

User Manual

A user's guide to ReverseHand

PREPARED BY

Cache Money

Table of Contents

<i>Purpose Overview</i>	3
System Overview	3
Who can benefit from using ReverseHand?	3
Key features of ReverseHand	3
<i>The System</i>	4
Deployment model	4
Overall system architecture	4
Requirements	4
Hardware Requirements	4
Software Requirements	4
Overview of main features	5
Logging In	5
Login	5
Navigation bar	5
.....	5
Sign-Up	6
Navigating the consumer client	7
.....	8
.....	9
Navigating the tradesman client	11
Error Handling	14



Purpose Overview

System Overview

ReverseHand is a mobile application that was created with the vision of reducing any power imbalances that consumers may face when seeking trade services. To achieve this, the mobile application allows consumers to make their needs for services known in the form of job listings, where tradesmen are able to place bids in order to be chosen and hired.

Overall, the system aims to increase convenience for all concerned parties while reducing the time taken to get services rendered and reduce the cost of those services by providing the consumer with options to choose from, and independent tradesman the opportunities to find work.

Who can benefit from using ReverseHand?

- Users looking for non-union, freelance tradies.
- Users who are non-union, freelance tradies looking for nearby job opportunities.

Key features of ReverseHand

This Project aims to allow the consumer to post an advert detailing the issue that requires fixing, thereafter tradesmen can view the advert and if it is within their domain and they are interested, they may place a bid to take on the job.

The key takeaway is that a tradesman must compete to acquire the service which will drive the price of the service down, and increase the quality of service provided by the tradesman.

Consumer Users

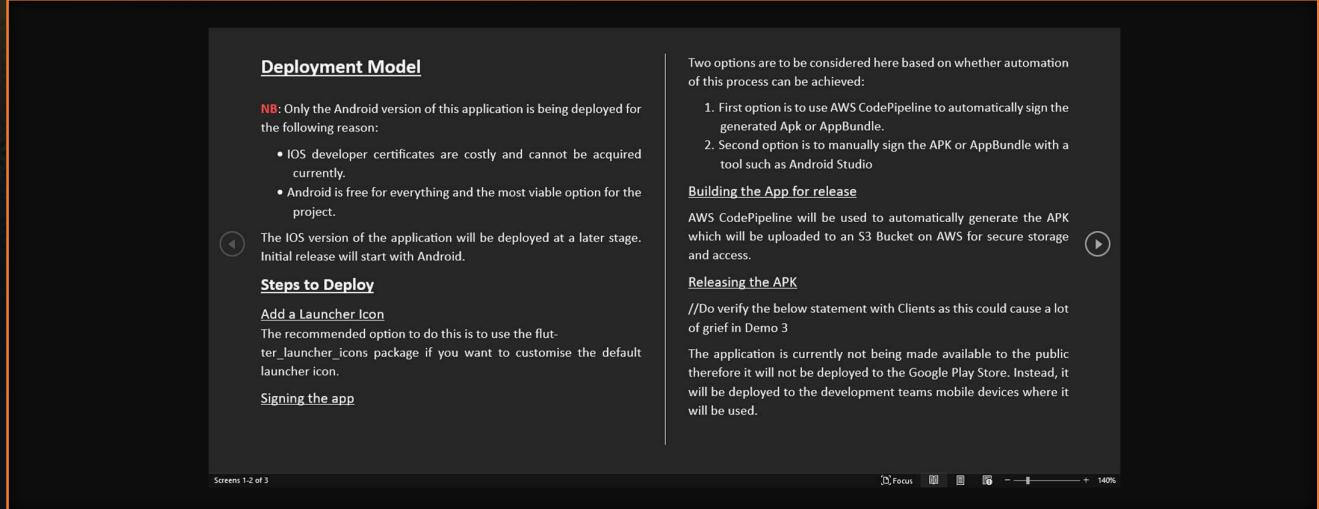
- Creating new job advert listings
- Viewing bids
- Shortlisting and finally accepting a bid

Tradesman Users

- Viewing job advert listings
- Placing a bid on a job

The System

Deployment model



Deployment Model

NB: Only the Android version of this application is being deployed for the following reason:

- IOS developer certificates are costly and cannot be acquired currently.
- Android is free for everything and the most viable option for the project.

The IOS version of the application will be deployed at a later stage. Initial release will start with Android.

Steps to Deploy

Add a Launcher Icon
The recommended option to do this is to use the flutter_launcher_icons package if you want to customise the default launcher icon.

Signing the app

Two options are to be considered here based on whether automation of this process can be achieved:

1. First option is to use AWS CodePipeline to automatically sign the generated Apk or AppBundle.
2. Second option is to manually sign the APK or AppBundle with a tool such as Android Studio

Building the App for release

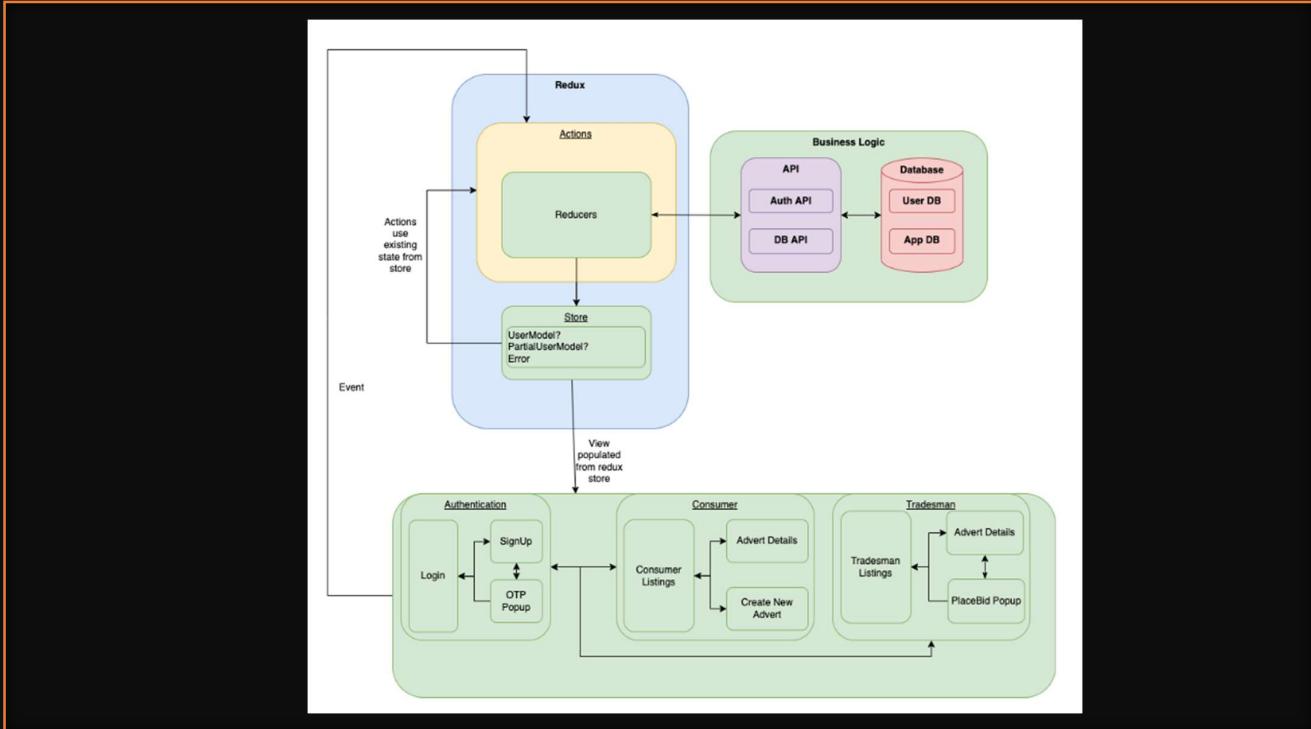
AWS CodePipeline will be used to automatically generate the APK which will be uploaded to an S3 Bucket on AWS for secure storage and access.

Releasing the APK

//Do verify the below statement with Clients as this could cause a lot of grief in Demo 3
The application is currently not being made available to the public therefore it will not be deployed to the Google Play Store. Instead, it will be deployed to the development teams mobile devices where it will be used.

Screens 1-2 of 3

Overall system architecture



Requirements

Hardware Requirements

A mobile device, either IOS or Android.

Software Requirements

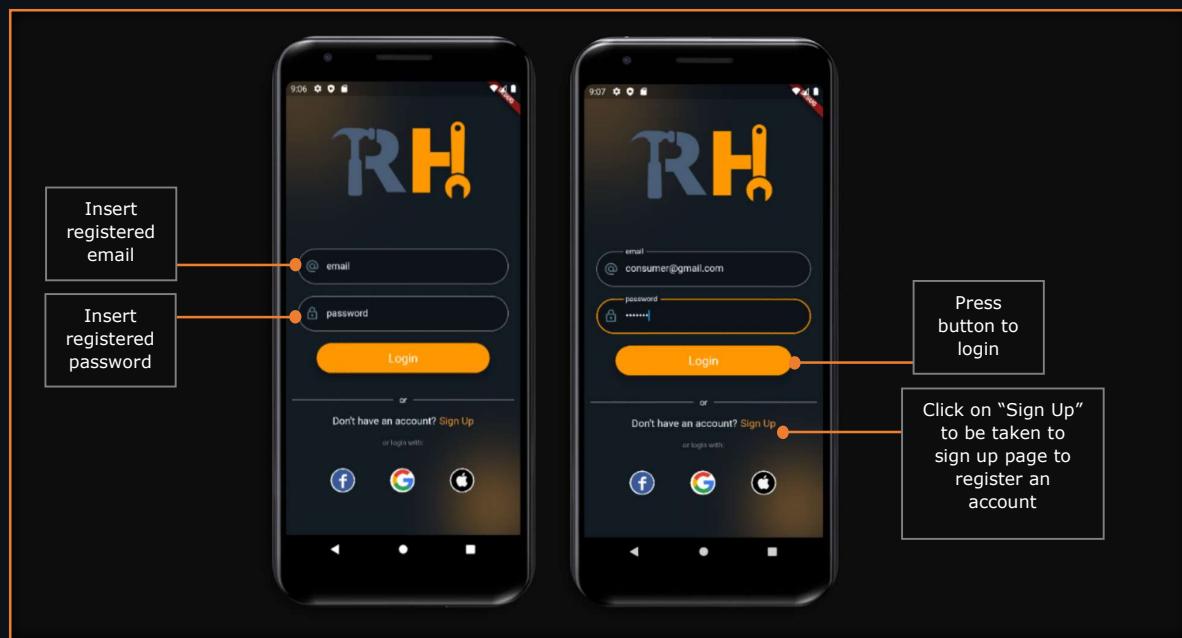
Any software installed on an IOS or Android mobile device.

Overview of main features

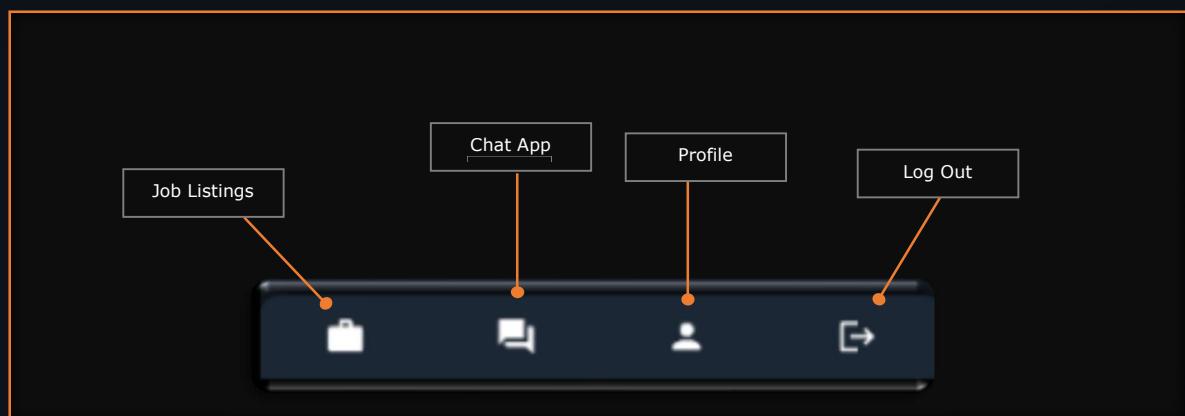
Logging In

The application opens up on the **login** screen. Here a user is able to log in to their registered account. In the case that the user does not already have an existing account they may navigate to the **sign-up** screen where they may create a new account as either a consumer or a tradesman. Once a user has entered their relevant information within the sign-up screen and has selected to sign up, the user will be prompted to enter an OTP sent to the email they registered with. A user will have to enter the OTP to be navigated back to login and be able to in.

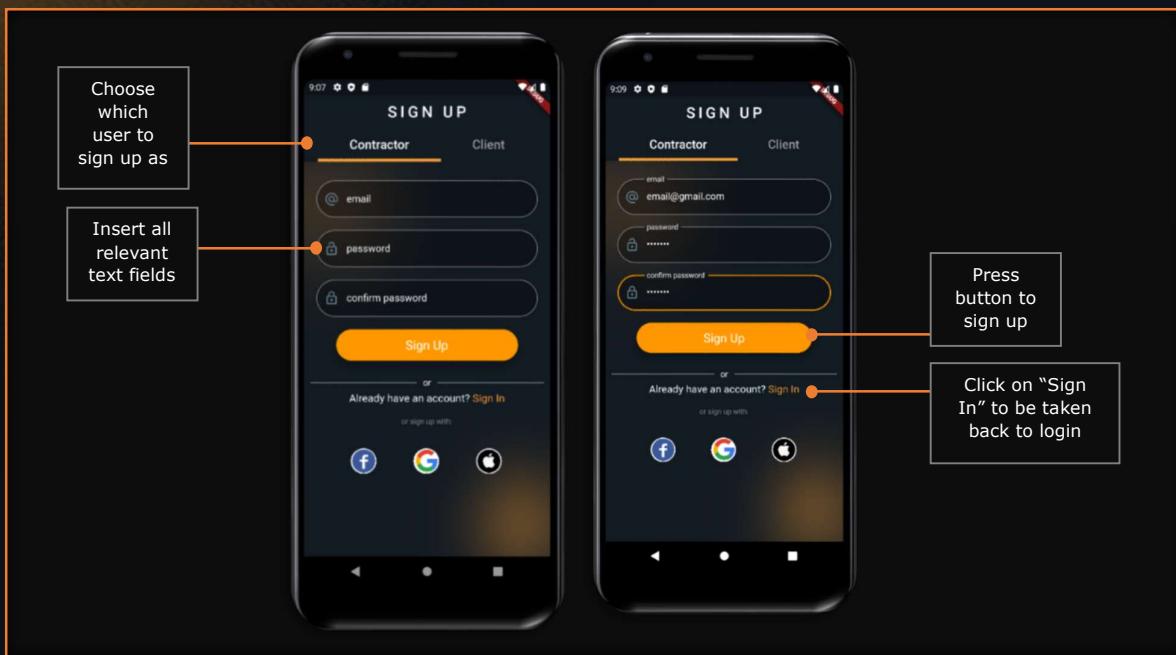
Login



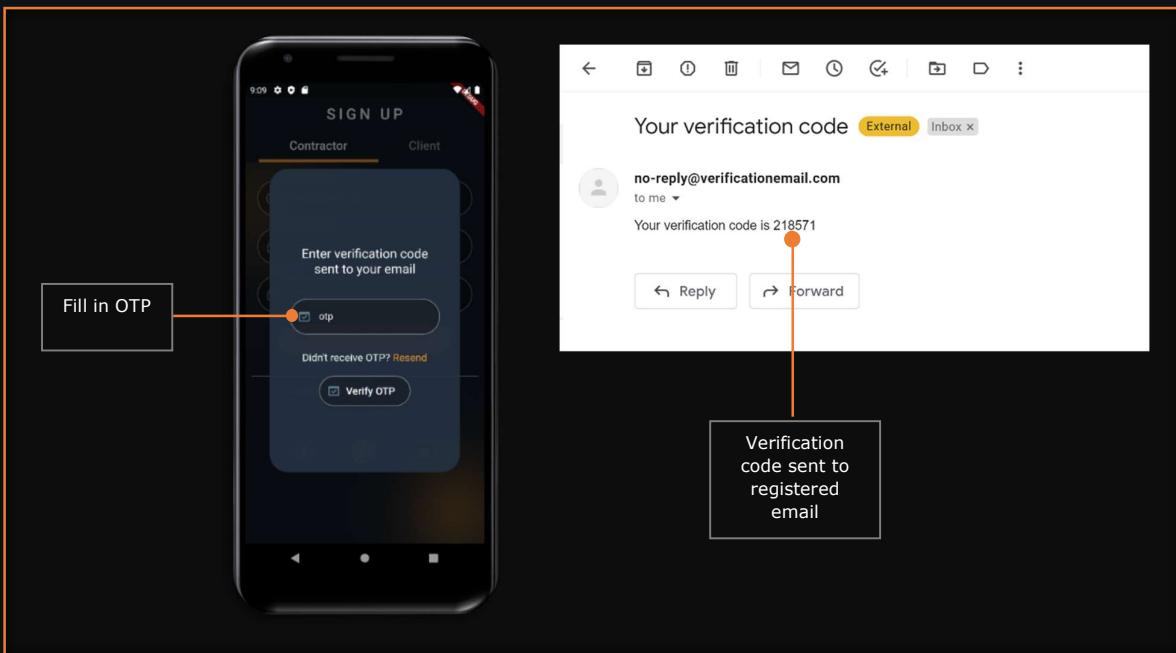
Navigation bar



Sign-Up



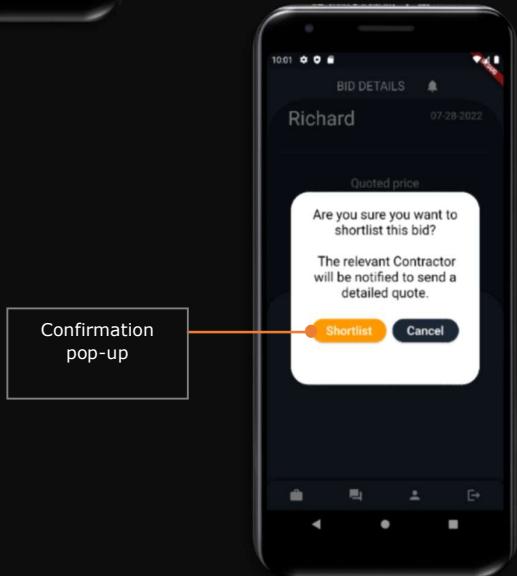
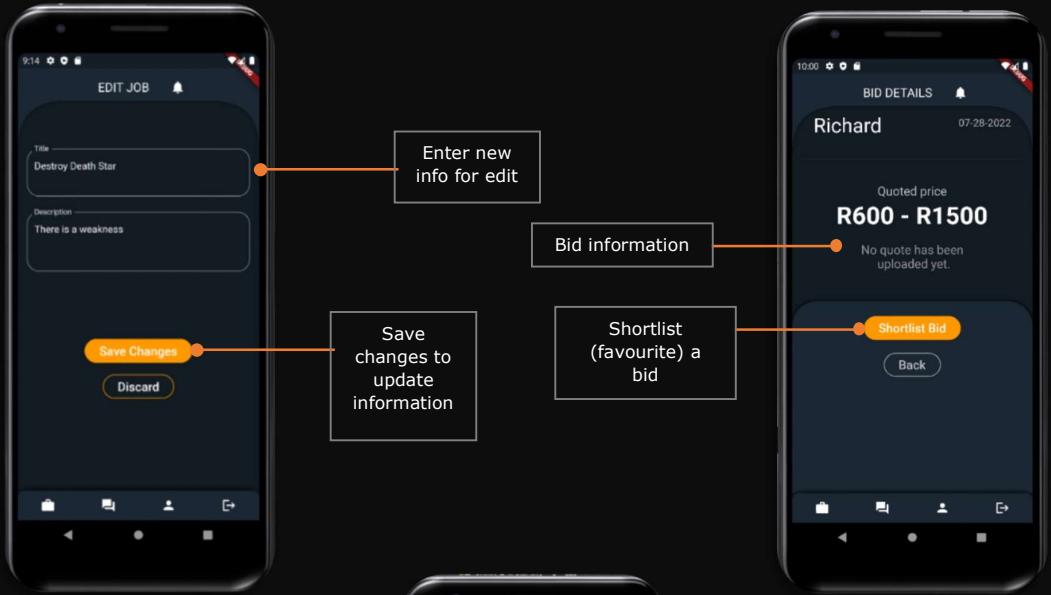
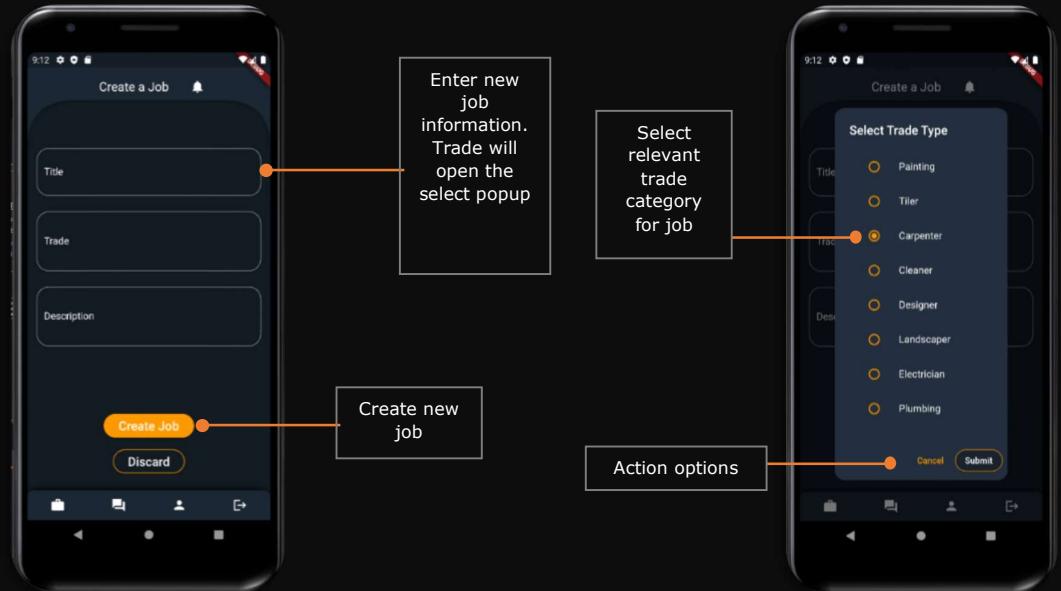
Sign-Up (cont.)

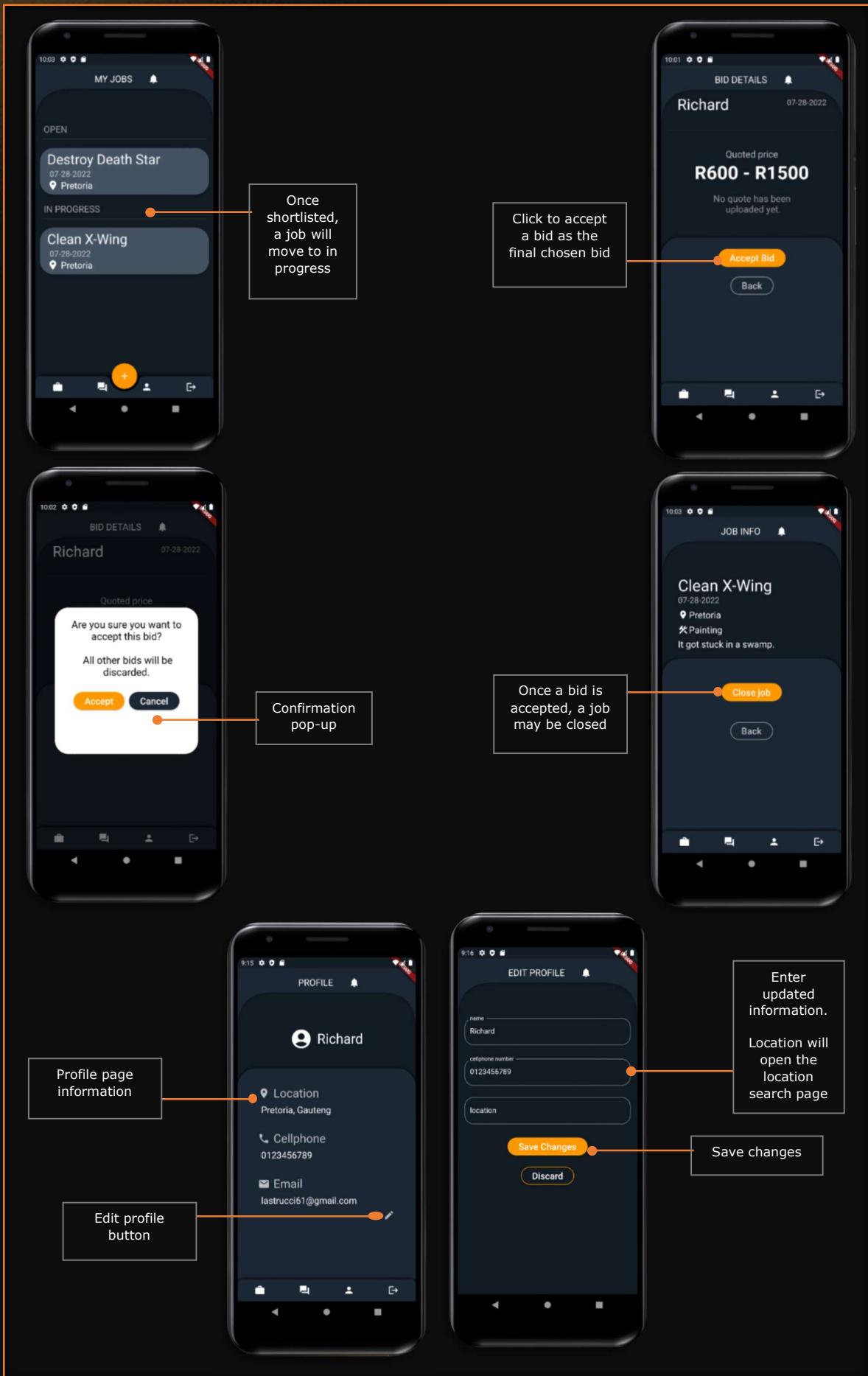


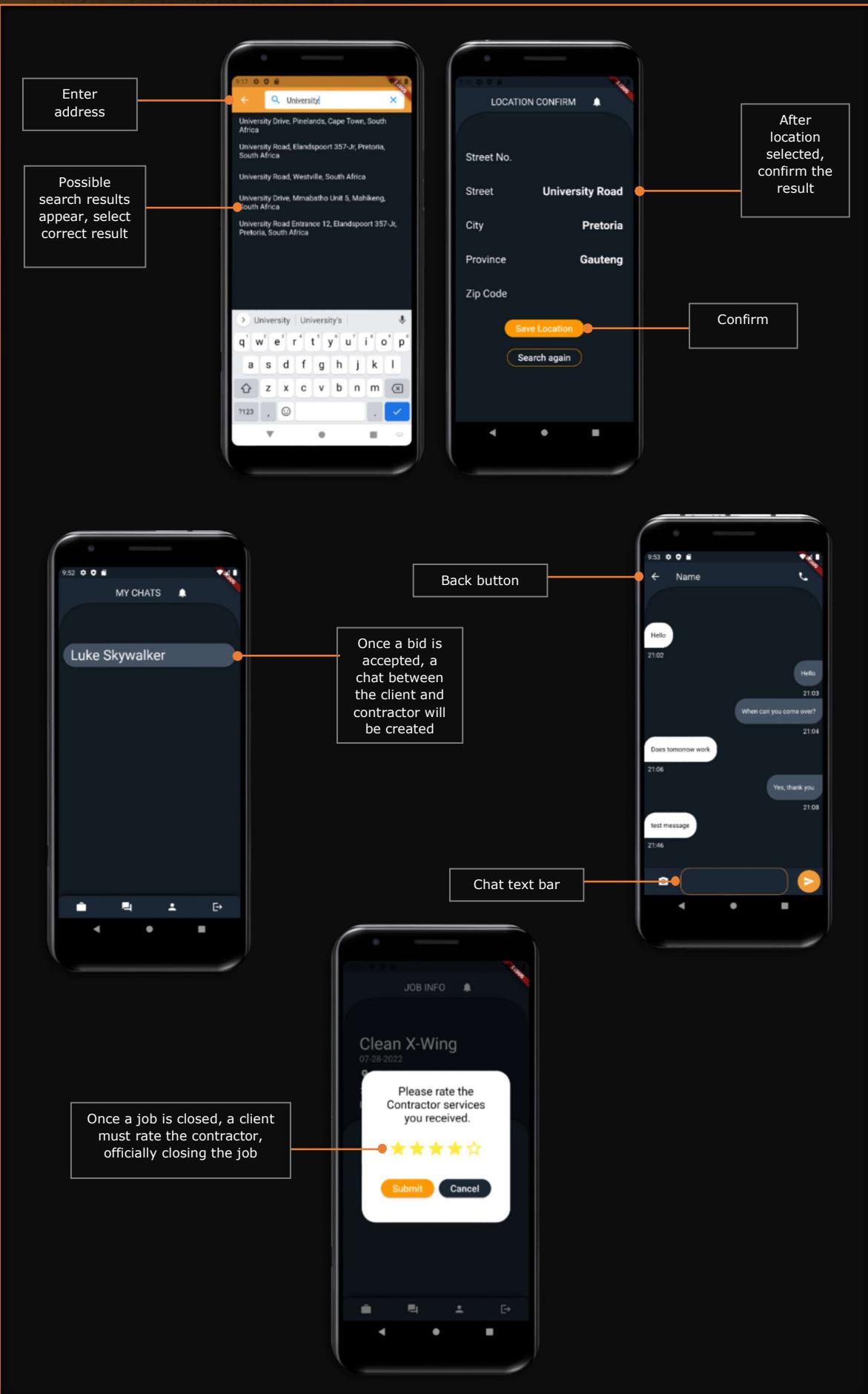
Navigating the consumer client

Once a user with a registered consumer account logs in successfully, they will be able to navigate the consumer client. Here they are able to view their current job advert listings, and create new adverts. Inside an advert detailed view, a user is able to view all received bids and shortlist and accept them. Additionally, clients may edit and delete their job adverts, view, edit their profiles and chat with accepted tradesman users









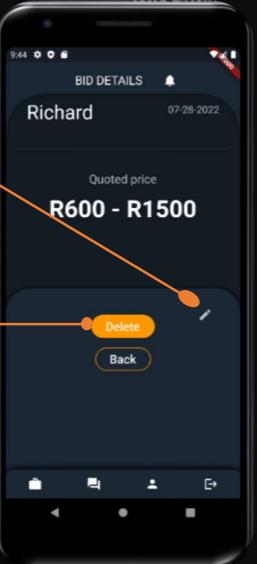
Navigating the tradesman client

Once a user with a registered tradesman account logs in successfully, they will be able to navigate the tradesman client. Here they are able to view all available job advert listings that are biddable. Within a detailed view of an advert, a user is able to place a bid on the advert through a pop-up, where they enter their proposed rates for the job and a short description. They can also view their notifications in an activity stream, view their profile, chat messages and edit or delete their placed bids, after viewing competitors bids.



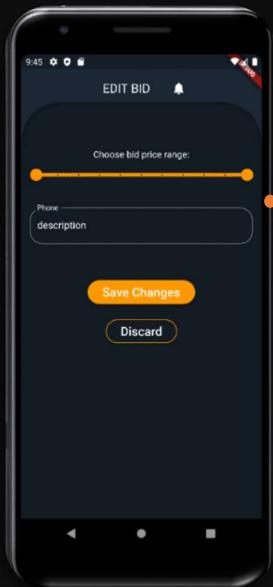


Click on a bid to view it. You may only view your own bids detailed screens



Edit bid

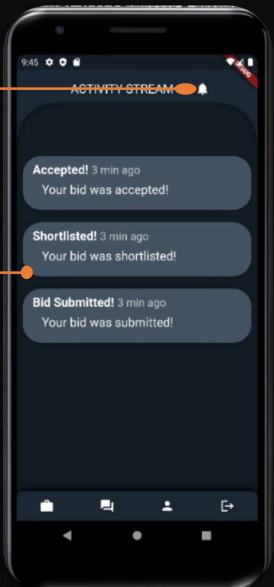
Delete bid



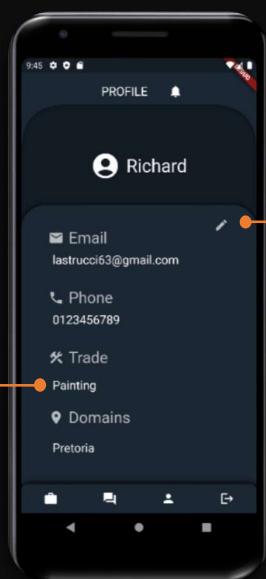
Edit bid information accordingly

To access activity stream

Notifications



Click to edit profile

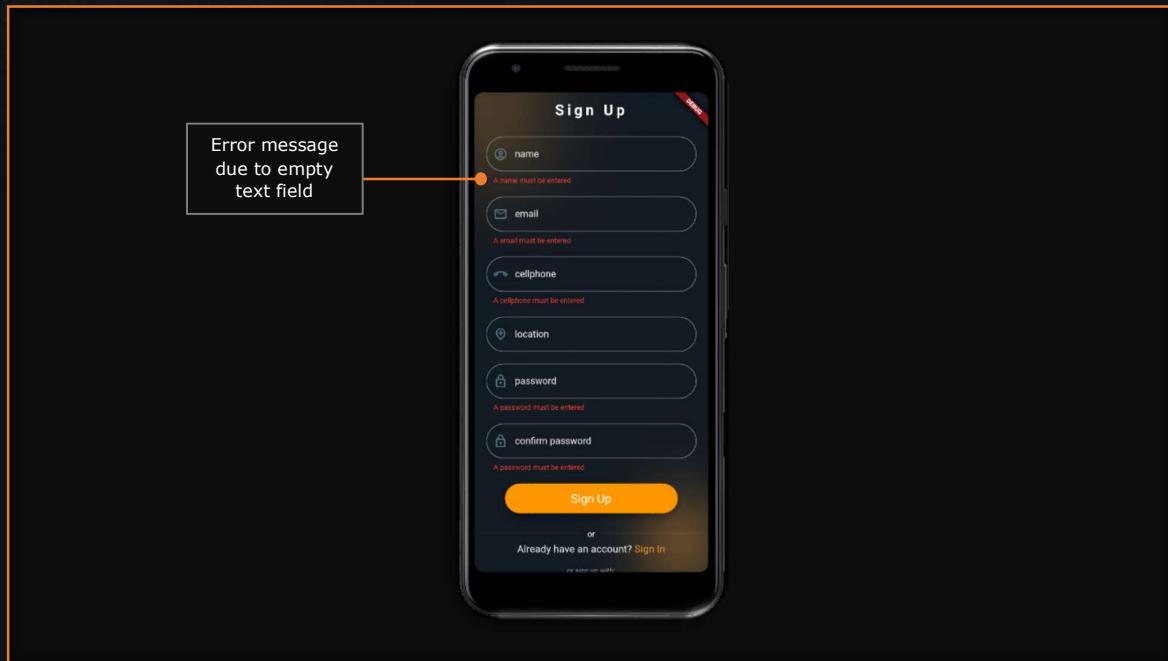


Trade and Domains are **scrollable**. Make sure to scroll on them to view all the information!



Error Handling

The app makes use of validation checking as a form of error handling. Within the sign-up page, a user will be prompted to fill in all available text fields if they fail to do so before pressing the *Sign-Up* button. These prompts take place as error messages shown beneath the relevant text fields.



E N D
