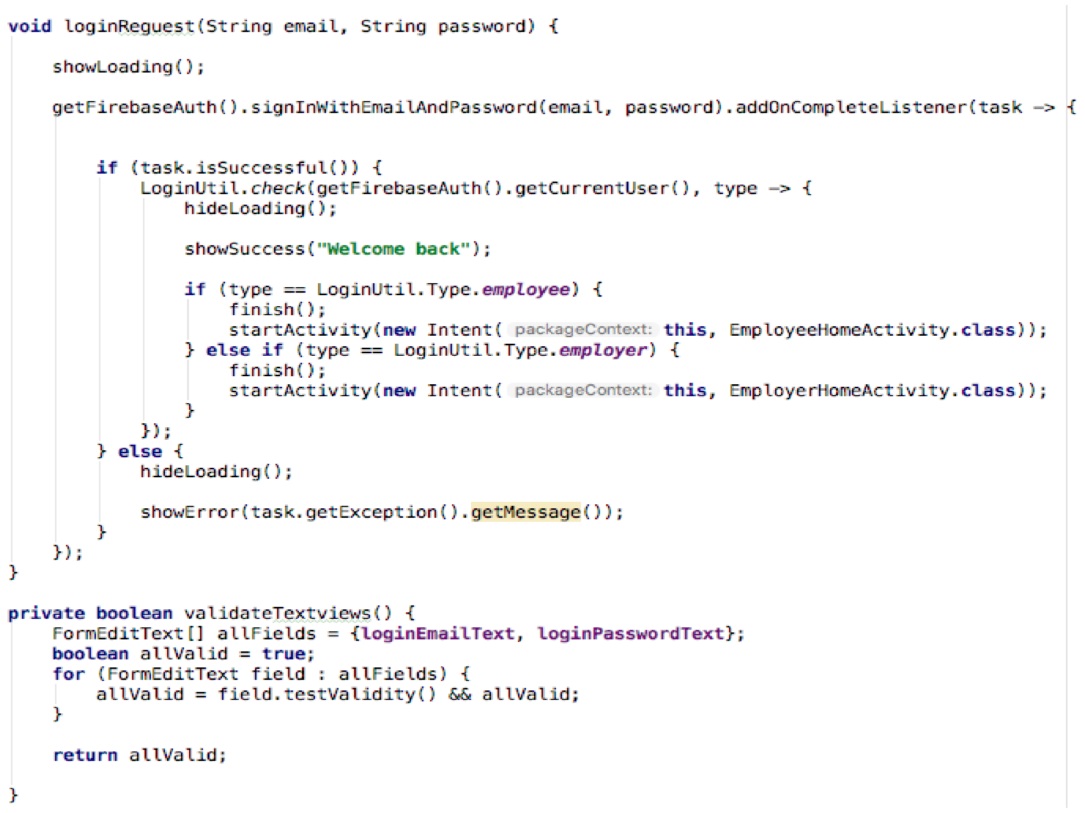


```
@Override
public String getProgressDialogName() { return null; }

private void signUp() {

    progressDialog.setMessage("Registering..");
    progressDialog.show();
    progressDialog.setCancelable(false);


    mAuth.createUserWithEmailAndPassword(emailTextview.getText().toString(), passwordTextview.getTe
            .addOnCompleteListener( activity: this, task -> {

                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    Timber.d( message: "createUserWithEmail:success");

                    String uid = task.getResult().getUser().getUid();
                    User userModel =
                            new User(firstnameTextview.getText().toString(), emailTextview.getText(

                    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
                        addUserToCurrentDatabase(userModel, uid);
                    }
                      updateUI(user);
                } else {
                    progressDialog.cancel();
                    // If sign in fails, display a message to the user.
                    Timber.w(task.getException(),  message: "createUserWithEmail:failure");
                    showError(task.getException().getMessage());
                      updateUI(null);
                }

            });
}

private boolean addUserDetails(FirebaseUser currentUser) {


    UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()
```

Login

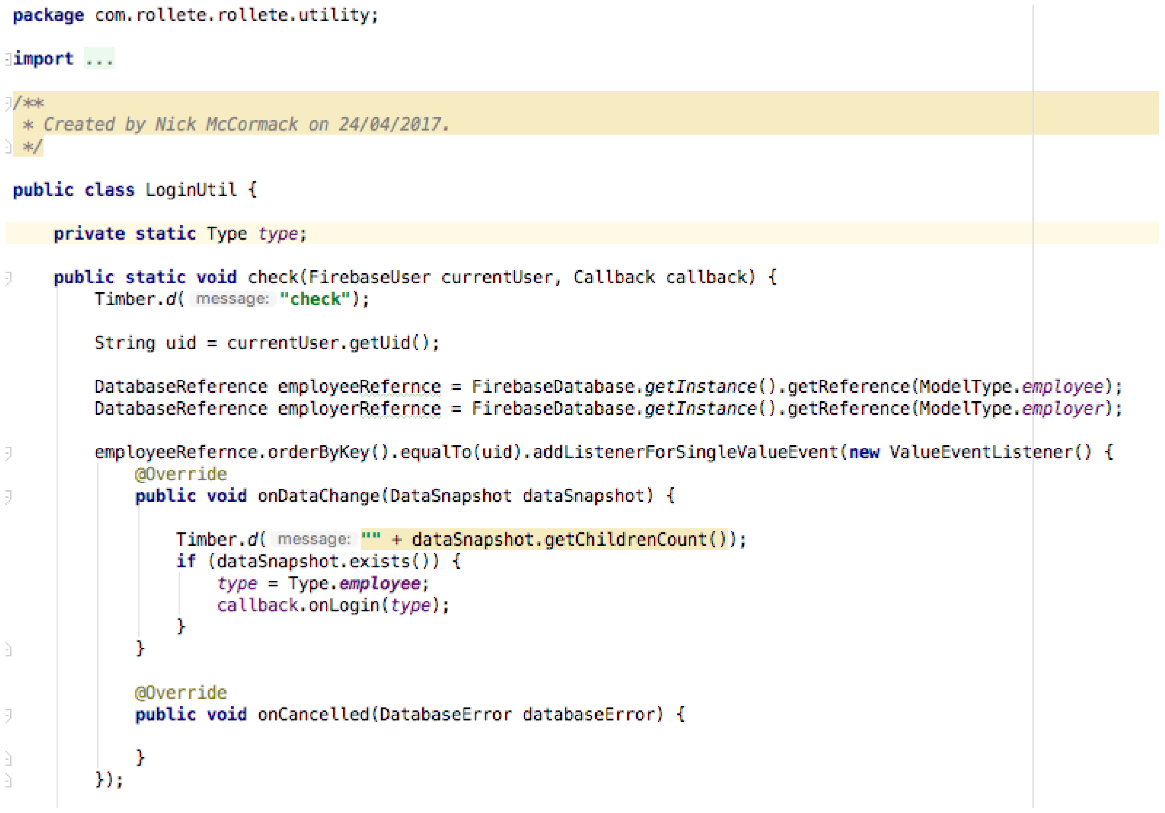


```
        findViewById(R.id.employee_button).setOnClickListener(v -> {
            Intent intent = new Intent( packageContext: LoginActivity.this, SignupActivity.class);
            intent.putExtra( name: "type", ModelType.employee);
            startActivity(intent);
        });

        findViewById(R.id.employer_button).setOnClickListener(v -> {
            Intent intent = new Intent( packageContext: LoginActivity.this, SignupActivity.class);
            intent.putExtra( name: "type", ModelType.employer);
            startActivity(intent);
        });

        getSupportActionBar().hide();
    }

    @Override
    public String getProgressDialogName() { return "Loading"; }

    @OnClick({R.id.forgot_password, R.id.LoginButtonLoginScreen})
    public void onViewClicked(View view) {
        switch (view.getId()) {
            case R.id.forgot_password:
                getFirebaseAuth()
                        .sendPasswordResetEmail(
                                getFirebaseUser()
                                        .getEmail()
                        );
                break;
            case R.id.LoginButtonLoginScreen:
                if (validateTextviews()) {
                    loginRequest(loginEmailText.getText().toString(), loginPasswordText.getText().toString());
                }
                break;
        }
    }
```

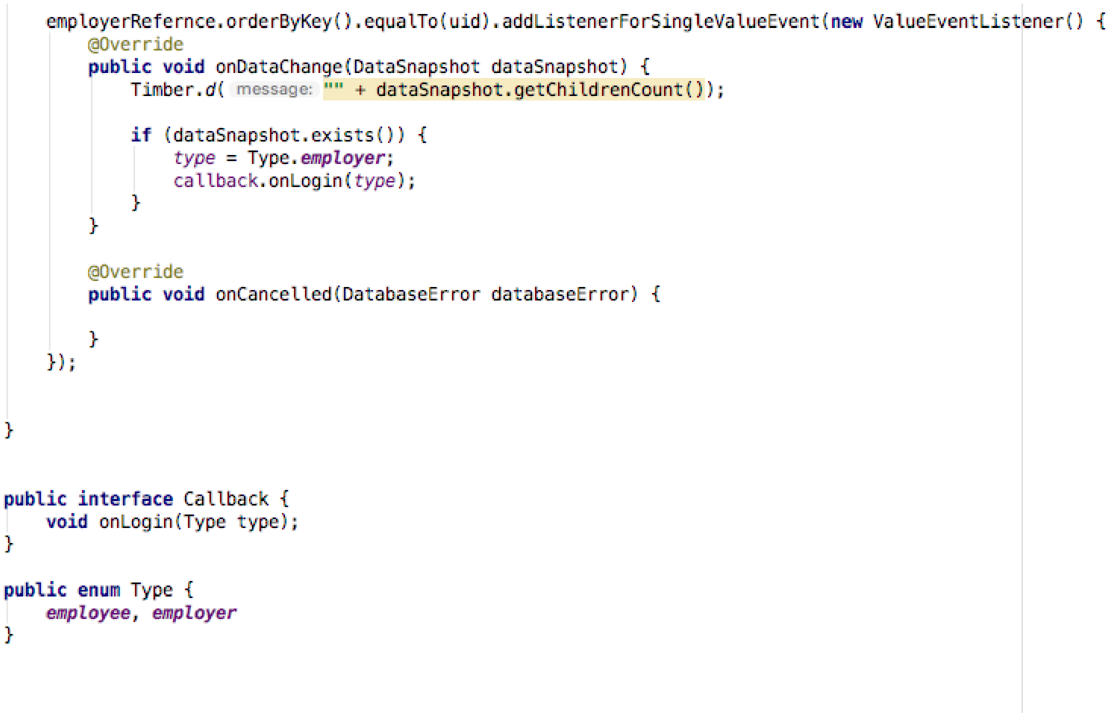


```
    void loginRequest(String email, String password) {

        showLoading();

        getFirebaseAuth().signInWithEmailAndPassword(email, password).addOnCompleteListener(task -> {


            if (task.isSuccessful()) {
                LoginUtil.check(getFirebaseAuth().getCurrentUser(), type -> {
                    hideLoading();

                    showSuccess("Welcome back");

                    if (type == LoginUtil.Type.employee) {
                        finish();
                        startActivity(new Intent( packageContext: this, EmployeeHomeActivity.class));
                    } else if (type == LoginUtil.Type.employer) {
                        finish();
                        startActivity(new Intent( packageContext: this, EmployerHomeActivity.class));
                    }
                });
            } else {
                hideLoading();

                showError(task.getException().getMessage());
            }
        });
    }

    private boolean validateTextviews() {
        FormEditText[] allFields = {loginEmailText, loginPasswordText};
        boolean allValid = true;
        for (FormEditText field : allFields) {
            allValid = field.testValidity() && allValid;
        }

        return allValid;

    }
```

Determine if Employee or Employer

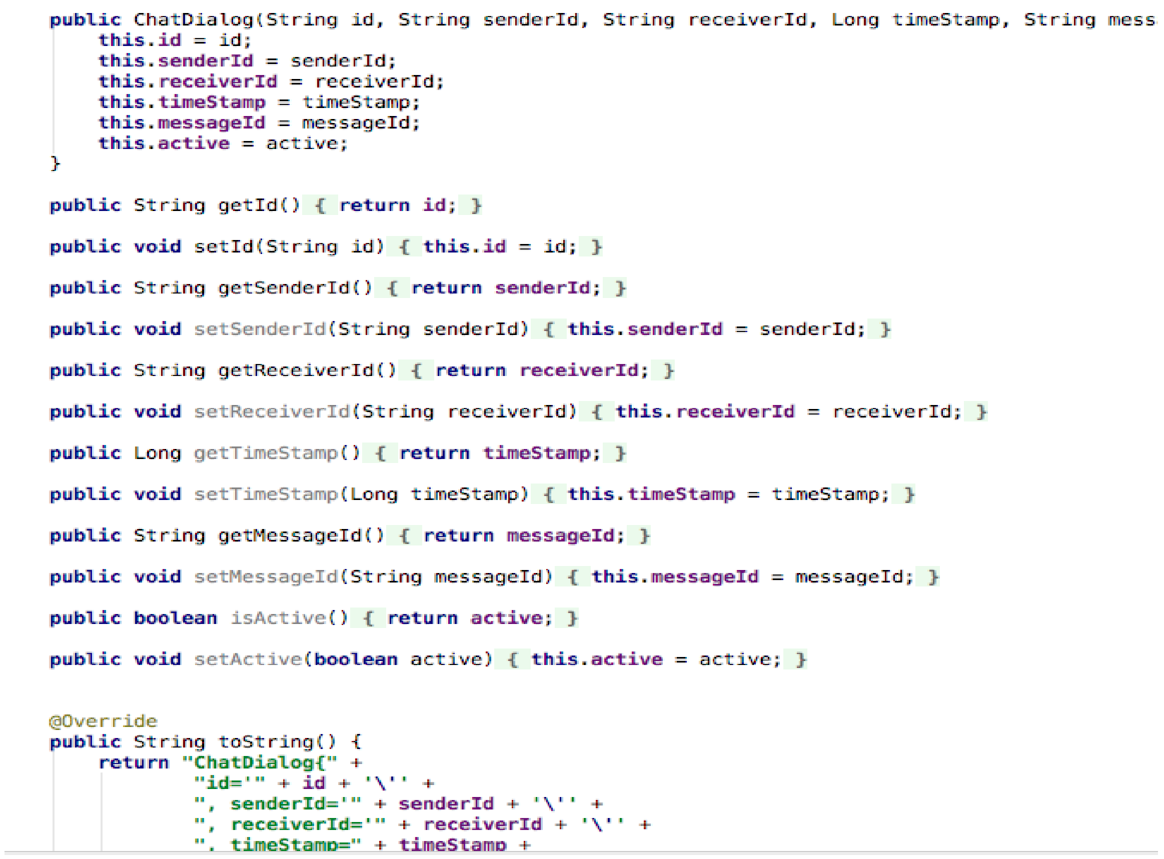


```
package com.rollete.rollete.utility;

import ...

/**
 * Created by Nick McCormack on 24/04/2017.
 */

public class LoginUtil {

    private static Type type;

    public static void check(FirebaseUser currentUser, Callback callback) {
        Timber.d( message: "check");

        String uid = currentUser.getUid();

        DatabaseReference employeeRefernce = FirebaseDatabase.getInstance().getReference(ModelType.employee);
        DatabaseReference employerRefernce = FirebaseDatabase.getInstance().getReference(ModelType.employer);

        employeeRefernce.orderByKey().equalTo(uid).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                Timber.d( message: "" + dataSnapshot.getChildrenCount());
                if (dataSnapshot.exists()) {
                    type = Type.employee;
                    callback.onLogin(type);
                }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
```

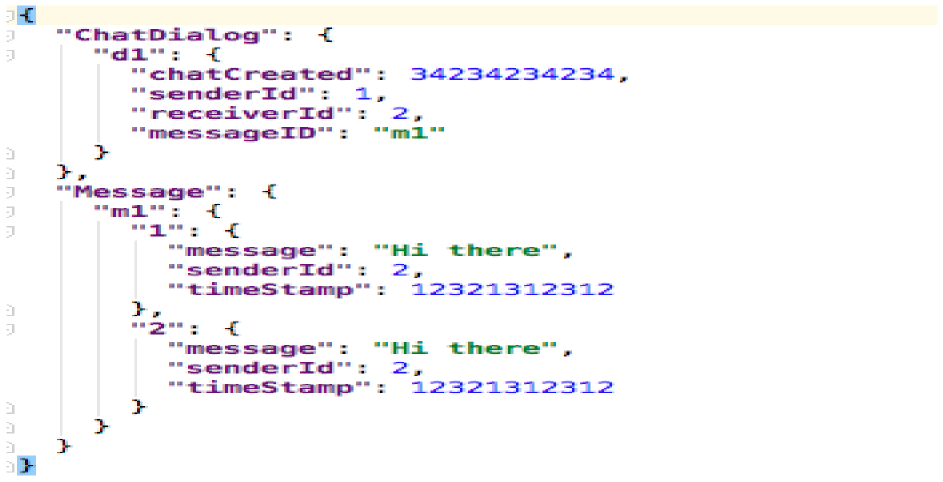


```
        employerRefernce.orderByKey().equalTo(uid).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                Timber.d( message: "" + dataSnapshot.getChildrenCount());

                if (dataSnapshot.exists()) {
                    type = Type.employer;
                    callback.onLogin(type);
                }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });


    }


    public interface Callback {
        void onLogin(Type type);
    }

    public enum Type {
        employee, employer
    }
```

Chat Function

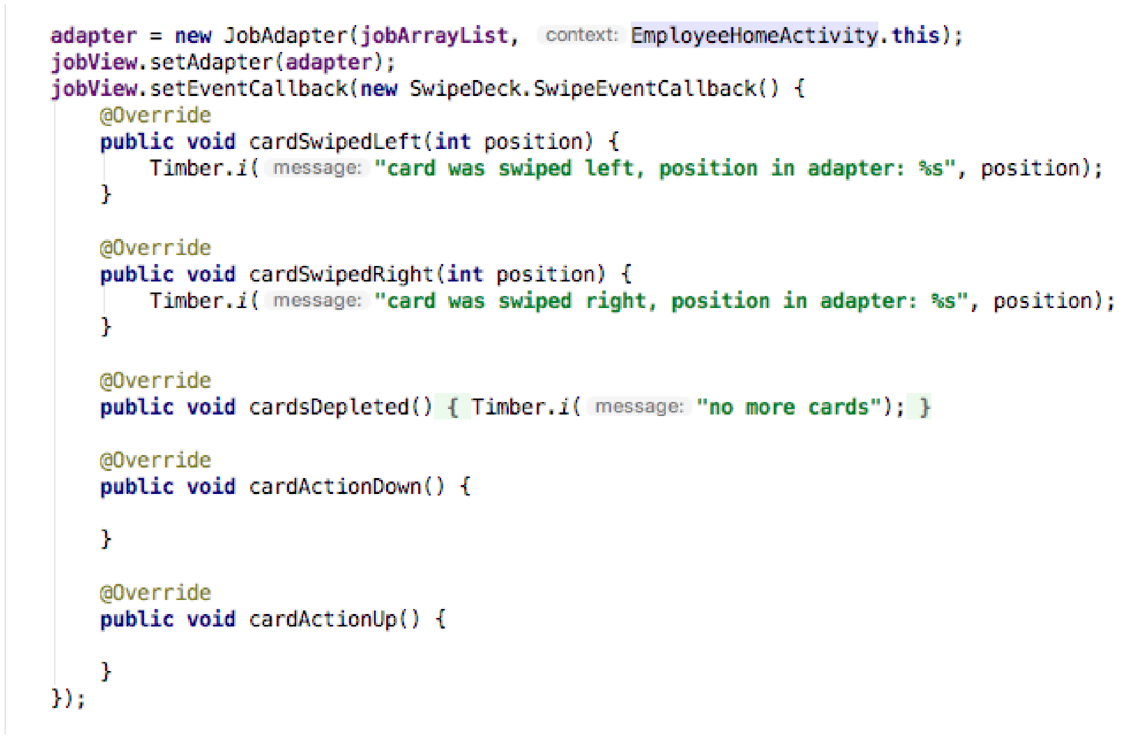


```
public ChatDialog(String id, String senderId, String receiverId, Long timeStamp, String mess
    this.id = id;
    this.senderId = senderId;
    this.receiverId = receiverId;
    this.timeStamp = timeStamp;
    this.messageId = messageId;
    this.active = active;
}

public String getId() { return id; }

public void setId(String id) { this.id = id; }

public String getSenderId() { return senderId; }

public void setSenderId(String senderId) { this.senderId = senderId; }

public String getReceiverId() { return receiverId; }

public void setReceiverId(String receiverId) { this.receiverId = receiverId; }

public Long getTimeStamp() { return timeStamp; }

public void setTimeStamp(Long timeStamp) { this.timeStamp = timeStamp; }

public String getMessageId() { return messageId; }

public void setMessageId(String messageId) { this.messageId = messageId; }

public boolean isActive() { return active; }

public void setActive(boolean active) { this.active = active; }


@Override
public String toString() {
    return "ChatDialog{" +
            "id='" + id + '\'' +
            ", senderId='" + senderId + '\'' +
            ", receiverId='" + receiverId + '\'' +
            ", timeStamp=" + timeStamp +
```



```
{
  "ChatDialog": {
    "d1": {
      "chatCreated": 34234234234,
      "senderId": 1,
      "receiverId": 2,
      "messageID": "m1"
    }
  },
  "Message": {
    "m1": {
      "1": {
        "message": "Hi there",
        "senderId": 2,
        "timeStamp": 12321312312
      },
      "2": {
        "message": "Hi there",
        "senderId": 2,
        "timeStamp": 12321312312
      }
    }
  }
}
```

Job Posts Swipe Deck



```
adapter = new JobAdapter(jobArrayList,  context: EmployeeHomeActivity.this);
jobView.setAdapter(adapter);
jobView.setEventCallback(new SwipeDeck.SwipeEventCallback() {
    @Override
    public void cardSwipedLeft(int position) {
        Timber.i( message: "card was swiped left, position in adapter: %s", position);
    }

    @Override
    public void cardSwipedRight(int position) {
        Timber.i( message: "card was swiped right, position in adapter: %s", position);
    }

    @Override
    public void cardsDepleted() { Timber.i( message: "no more cards"); }

    @Override
    public void cardActionDown() {

    }

    @Override
    public void cardActionUp() {

    }
});
```

3.6.1 Testing

With regards testing the Android Studio IDE has the internal capabilities to test applications using the Junit frameworks. Within these tests we can carry out both local unit tests and also instrumented unit tests. Local tests are run on a local Java Virtual Machine while instrumented are run in an android device. Tests carried out have helped improve functionality of the application. I have carried out integration testing as well to conclude if the sign up and login features are behaving correctly . I have also carried out field-testing by taking the app out into the real world and documented the response it has within an actual working android device. The application has been tested in a variety of real devices running various software versions. There has been no issues to date with compatibility.

3.6.2 Customer Testing

I had planned to carry out testing with a number of individuals, the tests will involve "think out loud testing". This is where users will speak aloud while carrying out tasks. Users will be asked to speak out loud their thoughts or feelings while all will be documented and will help with the aspects users my find difficult while using the application. I have also carried out a questionnaire on the application after the think out loud session has been completed. All results will be documented and analyzed.

3.6.3 Results of Customer Testing

Task	Outcomes
Create Account	All participants had no major issue or hesitation in completing this task. The task was completed quickly and efficiently**.**
Login	Similar results as Create Account Users had no issues and completed quickly and efficiently.
Create A Profile "Employee"	Employee's where quickly and easdily able to create a profile.
Create A Profile "Employer"	Employers quickly and easdily able to create a profile quickly and efficiently.
Edit Profile "Employer"	Completing this task was no issue for the users.
Post Job "Employer"	Employer users had a few problems posting jobs, the layout was a bit difficult with the number of fields to fill out.

Edit Existing Post “Employer”	Employer users had no issues editing a job posted after once creating one the first time.
Edit Profile “Employee”	Completing this task was no issue for the users.
View Jobs Posted “Employee”	Employee users had no problems finding the jobs that where posted.
Upload Photo	Users had no issue using this function.
Contact Employer	Users had a bit of difficulty with this function as it is not obviiously found.

3.6.4 Effectiveness

The tests concluded, that users did not have many issues with the navigation and finding the areas of the app that where requested of them. Users once asked to complete a task did so almost immediately. Once users had identified the requested task they carried out the task in a manner that was expected of them. The only issue within the testing was the messaging function. Users found it difficult to contact the employer of a posted job. The error was the users were navigating to the message icon. The correct path is to contact the employer directly from the post where it says “VIEW”. All in all I believe this test was a success.

3.6.5 Efficiency

I had set an unlimited time to complete the required tasks for the test subjects; the average time of completion was 23 minutes. In regards to their interaction with the application the results showed good efficiency with no major issues as in the end all tasks where completed by users. The most time consuming task was the contact an employer about a job post.

3.6.6 Evaluation

As stated before my application will be tested and evaluated within the Android studio, where firebase will offer the analysis and reports regarding crashes or system failures. I also plan to integrate customer feedback to help with bugs that may occur on certain functions on particular devices that where not caught before release. These bugs will be addressed in future updates.

4 Conclusions

I believe the creation of this application has given a new aspect to the way recruitment is done. It is of my opinion, that the current system is a dated process that is time consuming and tedious. I plan to modernize the system, like Uber did for taxis or Airbnb did for travel and accommodation. This app, if created and marketed correctly can do just that. I don’t believe there is anything like it in the current market. Of course there are similar systems but none like this I feel it is a game changer, again like Tinder there was Match.com and other well established systems out there but Tinder seen these systems were out dated also and moved to change and build on the pre existing system to create something new. I may not be the greatest or technically minded coder but I have started to create something to build upon. Rome wasn’t built in a day, most apps today are never finished no application is ever reached a finish line. They are always building and creating new features and functions. It is the same in this application for myself. I have left out a lot of features I wished I had of had the time and expertise to follow through with such as creating system administrator’s, a job matching system that brought the recommended jobs to the employees interests or field of expertise. I did start the foundations of this by creating tags for job skills but as stated I could not find the time or did not have the skills required to complete this functionality. Other features such as a system that uses GPS to find jobs in the local area for users would be something I would like to see implemented in the future. The potential of this application is exponential and go anywhere in the right hands, but at this moment in time those hands are not mine. I have brought this app to the point of which my skills as a programmer will allow.

5 Further development or research

Some of the key features I was not able to implement such as the job matching system where the job tags would connect to the tags the Employee user adds to their own profile. Thus only pulling in jobs that they are only interested in viewing but due to a lack of coding expertise I was unable to implement this core feature. Over time the system could evolve to have further features. I would like to develop features such as a scheduler and mapping system to create events for interviews by matched Employers and Employees. I would also expect to export the application to IOS in the future. There is also potential development to create a desktop application version of the application. On top of features changes and evolution I would have to look at the compatibility of past Android software versions and devices. In this case updates in the future will not be made available for these outdated devices and versions of Android. Future updates will be distributed with improving current security and patching bugs found in the current version. For the growth of the application to maintain reliability I would need to have a system in place to be able to handle an increase in traffic. Such as increasing the database and the server provider as using Firebase for now for free is only feasible as there is little or no traffic once traffic grows Google will expect the account to pay for handling traffic. I would also need to introduce an evaluation system in which users can give feedback and report and bugs or necessary improvements they feel the application would benefit from.

6 References

7 Appendix

7.1 Project Proposal

Objectives

My primary objective is to create a fully functional android application that will create a simplified environment for both job searching and recruitment. The application will push jobs according to relative interests of the user's mobile device. The mobile application will be lightweight front end with an appealing simple interface.

For the first phase of functionality is to push the relevant job postings to the user. The user will have the app where they will swipe left to remove the role or swipe right to show an interest. All roles that are swiped right are going to be stored in the user's own basket.   There will be a desktop version that will not be as simple and have a bit more complexity to it. Users will receive job positions available in their location and desired field. User will swipe left to remove or right to show interest in job. The preferred jobs that were stored in a basket and can be applied for later when there is more time available to them.

On the recruiter side of the application they will receive the report of the level of interest and preview of potential candidate details. Recruiters will only have access to a preview of candidate’s profiles. They must make a request to the candidate to view the full bio. Again this will be in a swipe left or swipe right format. Recruiters will also be allocated a number of promotions for the day. A promotion will be the recruiter’s way of showing an interest in a potential candidate. To try encourage them to apply for an advertised role.

I want the application to be secure for both users and recruiters. There will be very sensitive data contained within profiles. I will do this with secure logins and try use and secure the database.

Another objective will be to use API’s from current recruitment sites to pull in job posting information. I will than need to manipulate this information into more manageable condensed pieces

I would also like to try and add the functionality of an interview scheduling system within the application if possible. This would be a secondary objective to the main app if not completed it can be incorporated into future updates, the scheduling system will allow recruiters and role searchers to arrange an interview or meet and greet. They will have a built in calendar that will keep track of appointments. This will notify them when an upcoming event is on the calendar.

Background

I have been working in a retail role for the past ten years and have been wanting to make a switch to the IT sector in some capacity. The only way to do this is get on the recruitment and job posting sites, correct? Well in a perfect world that is all it would take. I don't feel I’m an unemployable person. I have worked at a management role at which I have excelled at to some degree loyally for 10 years. Personally I found the way of which recruitment is done today to be time consuming, shady and demoralizing.

My idea for the application “ Rolette” arose from the amount of time I personally would spend looking through job websites and apps. I found that I would continuously have to keep entering my search criteria over and over again. By them time I had finally completed the search criteria parameters, it was a case of sieving through the seemingly endless amount of postings that were either not relevant to the roles I was looking for or I was far too under qualified to apply for. Personally I found that when I read a posting, it would be quite demoralizing to stumbling upon roles that I was too under qualified to apply for. Sometimes I would just stop looking with a feeling that my skill set was not meaningful in the work force.

Upon further investigation roles I have been applying for seem to have been overinflated to some extent. With qualifications and experience required in some relatively menial roles within the IT sector. I feel there is a big culture of nepotism in Irish recruitment process, not what you know but who you know. My hope is for this application to speed up the process of recruitment and make it a bit fairer and less disheartening.

My mobile application will have the parenthesis of job applications being regarded as throw away applications. User’s will have less invested interest in the job they applied for as they technically did not invest the same time as they would with standard applications of other jobs sites. The app does have similarities to “Tinder” in the way it shields you from the rejections that you may not be privy to.

The ease of use that will be user friendly to all warps of life. The target market would be any person from ages of 18 - 65 and no specific gender. The App will be free to download from the Play Store.

Technical Approach

Research of this App began with evaluating the top apps of the same category within the Play Store. I found some of the more popular apps such as LinkedIn Jobs, Jobs.ie, Indeed.ie, but none of these will be the same function of the app I plan to develop. The apps stated are job **search** apps, which require the user to do the manual searching. They do have the facility to save users preferences on job searches and the user's resume/CV/cover letter. Rolette selling point is the convenience of having this information pushed to the user in a simple graphical UI. One app called Career Builder did have one of the features I was planning to implement, that once a user likes a role. It is then saved against the profile to be applied for later but once more the app requires a search by the user manually.

First and main requirement for a user is to have an Android device running version 4.4

Next is to have a very simple front end UI with very little complexity for the user to interact with. Very simple compact information being pushed to their device. I want users who have very little technology expertise to feel comfortable with the interaction with the application.
The most technical part of the process for the user should be the setting up of their profile.

To create their account, the user will be required to have a Google account or another Email account. Person should have to fill in their personal details and upload a CV and Bio about themselves. The CV and Cover Letter will be able to be uploaded from their personal computer or Google Drive.

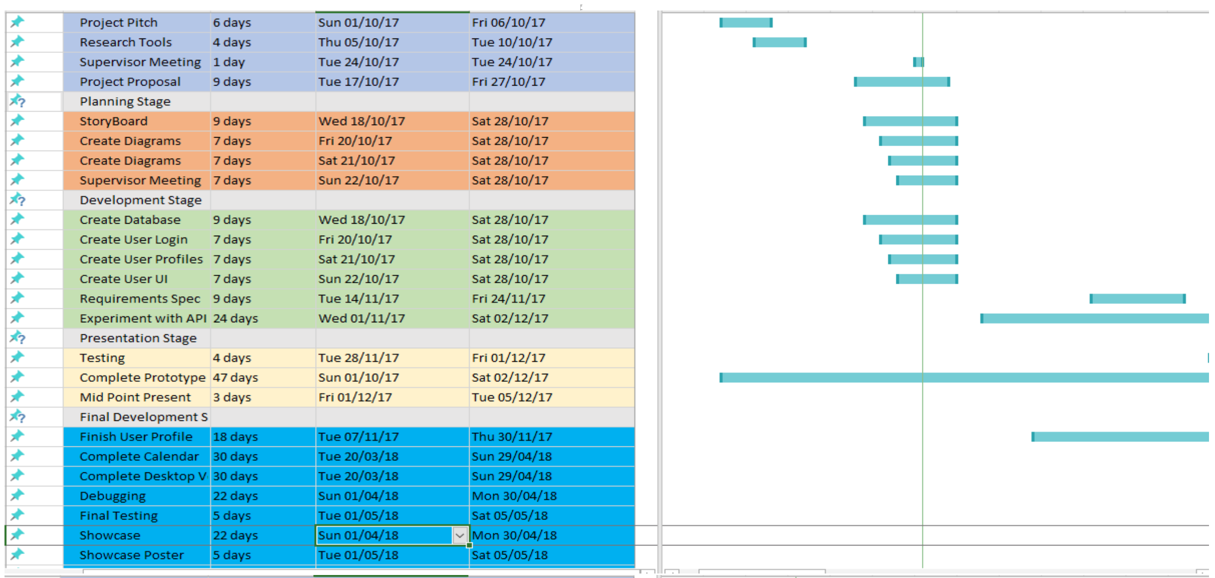
For a recruiter they will follow the same steps as a principal user and will require either a Google account or another Email address. Instead of a CV and Bio recruiters will fill in company info and jobs they wish to advertise by categories.

Application will be programmed within the Android Studio and then hosted with the database within Firebase.

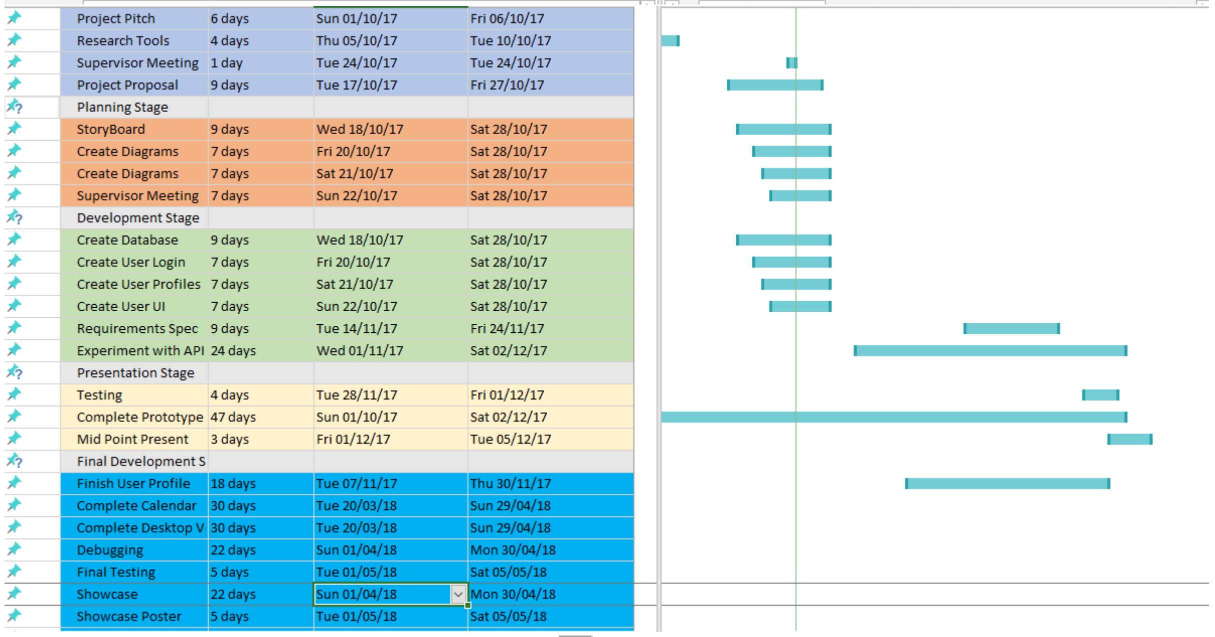
Special resources required

I was originally going to use an IDE called Xamarin studio that allowed you to create the application to be compatible both in Play Store and IOS App Store. When downloading these tools, I found my version of OS was not compatible with the latest version of Xcode. I also realised I would have to program the application in C# which I would not totally be familiar with. I then decided to turn my attention totally towards Android, using Android Studio and MySQL database. I had to increase the RAM of my personal computer from 4GB to 16GB and installed a new SSD with 1TB of memory. As I felt that the running of Android Studio and MySQL was making my machine very unresponsive. I then decided to change to primarily use Firebase to host the application and the database. This has increased the stability on my computer and the responsiveness.

7.2 Project Plan



Project Pitch	6 days	Sun 01/10/17	Fri 06/10/17
Research Tools	4 days	Thu 05/10/17	Tue 10/10/17
Supervisor Meeting	1 day	Tue 24/10/17	Tue 24/10/17
Project Proposal	9 days	Tue 17/10/17	Fri 27/10/17
Planning Stage			
StoryBoard	9 days	Wed 18/10/17	Sat 28/10/17
Create Diagrams	7 days	Fri 20/10/17	Sat 28/10/17
Create Diagrams	7 days	Sat 21/10/17	Sat 28/10/17
Supervisor Meeting	7 days	Sun 22/10/17	Sat 28/10/17
Development Stage			
Create Database	9 days	Wed 18/10/17	Sat 28/10/17
Create User Login	7 days	Fri 20/10/17	Sat 28/10/17
Create User Profiles	7 days	Sat 21/10/17	Sat 28/10/17
Create User UI	7 days	Sun 22/10/17	Sat 28/10/17
Requirements Spec	9 days	Tue 14/11/17	Fri 24/11/17
Experiment with API	24 days	Wed 01/11/17	Sat 02/12/17
Presentation Stage			
Testing	4 days	Tue 28/11/17	Fri 01/12/17
Complete Prototype	47 days	Sun 01/10/17	Sat 02/12/17
Mid Point Present	3 days	Fri 01/12/17	Tue 05/12/17
Final Development S			
Finish User Profile	18 days	Tue 07/11/17	Thu 30/11/17
Complete Calendar	30 days	Tue 20/03/18	Sun 29/04/18
Complete Desktop V	30 days	Tue 20/03/18	Sun 29/04/18
Debugging	22 days	Sun 01/04/18	Mon 30/04/18
Final Testing	5 days	Tue 01/05/18	Sat 05/05/18
Showcase	22 days	Sun 01/04/18	Mon 30/04/18
Showcase Poster	5 days	Tue 01/05/18	Sat 05/05/18



Project Pitch	6 days	Sun 01/10/17	Fri 06/10/17
Research Tools	4 days	Thu 05/10/17	Tue 10/10/17
Supervisor Meeting	1 day	Tue 24/10/17	Tue 24/10/17
Project Proposal	9 days	Tue 17/10/17	Fri 27/10/17
Planning Stage			
StoryBoard	9 days	Wed 18/10/17	Sat 28/10/17
Create Diagrams	7 days	Fri 20/10/17	Sat 28/10/17
Create Diagrams	7 days	Sat 21/10/17	Sat 28/10/17
Supervisor Meeting	7 days	Sun 22/10/17	Sat 28/10/17
Development Stage			
Create Database	9 days	Wed 18/10/17	Sat 28/10/17
Create User Login	7 days	Fri 20/10/17	Sat 28/10/17
Create User Profiles	7 days	Sat 21/10/17	Sat 28/10/17
Create User UI	7 days	Sun 22/10/17	Sat 28/10/17
Requirements Spec	9 days	Tue 14/11/17	Fri 24/11/17
Experiment with API	24 days	Wed 01/11/17	Sat 02/12/17
Presentation Stage			
Testing	4 days	Tue 28/11/17	Fri 01/12/17
Complete Prototype	47 days	Sun 01/10/17	Sat 02/12/17
Mid Point Present	3 days	Fri 01/12/17	Tue 05/12/17
Final Development S			
Finish User Profile	18 days	Tue 07/11/17	Thu 30/11/17
Complete Calendar	30 days	Tue 20/03/18	Sun 29/04/18
Complete Desktop V	30 days	Tue 20/03/18	Sun 29/04/18
Debugging	22 days	Sun 01/04/18	Mon 30/04/18
Final Testing	5 days	Tue 01/05/18	Sat 05/05/18
Showcase	22 days	Sun 01/04/18	Mon 30/04/18
Showcase Poster	5 days	Tue 01/05/18	Sat 05/05/18

Project1.mpp

7.3 Monthly Journals

7.3.1 September

My Achievements

This month, I was tasked with the creating my project that will be my final year software project. I had a few I wanted to do but the idea was to decide on one and go before three of my lectures and pitch my idea. They would subsequently decide whether if the idea was feasible or not challenging enough. I wanted to develop an app that would be beneficial to a broad group and relevant to the broad group.

My Reflection

I finally decided to go with an android app as I had wanted to do a cross platform application by using an IDE called Xamarin but my computer version does not support the latest version of Xcode and had to settle for android only.

Next Steps

I will try gather more requirements as how to develop my application. Next month I will need to create my project proposal this will include a Gantt chart of my activities that I plan to implement.

7.3.2 October

My Achievements

My proposal was approved by the three lecturers they name of the application is called Rolette. It will be a recruitment-based application that will involve jobs being posted to the user rather then the old process of searching for the jobs. I have got an idea of the functions of which I want the application to include. Now I will just need to research on how to implement them.

My Reflection

I was pleased my application was approved by the panel with revisions I just hope I have not bitten off more than I can chew by over promising and under delivering. My knowledge of the android studio IDE will need to be refreshed.

Next Steps

I have a lot of other college work that needs attention this month I will need to allocate time each week to research my ideas. will need to look over the documents that need to be delivered next month.

7.3.3 November

My Achievements

This month, I was tasked with the submitting the requirement spec report and the technical report before my midpoint presentation. I have received my date for my presentation and it will be on the 5th of December at 5.30pm with my supervisor Padraig involved. I must still complete the presentation template before the deadline

My Reflection

I have no actual working code to present at the presentation. I had been advised by one of the lecturers that the code is not essential at this point but if I can create some wireframes to show my design and the features I wish to implement then I should be ok.

Next Steps

I really need to get started with coding the application but with our current workload and being in full time employment I am finding it very tough to allocate any time to the code most of my time that I am using with this software project in mind is to complete the documents.

7.3.4 December

My Achievements

This month, I had my Christmas exams this month so I had to spend a lot of time studying and finishing CA’s for this semester. I feel I wasn’t be able to dedicate much time to this project over this period also with it being Christmas and the time we spent this semester with an extra subject as to any other semester it was hard to get around to working on this application.

My Reflection

I have still not got around to actually starting this project and the pressure was immense over Christmas as I work in retail there is no time off and this is the busiest time of the year on top of that I had the added pressures of studying and finishing CA's.

Next Steps

Next month I plan to get back to college and start my project properly. I need to get the Firebase and the application in android studio connected together. I will watch some tutorials online of how to implement this.

7.3.5 January

My Achievements

This month, I got back to college after the Christmas break ready to start working on this project. I did complete the goals of last month and I did get the application up and running with the Android studio IDE and Firebase. I found Firebase a very useful tool to use and can see why it was so highly recommended to use with Android applications.

My Reflection

I believe last month was a successful month as I have finally gotten the time to start working on my application. Although just a very simple splash screen I feel I have achieved something this month and feel I can carry this forward to the next month and build upon it to add the next layer of functionality. I'm behind on my expected time scale.

Next Steps

Next month I plan to get the Login system working for the application and this will involve creating two separate types of logins for users an employee user and an employer user. All my previous coding projects we have only ever-coded a login for one user. This is not a task I have ever had to do before and will require a good deal of research to complete. I will have to watch more online tutorials on the android development system and try to apply them to my own project if possible.

7.3.6 Febraury

My Achievements

This month, I had tasked my self with the job of creating the logins for both types of users, employee and employer. I had watched a lot of online tutorials on the setting up of a login and registration system on an android application. I found a good tutorial online that explained how to redirect to different activities, creating a different user type, when user login is successful, you can retrieve the type of user and redirect to specific activity.

My Reflection

I believe last month to be very successful as achieved my goal from the previous month. I would only hope that it is going to last as the start of CA's distributed has begun, the class has been tasked with an individual java software project which I feel is unfair to give to students at this point of the year. The only software development project we should be focusing on is this project and the project account for 50% of our overall marks for this year. If it were a group project it would mean a little less pressure on us for this final semester.

Next Steps

I want to implement the job functionality where employers can post jobs to the application via Firebase storing the information and then see if I can get this displayed to the employee users on the other activity. Again I will have to find relevant tutorials in this area hope to apply them to my own project.

7.3.7 March

My Achievements

This month, I was tasked with the creating of a system to allow the employers profiles to post jobs that can be seen on the employee profiles. I have been still working on this, as this task was not was easy as I first thought it would be. This is really slowing down the progress of the project. I have though, got the backend to accept the posts and are being stored in firebase. The next thing though is to display them to the other type of user.

My Reflection

I have not completed the goals I had set out initially and found the project tasks a bit overwhelming I was not sure when I started if going with mobile was a good idea as I was not able to join the mobile stream of this course. I will still perceiver on with the project.

Next Steps

Create a chat feature for users to interact with each other and finish the job viewing functionality. I really need to get these functions running as the following month we will be starting the exams for the final semester and there are a lot of college CA's due, so I'm not sure if I can find the time to dedicate to the application. I will try but I feel a little pressed for time and the project may not live up to my expectations.

7.3.8 April

My Achievements

This month, I was tasked with the competing the chat feature and polishing off the job viewer on the employee profile side of the application. This was not a good month for the project as I had a lot of course work due this month and then we have exams starting. I have to finish a big project for Secure Programming, which was an individual project and required a lot of attention as it was worth 50% of the course marks.

My Reflection

With the course work due and exams about to begin I neglected the project to the point I was losing sleep over it. Part time study and full time work does not help with the completing of this project.

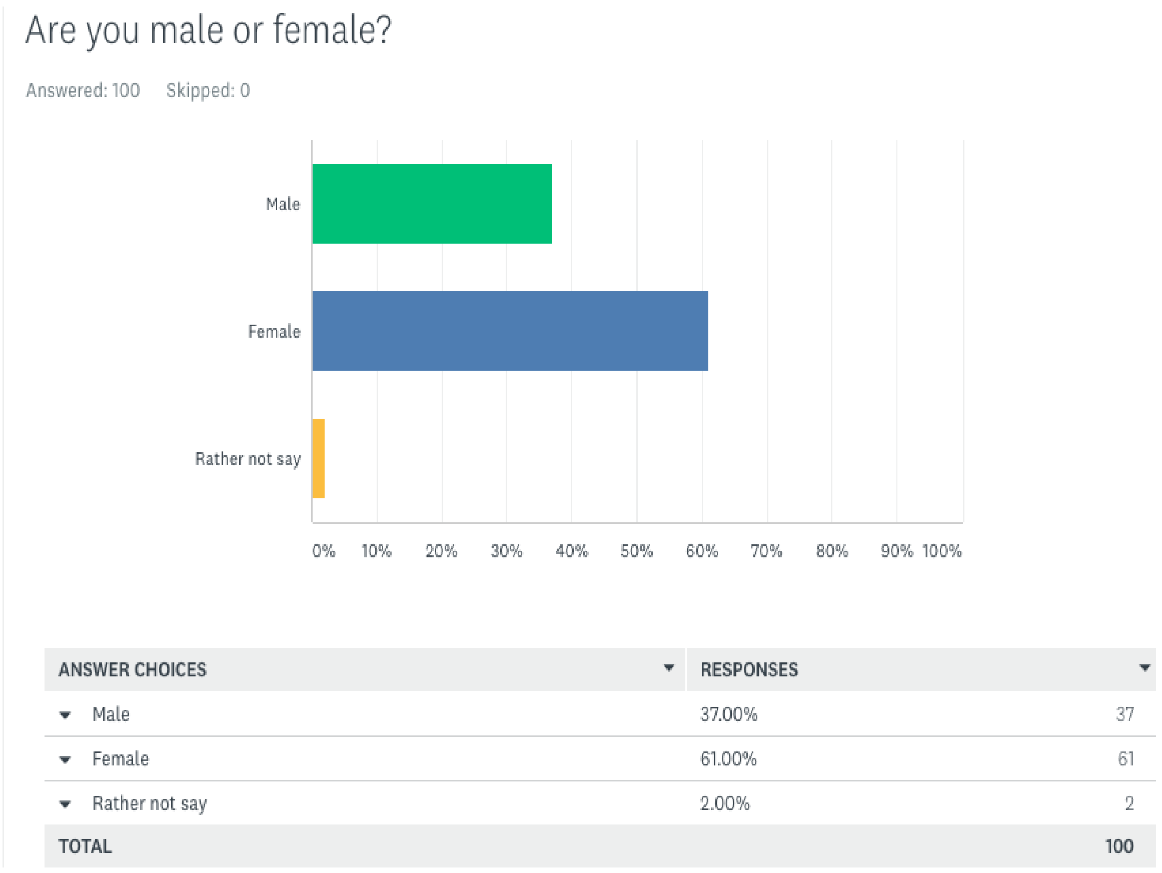
Next Steps

I really need clean up any bugs that I have found such as not being able to view employee's profiles and making stricter conditions on the passwords. I may add some more features to make up for the lack of features initially promised such as a profile picture upload.

7.4 Other Material Used

Market Research Survey

Question 1

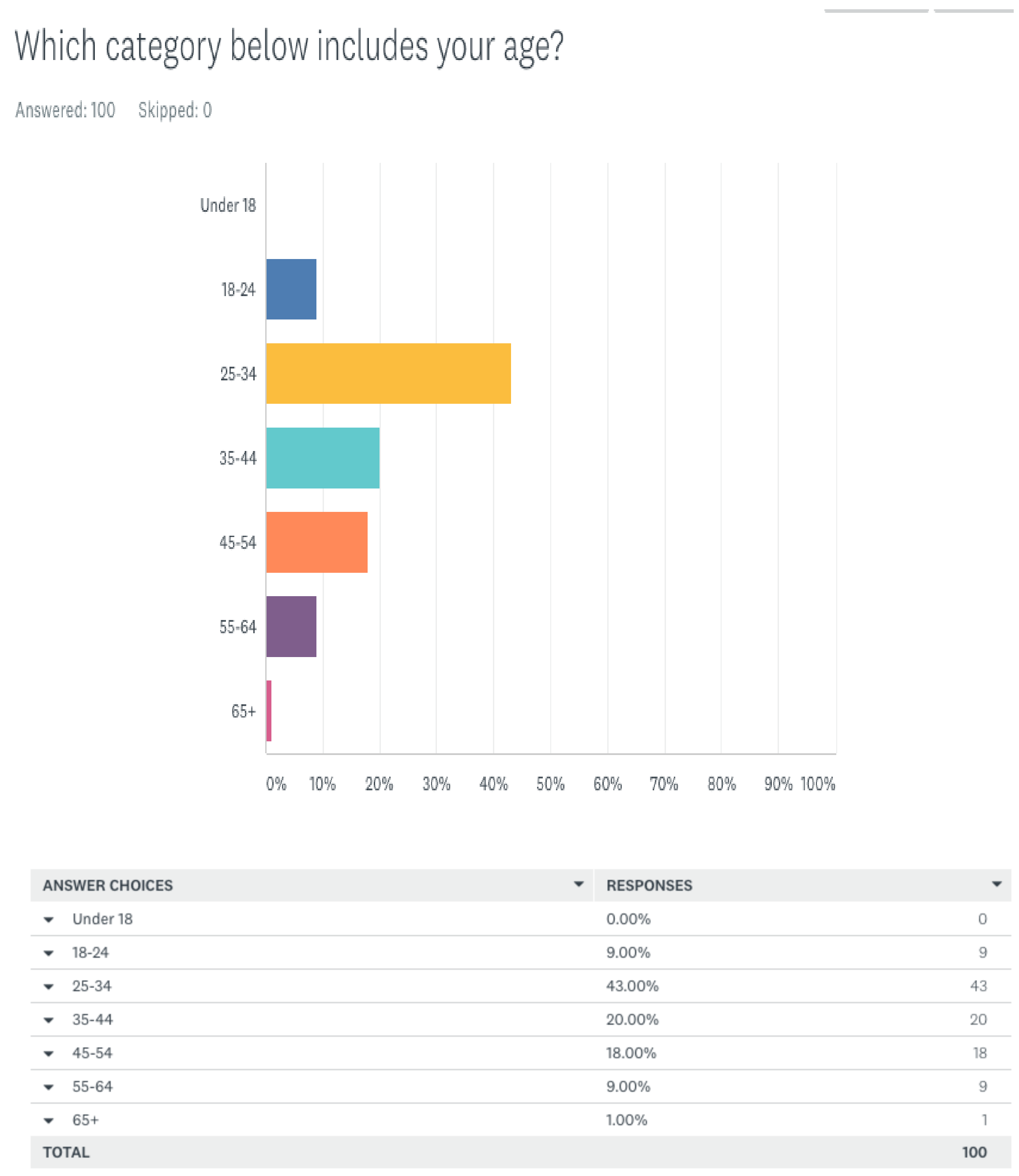


Are you male or female?

Answered: 100 Skipped: 0

ANSWER CHOICES	RESPONSES	
Male	37.00%	37
Female	61.00%	61
Rather not say	2.00%	2
TOTAL		100

Question 2


Which category below includes your age?

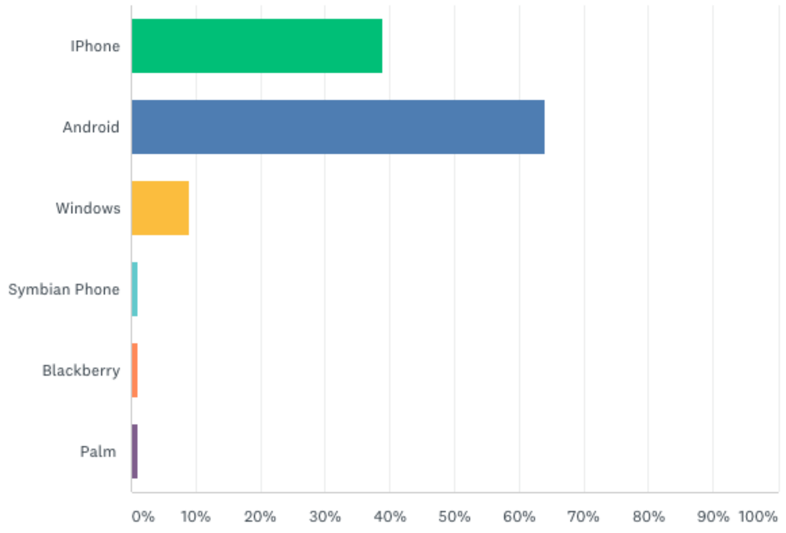
Answered: 100 Skipped: 0

ANSWER CHOICES	RESPONSES	
Under 18	0.00%	0
18-24	9.00%	9
25-34	43.00%	43
35-44	20.00%	20
45-54	18.00%	18
55-64	9.00%	9
65+	1.00%	1
TOTAL		100

Question 3

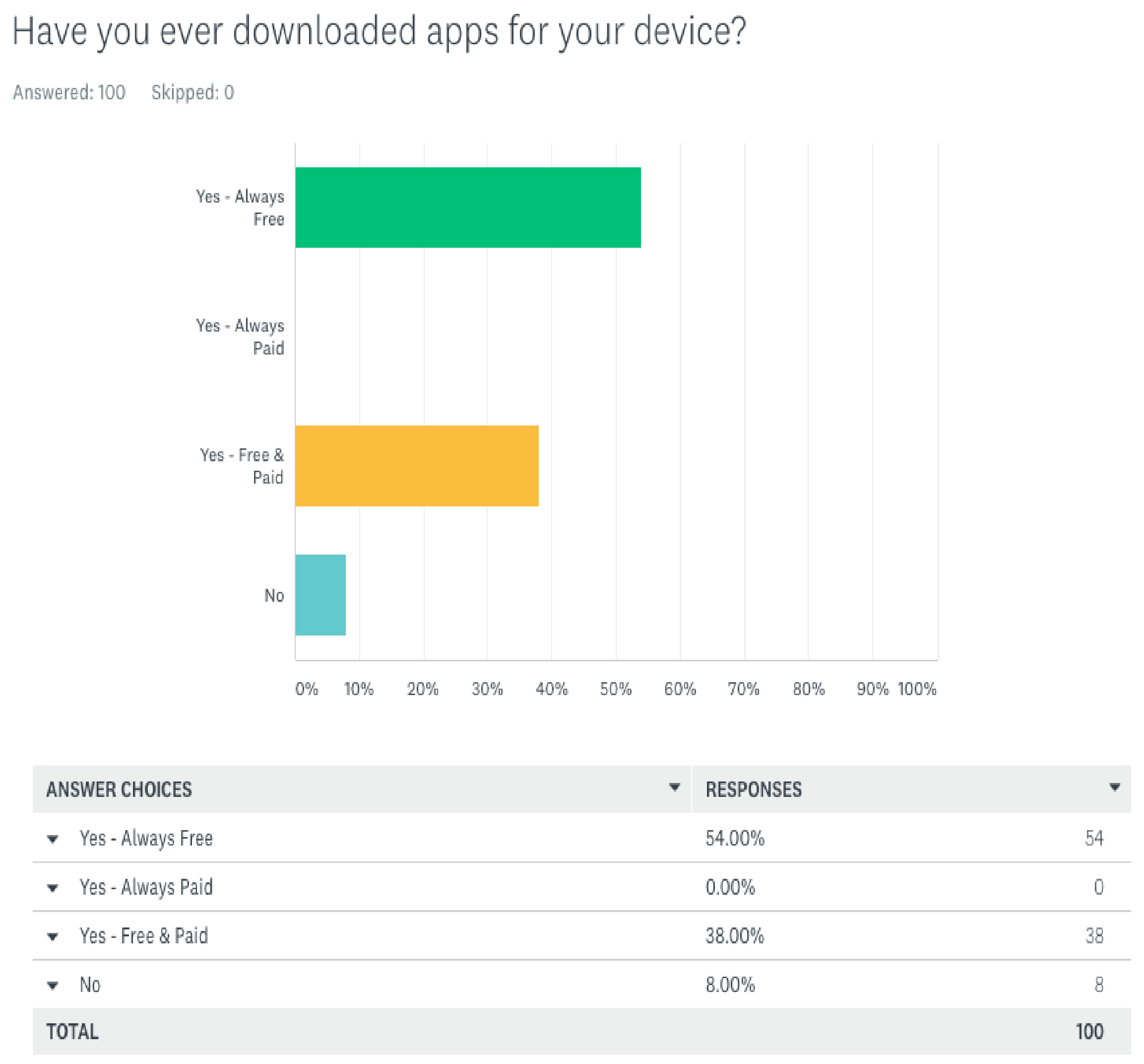
Which of the following mobile devices do you own?

Answered: 100 Skipped: 0



ANSWER CHOICES	RESPONSES	
IPhone	39.00%	39
Android	64.00%	64
Windows	9.00%	9
Symbian Phone	1.00%	1
Blackberry	1.00%	1
Palm	1.00%	1
Total Respondents: 100		

Question 4



Have you ever downloaded apps for your device?

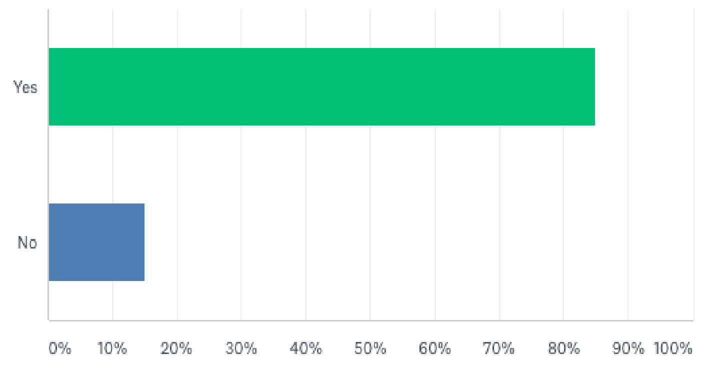
Answered: 100 Skipped: 0

ANSWER CHOICES	RESPONSES	
Yes - Always Free	54.00%	54
Yes - Always Paid	0.00%	0
Yes - Free & Paid	38.00%	38
No	8.00%	8
TOTAL		100

Question 5

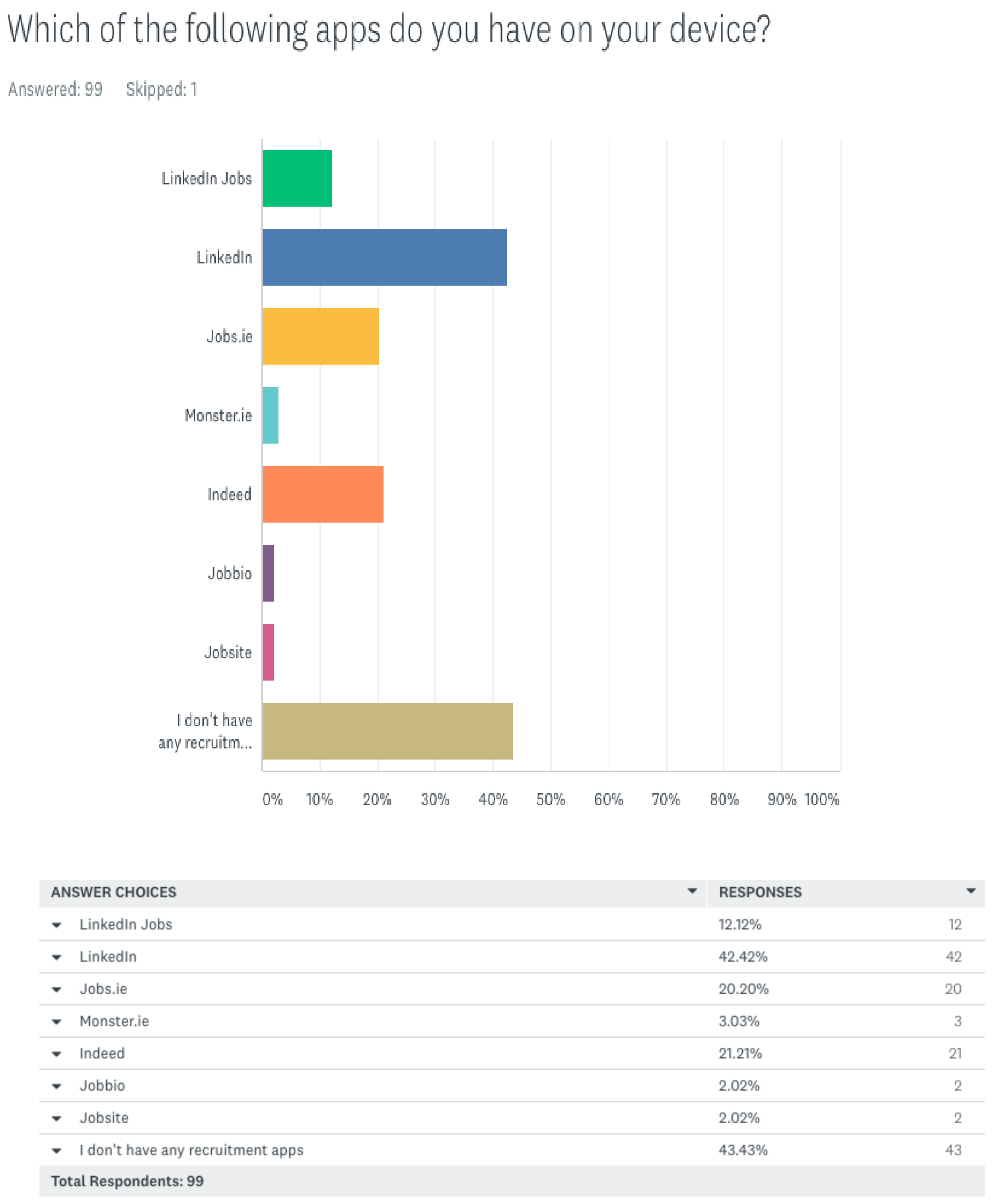
Are you currently employed

Answered: 99 Skipped: 1



ANSWER CHOICES	RESPONSES	
Yes	84.85%	84
No	15.15%	15
TOTAL		99

Question 6



Which of the following apps do you have on your device?

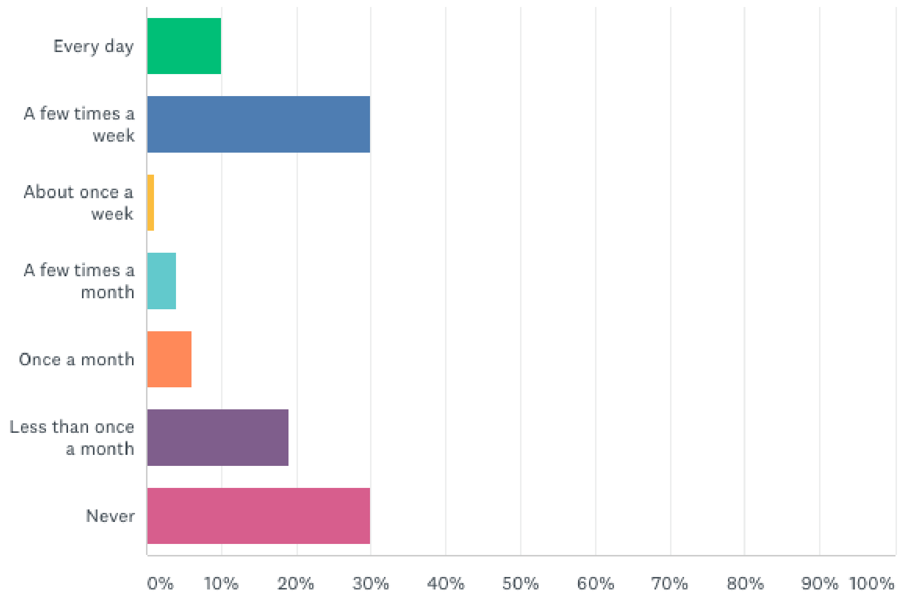
Answered: 99 Skipped: 1

ANSWER CHOICES	RESPONSES	
LinkedIn Jobs	12.12%	12
LinkedIn	42.42%	42
Jobs.ie	20.20%	20
Monster.ie	3.03%	3
Indeed	21.21%	21
Jobbio	2.02%	2
Jobsite	2.02%	2
I don't have any recruitment apps	43.43%	43
Total Respondents: 99		

Question 7

How often would you use one of these apps on your device?

Answered: 100   Skipped: 0

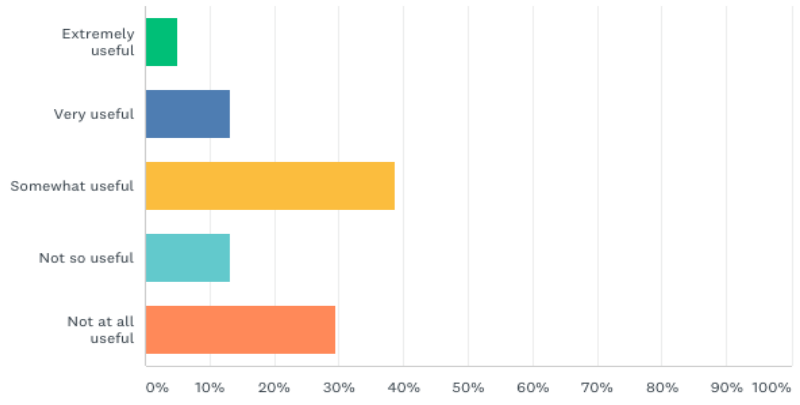


ANSWER CHOICES	RESPONSES	
Every day	10.00%	10
A few times a week	30.00%	30
About once a week	1.00%	1
A few times a month	4.00%	4
Once a month	6.00%	6
Less than once a month	19.00%	19
Never	30.00%	30
TOTAL		100

Question 8

How useful do you find these apps in gaining employment?

Answered: 98 Skipped: 2

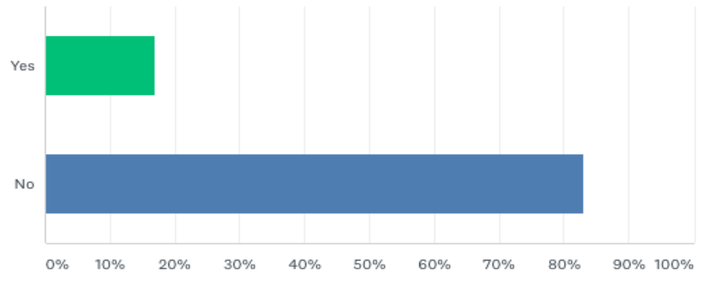


ANSWER CHOICES	RESPONSES	
Extremely useful	5.10%	5
Very useful	13.27%	13
Somewhat useful	38.78%	38
Not so useful	13.27%	13
Not at all useful	29.59%	29
TOTAL		98

Question 9

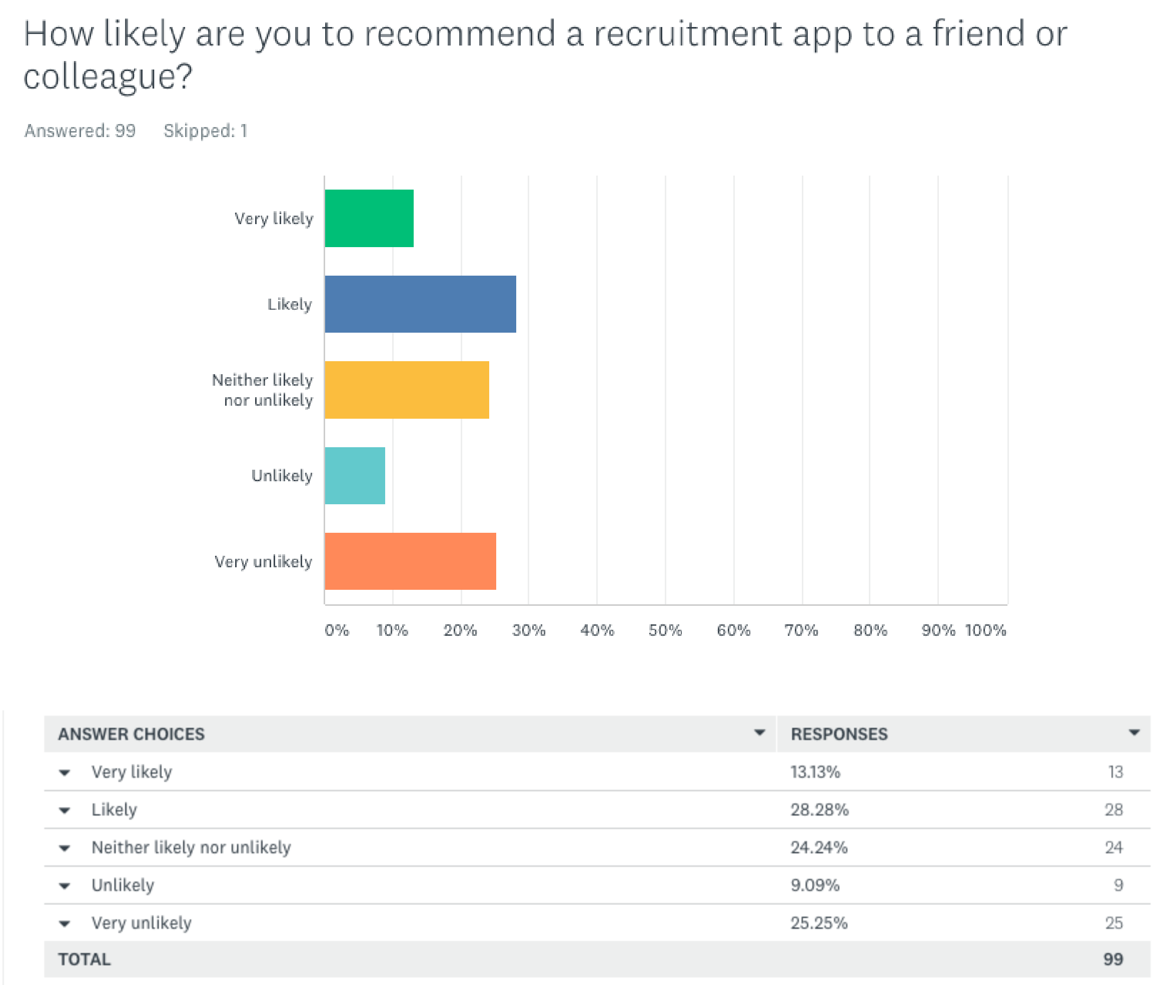
Have you ever found employment from one of these apps?

Answered: 100 Skipped: 0



ANSWER CHOICES	RESPONSES	
Yes	17.00%	17
No	83.00%	83
TOTAL		100

Question 10



How likely are you to recommend a recruitment app to a friend or colleague?

Answered: 99 Skipped: 1

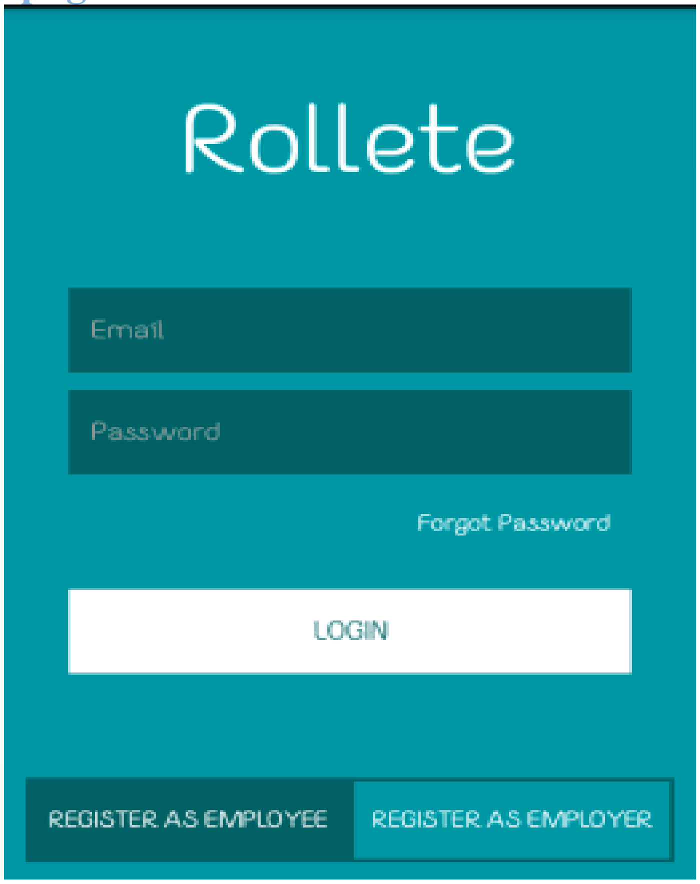
ANSWER CHOICES	RESPONSES	
Very likely	13.13%	13
Likely	28.28%	28
Neither likely nor unlikely	24.24%	24
Unlikely	9.09%	9
Very unlikely	25.25%	25
TOTAL		99

7.5 *User Manual*

Introduction

The purpose of this document is to provide a user manual to help guide the persons using this mobile application, how navigate and access the functions of this application.

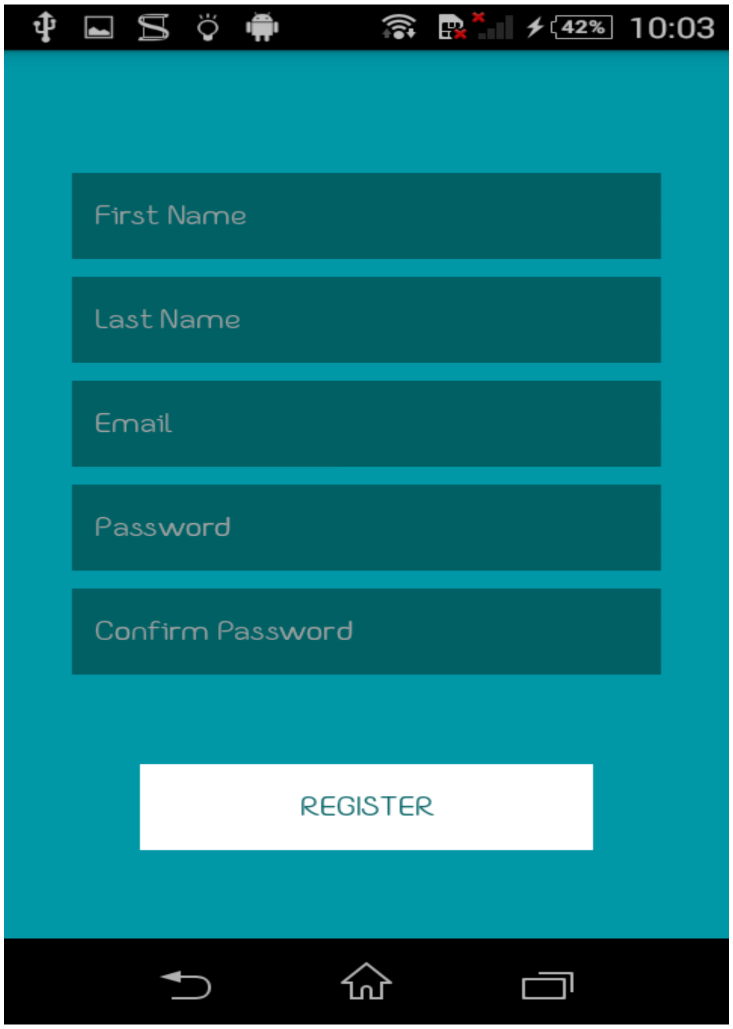
Rolette Homepage



- As we can see in this screenshot shot user can either register as an employee or an employer or just login with the credentials they have already registered with.

- These credentials will be authenticated against those stored in Firebase. If the details match the user will then be brought to either the Employee dashboard or an Employer Dashboard.

Registration Page



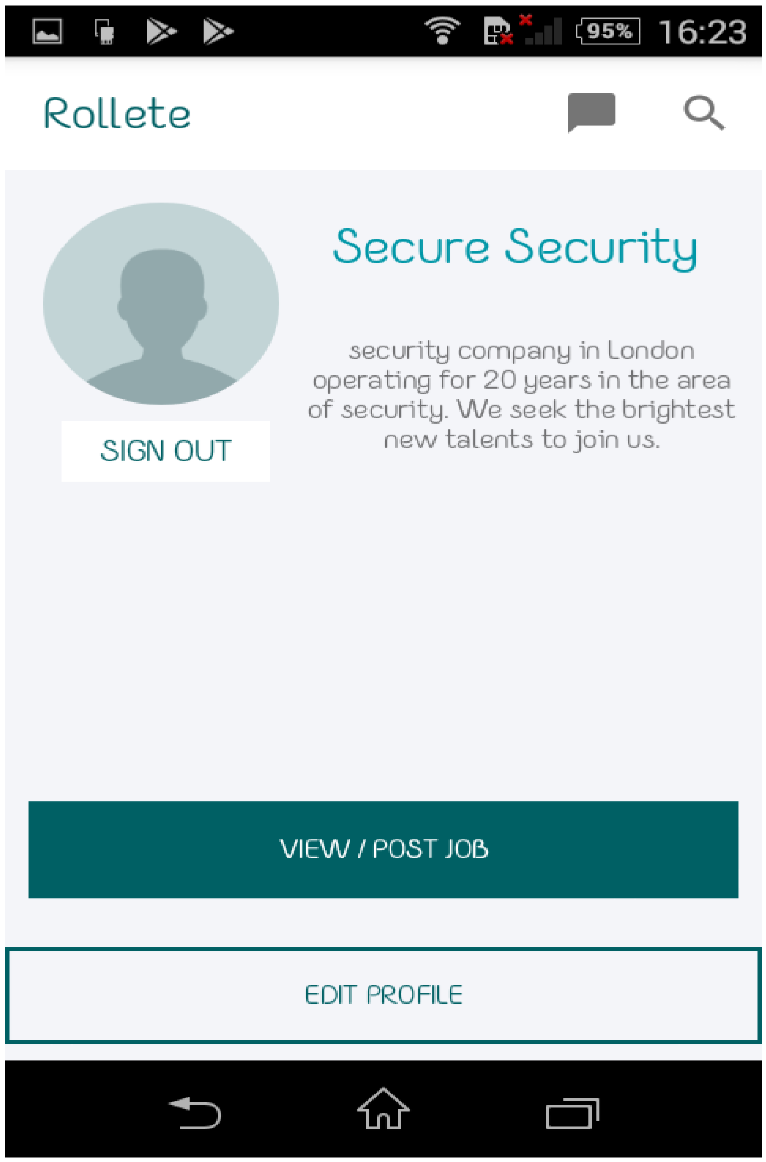
- Here we can see the registration page for either an employer or an employee. So if a user has not used the application before and is required to create a new account they will be brought to this page to complete the required fields.

- User will have to supply a valid email and a password. The Password must contain one uppercase letter and a number in order to register.

- Once complete click on the register tab and then the information is stored in the Firebase database where it is given a unique user ID.

- Once registered the user will then be redirected to the login page.

Employer Landing Page



Secure Security

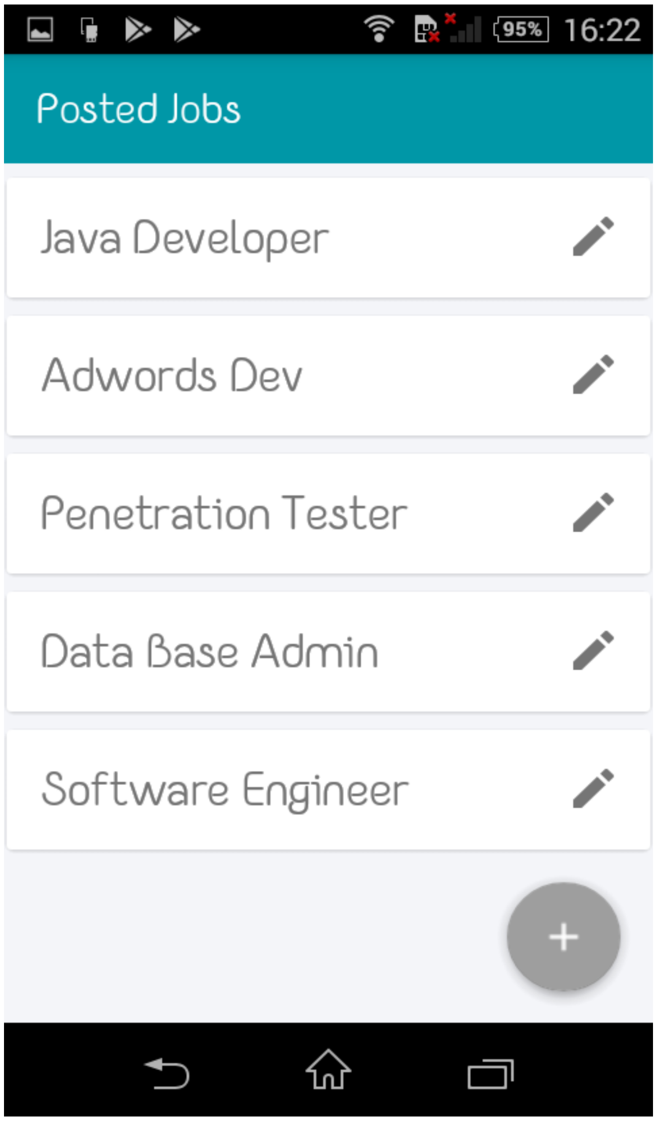
security company in London operating for 20 years in the area of security. We seek the brightest new talents to join us.

- Here once logged in as an employer this will be the landing page for the Employer user or the home page.

- Here the user has the option to either Edit Profile or View/Post Job or View Messages or Sign Out.

- First thing will be to give the company a bit of detail so it is recommended to fill out the profile before proceeding to posting jobs.

View/Post Job (Employer)



Posted Jobs

Java Developer

Adwords Dev

Penetration Tester

Data Base Admin

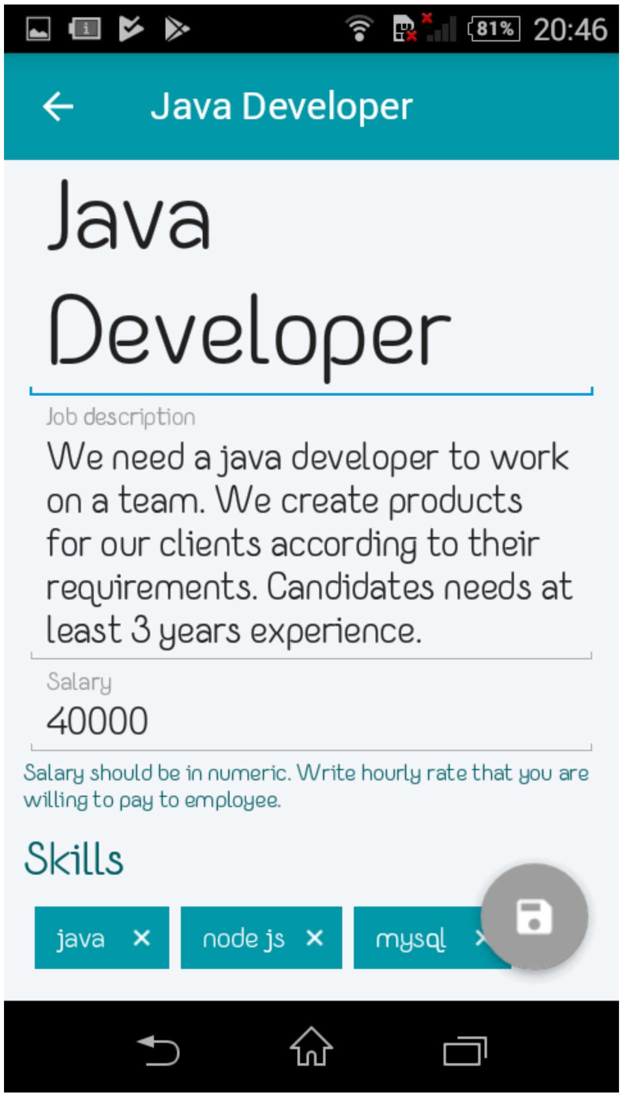
Software Engineer

- In the View/Post Job we can see the jobs that a company has posted already. From here you can edit an existing post or delete an existing post.

- Users may want to edit an existing post if they are not getting much interest in the posted job.

- If a user wants to create a new job post they simply have to tap the + icon on the screen and they will be directed to the next screen.

Post New Job



Java Developer

Job description

We need a java developer to work on a team. We create products for our clients according to their requirements. Candidates needs at least 3 years experience.

Salary

40000

Salary should be in numeric. Write hourly rate that you are willing to pay to employee.

Skills

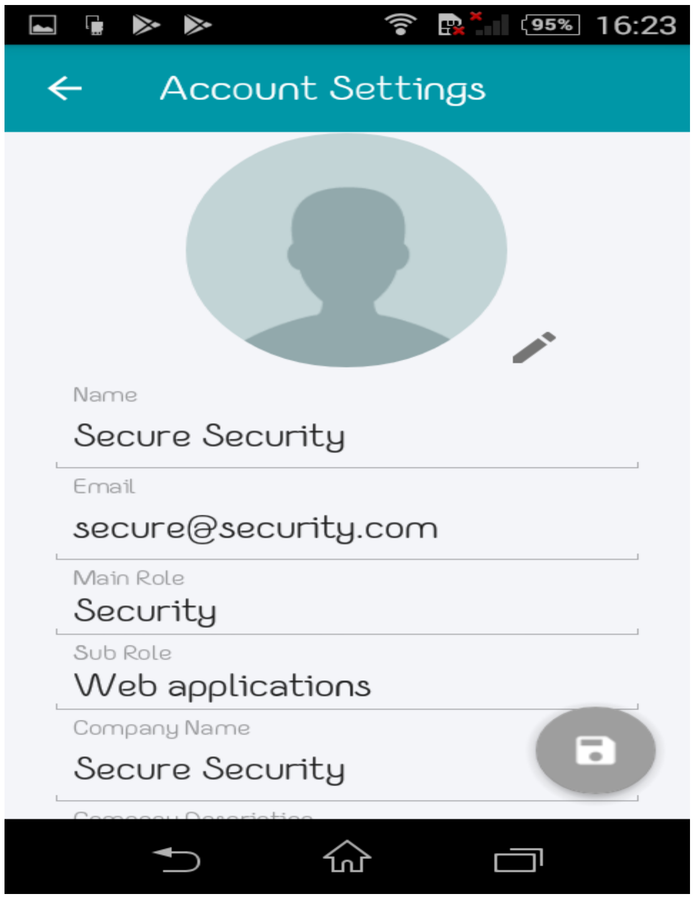
- Here we have the screen when the Employer user wants to post a new job. User selects the most suiting title to the job the wish to post on the application.

- Then the user needs to fill out the the Job Description field.

- The next field to fill out is the Salary then the Skills required to work in the role posted.

- User can then just hit the save icon when they are happy with the job and this will then be posted to the application.

Employer Edit Profile



Name
Secure Security
Email
secure@security.com
Main Role
Security
Sub Role
Web applications
Company Name
Secure Security

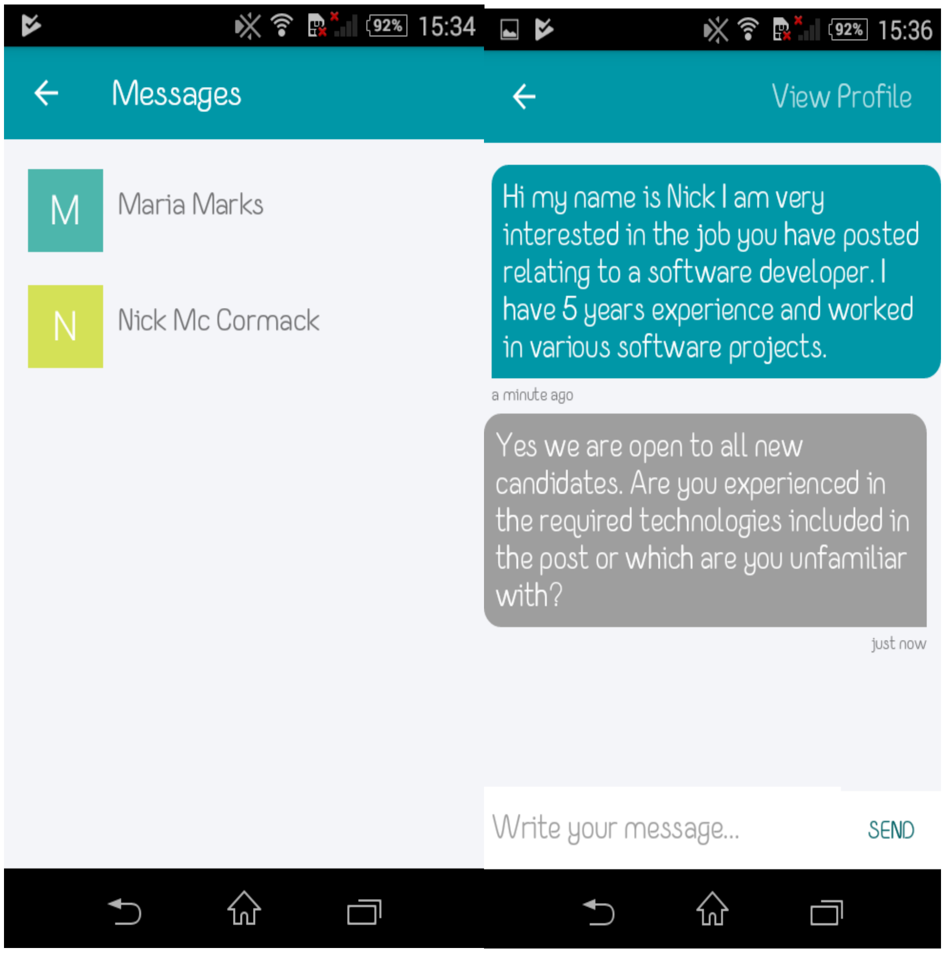
- The Edit Profile option on the Employer Home Screen allows the user when first registered to enter their company details.

- The Employer User can upload a profile image, then enter the other required fields, such as Company name, Email, Area of trade and there is aslso the

Company Bio where the company fills out a description on the companies history or what it stands for.

- Once completed the user then saves the details and these are then stored in the Firebase Database.
- Users that are already registered can use this option to update the details already entered. Again follow the same steps as the registration steps and save.
- These new details will update automatically and reflect instantly on the application.

Messages



Hi my name is Nick I am very interested in the job you have posted relating to a software developer. I have 5 years experience and worked in various software projects.

a minute ago

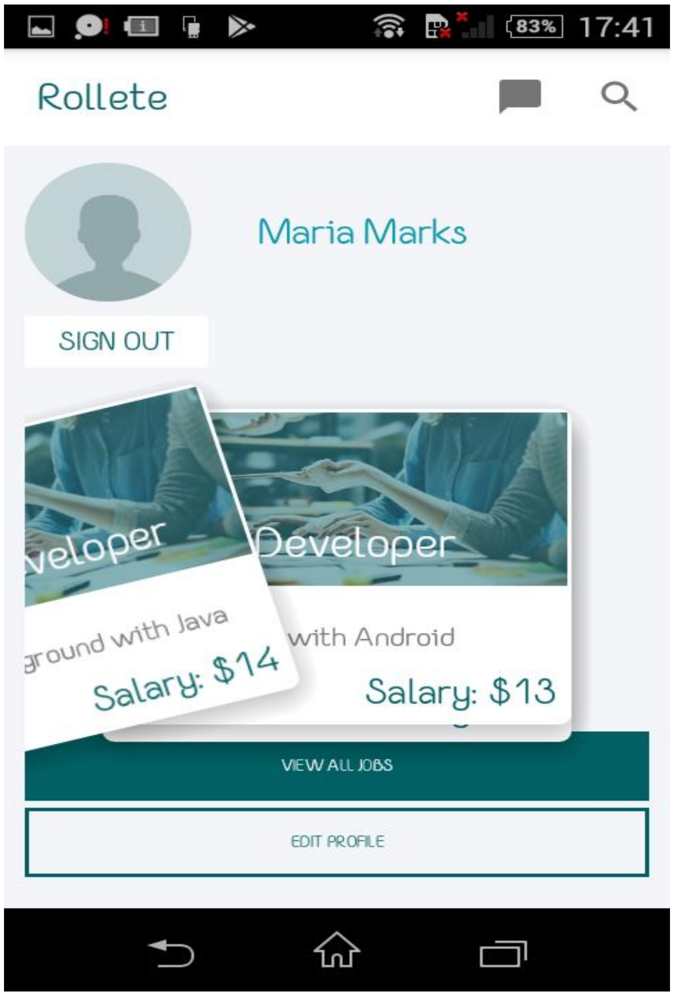
Yes we are open to all new candidates. Are you experienced in the required technologies included in the post or which are you unfamiliar with?

just now

- In this window the user can view the messages they have received an Employer can only receive messages about jobs they have posted.

- Employees can find this option inside of a job posting.

Employee Landing Page



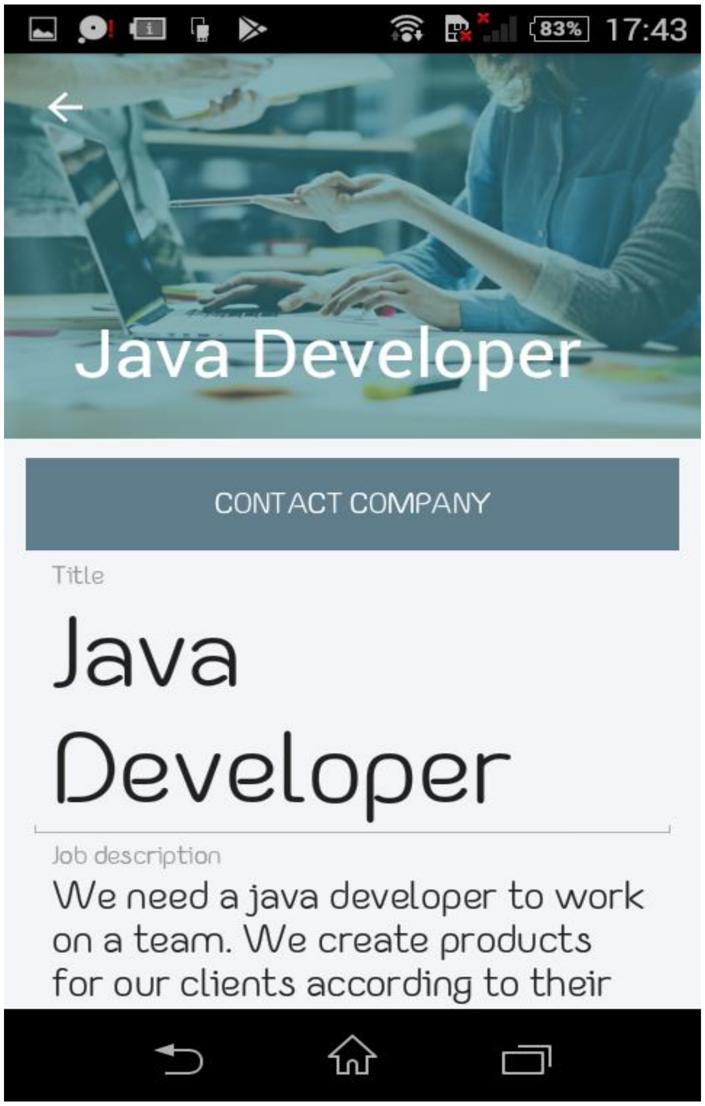
- Once an Employee user has logged in or registered successfully they will land here an be presented with the employee landing page.

- They have a number of options to go to from here, such as View Jobs, Edit Profile, View Messages or Sign Out.

- It is reccomended for a newly registered Employee to edit their details first as to gain interest from potential Employers.

- Next thing to do is to look through the jobs that have been posted, from here a user can select View Job and can read more detail of the job and even contact the employer who posted the job.

View Job Post



Title

Java Developer

Job description

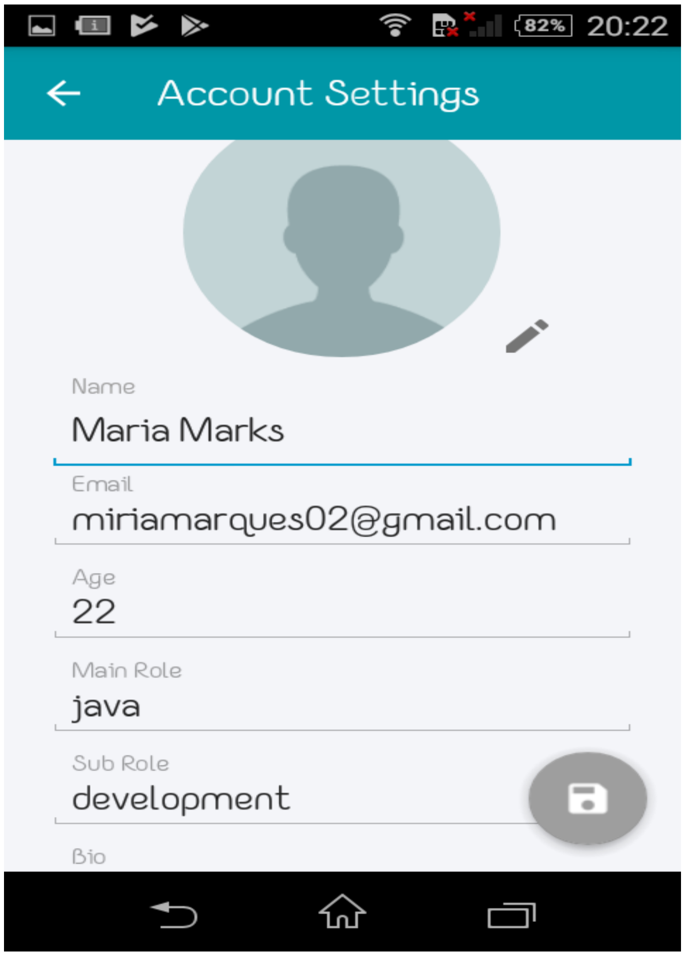
We need a java developer to work on a team. We create products for our clients according to their

- When an employee chooses a role they are interested in they can view it in more detail by tapping on the "View" option rather than swiping it away.

- This will then show the job post in more detail, such as what criteria is required from a candidate.

- When the employee taps on the contact company they can make direct contact with the company that posted the job.

Edit Employee Profile



Name
Maria Marks
Email
miriamarques02@gmail.com
Age
22
Main Role
java
Sub Role
development
Bio

- Here we have the edit profile option for the employee user.

- Employees can either enter their details upon registering or they can update their current details. All of these details are updated in real-time to the Firebase database.

- The user has also got the option to upload a profile picture.

- Fields to be filled out include Name, Age, Main Role, Sub Role and a Bio on themselves that would have a piece on their experience and background.

- The user then just hits the save icon and all the information will reflect on the users profile instantly.