Smart Calorie Tracker

1. **The User Interface:**

1.1 The User Interface was designed in Figma.

1.2 The website [Behance](http://www.behance.net/) was used for artistic inspiration and to see how

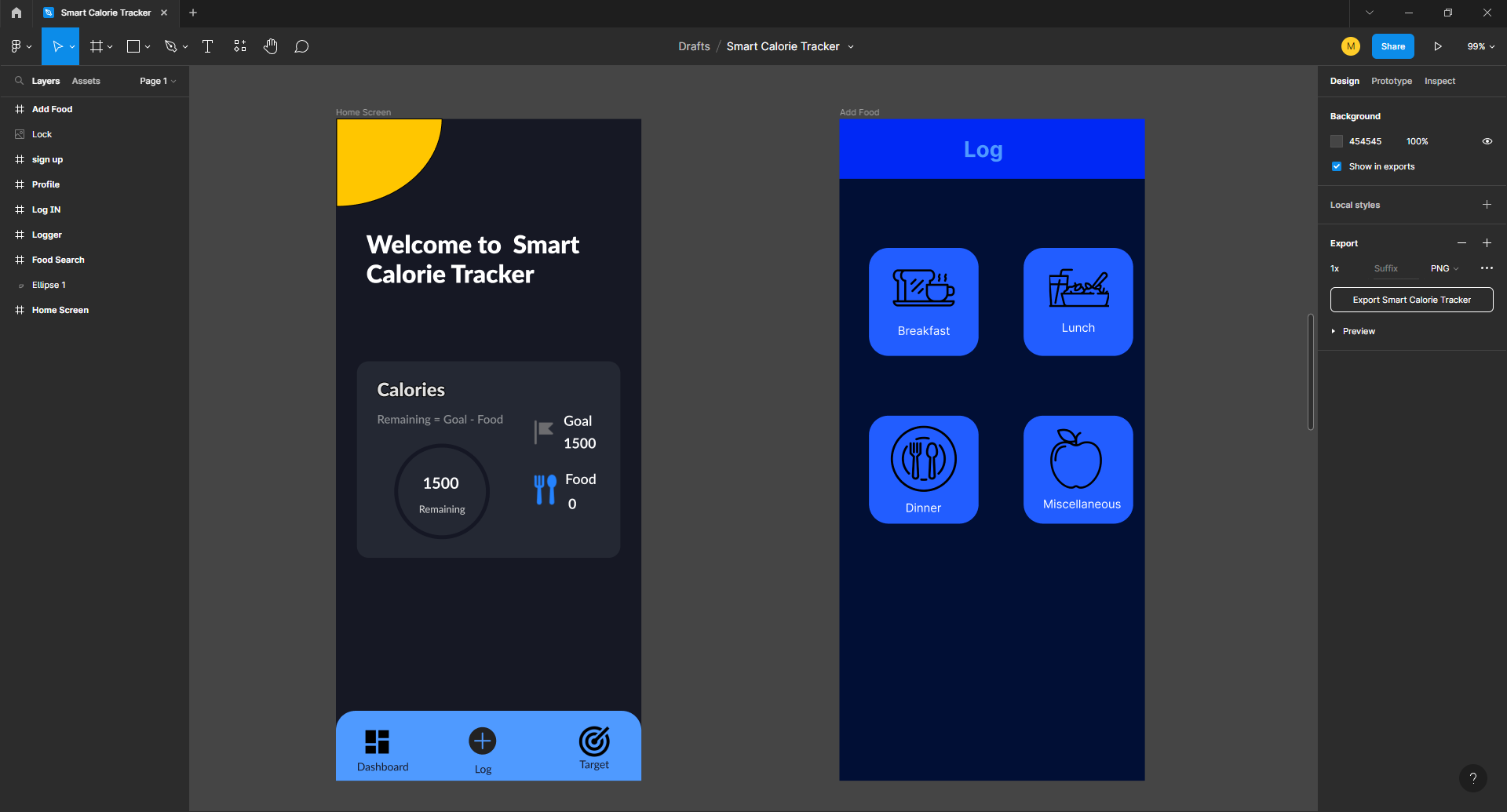
professional developers design their User Interfaces.

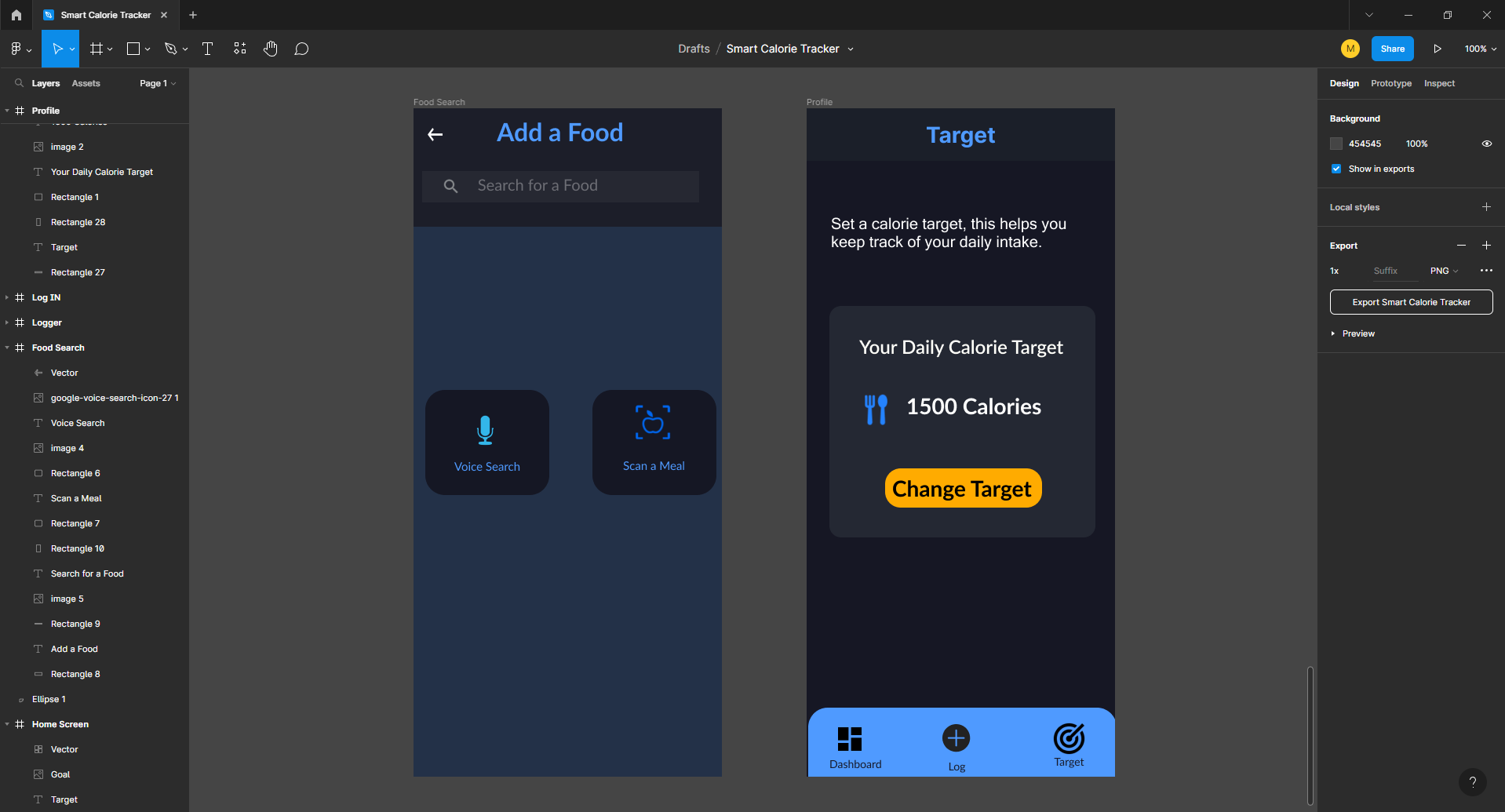
1.3 The design was based on the philosophy of creating an easy to use, intuitive

UI with rounded corners and using dark contrasting colors to relax the minds of

the users.

1.4 Some of the Figma Screens showing off the design -





1.5 The UI that was designed in figma was then implemented in android studio by

creating activities and various fragments.

1.6 On app startup the login activity which houses the Google login button of firebase

is displayed. After the user logs in the main activity is displayed.

1.7 The main activity houses the navigation bar which was implemented using

Google’s Material Design Library. It can has 3 buttons for easy navigation

throughout the app.

1.8 All other functionalities are implemented by using fragments.

1.9 Fragments are great in that they don’t take up the whole screen and can be

swapped with other fragments while keeping the same activity.

1.10 The main activity houses the dashboard fragment at start which displays the pie

chart that displays the critical information about the user’s calorie consumption.

1.11 The log fragment displays the meal choices for the food that is to be added.

The choices are breakfast, lunch, dinner and miscellaneous.

1.12 The add food option fragment displays all the ways the user can input what food

they have eaten.

1.13 The calendar fragment displays the calendar so that the user can input a date for

their meal.

1.14 The camera fragment is used to display the camera options if the user selects

the Scan option in add food option fragment.

1.15 The voice fragment displays the options for the user to record their voice for use

In adding food.

1.16 The search fragment is used to display the search bar that the user can use to

add food manually.

1.17 The search item fragment is also developed to display to the user manual

choices to add food.

1.18 Both of the above fragments differ in size and dimensions. Thus they are used to

display manual add option as and when required.

1.19 The add food fragment is used to display all the nutritional information about the

food that was added by the user. It also has Input fields for the user to enter how

much food in grams they have eaten.

1.20 The Target fragment display the target set by user and has input fields to take in

new target value as well.

1.21 The activities and fragments mainly use linear layout as well as constrained

layout in its design.

1.22 The activities and fragments mainly have standard android views in EditText,

TextView, Radio Buttons.

1.23 The design also relies heavily on Google’s Material Design Library to implement

some of its design.

1.24 Buttons with icons are implemented by using google’s material button. The

navigation bar and some other design elements use Google’s Material Design

library.

1.25 The app has many other design files but the ones listed here are the major ones,

Other files implement very small functionalities.

1. **The Backend:**

2.1. Firestore is used for Backend.

2.2. The Backend comprises 3 tables mainly user-info ,users,foodLimit.

2.3. The users table comprises of a collection uid (Unique Id) and then date and then

Food type (BreakFast, Lunch ,Dinner ,Miscellaneous) and then Food is Added.

2.4. Food is collected in object of FoodDatabase.java which have parameters :

Dish\_name, dish\_calorie , dish\_consumption.

2.5. The user-info table consists of uid and then stores a field daily-limit which serves as a global target for the upcoming days once set.

2.6. The foodLimit consists of uid which comprise of Date and then a document daily

which stores consumption ,daily\_limit.

2.7. The dailyLimit is collected in the object of the dailyLimit.java file .

2.8. Each day consumption is set to zero.

2.9. The Target and dish\_consumption accept only non negative numbers.

2.10. The user on login, check is done by checking in user-info if any record exist and

If it exists then update the daily\_limit to that value else initialize to 2300.

2.11. The google Authorization service provided by firebase is used to login.

1. **The ML Models:**

**Image model**

To generate a model for the purpose of image classification, we used Convolutional Neural Network(CNN). We tried different combination of hidden, convolution and max pooling layers to get the highest accuracy possible. To further enhance the performance of our model we integrated it with state of the art model for images EfficientNet we tried different models from B1 to B7 and performed hyper-parameter tuning to get the highest accuracy. With the help of EfficientNet and hyper-parameter tuning we were able to achieve about 87% accuracy. For further improvement we tried changing the dimensions of the image and then we fine-tuned the EfficientNetB5 with our data and we were able to gain more than 95% accuracy making the model ready to be used in production.

**Voice model**

For the purpose of voice analysis and identification of different dishes, we used tensorflow and RNN. Initially we were able to achieve an accuracy of 79% but to further improve the model we explored other state of the art models and algorithms. Then we trained the model with LSTM and were able to push accuracy to 85%. After working on different combinations of layers of GRU, LSTM and bidirectional LSTM, we were finally able to get an accuracy of 94% which made our model ready for further integration into the project.

1. **The API:**
   1. The application uses the EDAMAM Food Database API to fetch the calories contained in any food item.
   2. The API used is currently a trial plan for developers. If the application is used for professional use, the individual must buy the Edamam API.
   3. The challenge faced during this process was integrating the API calls to handle the asynchronous calls.
   4. The application uses the “Retrofit” framework and “GSONParser” to fetch the required data from the Edamam API.
   5. We used the View Models classes to handle the asynchronous calls and to manage Recycler Views used to present data fetched from the API.
   6. Another challenge was using Recycler View without prior knowledge of its working and uses.
   7. Using an adapter and view model together to create a scrollable list to display all the data fetched and the details about a food item.
2. **The App Structure:**
   1. The Application comprises mainly two Activities, the Main Activity and the Login Activity.
   2. The Login Activity:
      1. The activity consists of a single button that uses the Firebase authentication services to sign a user in using his google account.
      2. Once the user is logged in, the User is then redirected to the Main Activity, where the Dashboard screen greets him.
   3. The Main Activity:
      1. This activity has all the fragments used in the application to do various tasks.
      2. The Tasks are mainly of three categories: the dashboard, the log, and the target screen.
      3. The Dashboard Fragment, where the user can look at his daily target and the daily consumption of calories.
      4. The Log Fragment, where the user can check his past and currently consumed food items, which are categorized according to the meal, i.e., breakfast, lunch, dinner and miscellaneous. Also, the user can manually add the food items eaten using three different ways: by taking a picture of the food item and saying out loud what he ate to the app.
      5. The Target Fragment, where the user can change the daily target according to his discretion. And the user has been provided with a button to Log out of the application on this page.
      6. The scanner fragments ask the user for camera permission if not already given, and then the user can press the take an image button and take a picture. Then, the ML model will predict the food item and ask the user to enter the amount of food eaten in grams. Accordingly, the application will add calories consumed to the consumption.
      7. Similarly, the voice recorder fragment will ask the user for mic permissions and the user can speak to the application about what he ate. The application will ask for the user’s confirmation and the amount consumed in grams and change the daily consumption accordingly.
      8. Suppose the user is not satisfied with the application’s judgment about the food item at any point. In that case, the user can add the item manually by entering the name and amount in grams eaten by the user.