03/02/2022

I began to create the database for the application using cloud firestore. Using the entity model diagram that I’ve designed previously, I created two collections. The first collection named ‘available seats’ will have a document titled ‘vehicle’. This will contain every individual vehicle associated with the transport operator. Each vehicle document will have a field for each seat with a Boolean value of true or false, which will represent whether a seat is available or not

The second collection is named ‘users’. It will have a document containing all the users that have signed up to the application. Each user will have a collection for tickets and a collection for travel cards, as well as a field for address\_line\_1, address\_line\_2, city, county, phone number and post code.

The ticket collection will contain a document that will store multiple tickets, with each ticket having a field for activated, amount, date, operator, price, route, transaction ID and type. Every ticket will have one transaction associated with it as the tickets are single use only.

The travel card collection will also contain a document which will store multiple travel cards. Each travel card will have a field for balance, bus cap, multimode cap, train cap, tram cap and type. A collection for the transactions associated with the travel card is included. Similar to the ticket collection, each transaction will contain the amount, date, operator, and route.

I’ve setup the pre-requisites for firebase in the flutter environment by following the instructions from <https://firebase.flutter.dev/docs/overview> and watching some YouTube tutorials <https://youtu.be/sfA3NWDBPZ4>

04/02/2022

I’ve set up the file structure in my application as follows:

* Config – Any application specific configurations e.g. colours
* Models – Groups of data
* Pages – Screens
* Services – Logic
* Widgets – Reusable widgets
* Providers – State management

I began the implementation of the sign-in and sign-up pages. My goal is to provide a form allowing the user to sign up or sign in with their credentials. For now I will be only implementing sign in functionality using email/password to test it out. In the future, I would like to integrate 3rd party services such as Google and Apple sign in.

On the sign in form, I’ve added a basic form with a field for the email and password. On the sign up form, I’ve added a basic form with a field for the email, password, first name, last name, phone number, address line 1 & 2, city, county and postcode.

Next I connected the application to my firebase authentication service using the firebase auth package. I created a file called firebase\_auth.dart. This file will contain functions relating to the interaction of authenticating with firebase i.e. Sign in, sign up, sign out and obtaining specific data relating to the user.

I created a user model object names user.dart. This file has 2 classes. The first is userModel. This class will provide us with the uid (User Id) only when we want to authenticate with firebase instead of the various parameter we can receive. The second class is UserInformation. This class will provide us with additional information that isn’t provided by firebase but is obtained through the sign up page. The parameters that will be accessed is the email, first name, last name, phone number, address line 1 & 2, city, county and postcode.

My next step was to create a wrapper for the application. If the user is already signed in, then they don’t need to go through the whole phase of signing in again, they’ll just be redirected to the home page. Similarly, if a user isn’t logged in, then we want to guide them to the sign in or sign up page.

I've added the provider dependency to use in our application. Using streams, it will allow us to access the incoming data across our application and handle its state i.e. display the sign in/sign up page or the home page

Later as the application develops, I will add a splash screen and walkthrough/onboarding screen before the sign up/sign in pages. For now I just want to make sure the basic functionalities are working and implemented.

So far, I’ve been able to successfully create a user with the sign up page and sign in to the application. What I now want to achieve is connecting to the database and creating a user collection with the document id equating to the uid of the user. This will mean that I can store information about the user with their specific id like the parameters in the sign up page and the ticket/travel card information.