Coding Standards

Naming Conventions

General

- Classes are in UpperCamelCase.
- Functions and variables are in lowerCamelCase.

AWS

- All resolvers are written in nodejs, written in lowerCamelCase and postfixed with “Resolver”.
- Database partition and sort keys are prefixed with a string of characters followed by a #, followed by a uuid. This does not apply to certain cases such as domains or trade trade.
- Normal database values are in snake_case.

Flutter

All files are in snake_case and end with the appropriate postfix, the postfixes are the following:

- action - for all redux actions.
- model - for all models used in the redux pattern.
- page - for all main pages.
- widget - for widgets that make up pages.

In certain special cases theses conventions are not followed.

Project Structure

The project is divided into two main parts.

- The main app which can be thought as a main function in languages like C.
- Libraries which use one another as well as the which the main app calls.

All libraries as well as the app contains a folder called lib which is where all the logic and code goes.
There is also a pubspec.yaml file which contains all dependencies.
Each library contains a readme which explains the purpose of the library.

App

Apps lib folder solely contains a main.dart which is the entry point for the program and sets up our navigation routes.

Libraries

Libraries are further split into two subcategories:

UI Libraries

All UI libraries have the following subfolders:

- pages - contains all pages for the library.
- widgets - contains custom specialised widgets for the library.
- methods - contains methods for the library.

The general library is special so it does not have specialised components and does not contain pages.

- authentication - contains all authentication related ui e.g. login/signup.
- tradesman - contains all tradesman related ui.

- consumer - contains all tradesman related ui.
- chat - contains all chat related ui.
- notifications - contains all notifications related ui.
- general - contains all none specialised widgets used throughout the project.

Backend Libraries

- amplify - contains AWS Amplify code, this includes
  - The schema for graphql.
  - The resolvers code.
  - The configuration for the endpoints i.e. AWS AppSync.
  - Along with a lib folder there is also a sub amplify folder which contains all the functions and schema.
- geolocation - contains all backend logic for interacting with the Google Places API.
- redux_comp - contains the AppState, actions/reducers and models required by the redux pattern.

Below are some images of the folder structure. Note not all files and folders are shown only the most important ones.