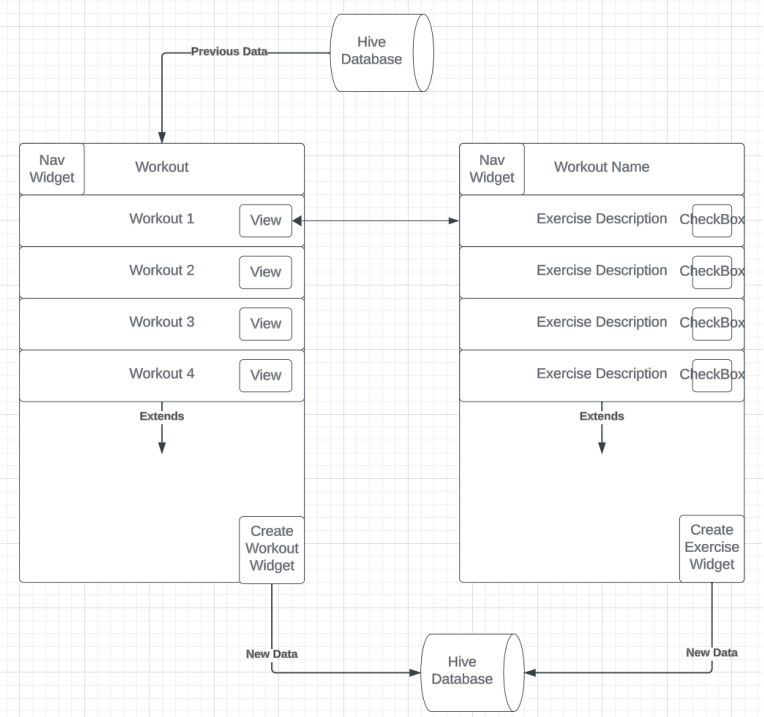
Design

Main Screen: Fitness Tracker.

- Users can input and view workouts that they have created.
- Users can view previously saved data.
- Users can create workouts and create exercises inside of those workouts which can then be stored in the database.
- Users can navigate ‘Nav Widget’ between application screens through a widget e.g., a drawer.
- Users can check off exercises as they complete them.
- Users must be signed in.



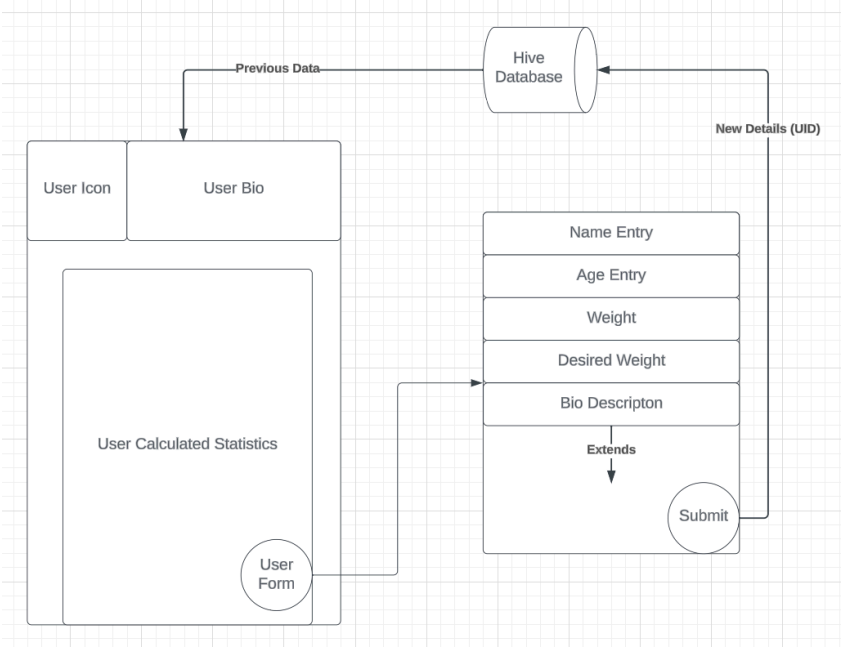
Functions:

- saveWorkout(): Will save a workout and display on screen
- goToWorkoutPage(): Pushes the context back to the main screen (Context back to original state).
- cancelWorkout(): Delete a workout and change display on screen
- signUserOut(): Will allow the user to return to the login screen.
- createNewExercise(): Elicits form and allows users to create the exercise by inputting data.
- cancelExercise(): Will allow the user to not submit the creation of the workout form.

User Profile

- Users can view personalised statistics based on information they have inputted through ‘User Form’.
- Users can view previously calculated statistics based on previous user form information.

- Users can start a ‘User Form’ and enter information which submits the information to the Hive database.
- Users can change and update their personal information at any time.



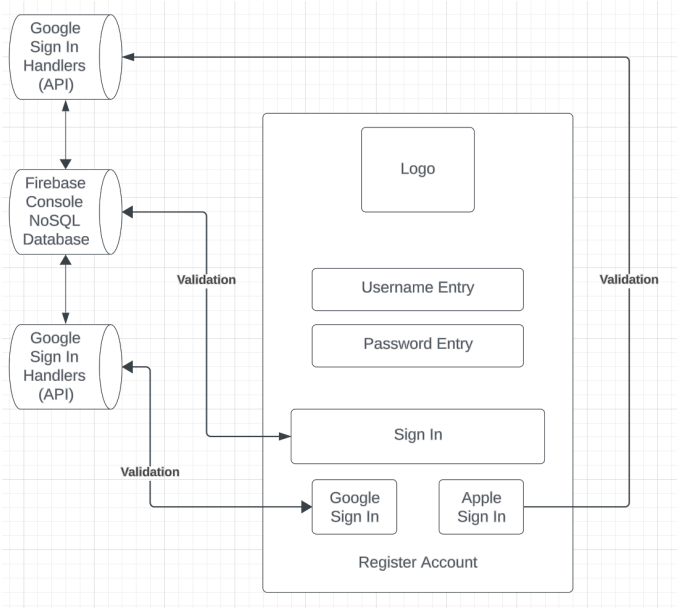
Functions:

- loadUserDetails(): Find previous saved information and update the profile information accordingly.
- setState(): Sets the state of the screen/widgets
- addUserDetails(): Produces form and allows the user to input their information.
- saveProfile(): User can save their information inputted on the form
- cancel(): User can return to the screen from the form if doesn’t want to submit the form.
- gatherStatistics(): This will use all of the user information to calculate statistical values to then display on the screen.

Log In Screen

- Interchangeable with Register screen for reusability.
- All information is validated before submitting to APIs and Firebase databases.
- If sign-in is unsuccessful then an error message will be elicited to the application.
- Google and Apple sign in APIs used to increase the speed of logging into the system.
- The user can use the Register Account button at the bottom to switch the next screen displayed below.

-



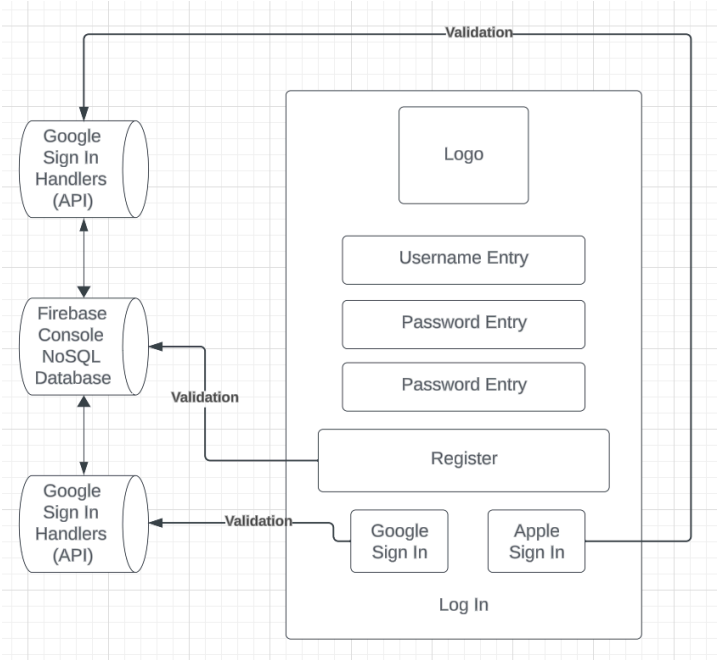
Functions:

- showErrorMessage(): User inputs invalid information and submits so error information is elicited to the console.
- signUserIn(): Will cause a communication with the Firebase Console to check user information against database records. If successful, the user is pushed to the main screen, else invalid information error is displayed.

Register Screen

- Similar to the Login screen to improve time efficiency.
- APIs still requested in case users don't want a manual sign-up process.
- Two password entries to allow for ensuring password matches and is approved for strong authentication.
- If all test conditions are met, the username and password are stored in a collection in the Firebase database.
- The Login button at the bottom allows users to switch to the screen above and log-in manually etc.

-

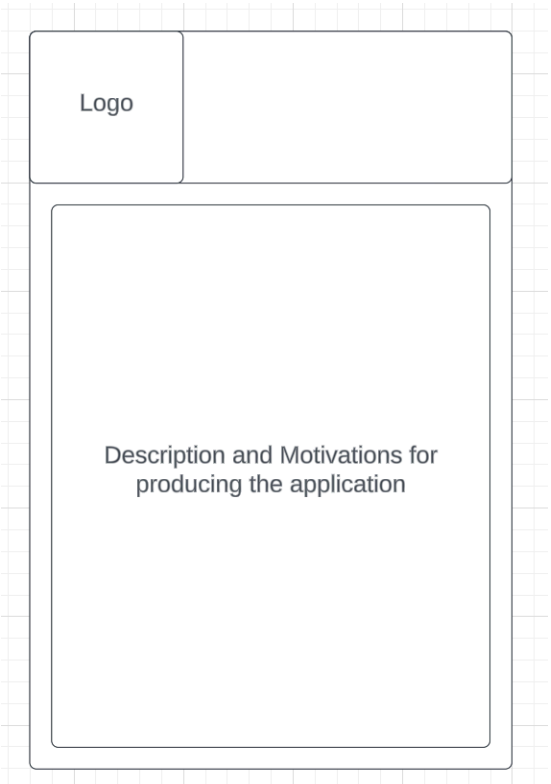


Functions:

- Validate(): Ensures email is valid e.g., ends in @gmail, Passwords are strong e.g., longer than 8 characters.
- signUserUp(): Information has been validated client side and will now be validated server side (Firebase Console) if successful then a record is created
- showErrorMessage(): Will elicit the error associated with the registration details.

About Us Screen:

- Very simplistic fundamental of the application.
- This description will showcase the motivations and the team behind the application in order to show a sense of community.
- This could potentially be expanded on for future development.
- Will showcase a level of branding to the application.



Functions:

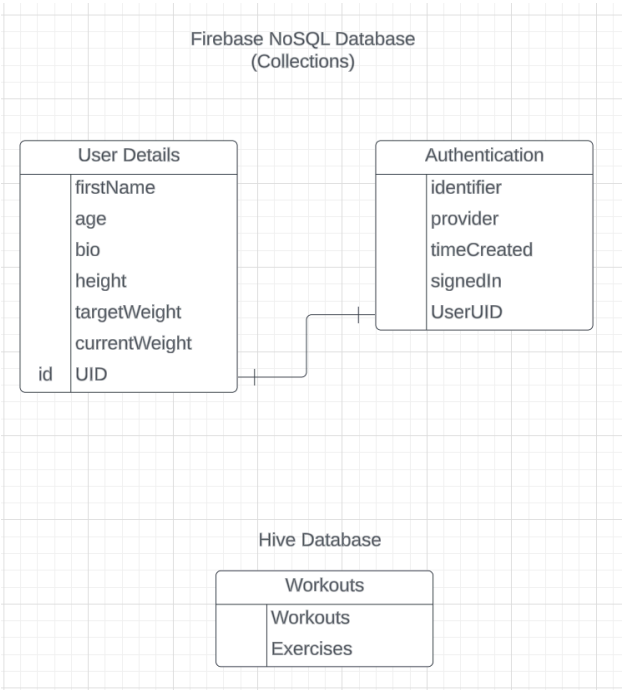
- Any extra features or stored company elements including manager profiles etc.

Firebase Console NoSQL Database:

- User details consist of the information that is stored from the user.
- This collection will be used to distinguish between unique users on the application.
- The UID is the primary key that is used to link users to their stored information and to load previously stored information.

Functions:

- storeUserDetails(): Will create a record or replace a record associated with UID and store validated information based on the client.
- getUserDetails(): Will obtain the record associated with the UID for formatting and display of user information.



Hive Database:

- Stores the relevant workouts and the associated exercises with the workout.
- The use of both Hive database and Firebase Console database is an experiment to determine the fundamental storage component for future expansion of the application.

Functions:

- initialiseWorkoutList(): Obtain the list of workouts associated with the users UID.
- addWorkout(): Add workout to the list of workouts associated with the users UID.
- addExercise(): Add exercise to the workout associated with the users UID.
- getRelevantWorkout(): Will obtain a workout from the user's list of workouts.
- removeWorkout(): Will delete the user's workout and associated exercises.
- removeExercise(): Will delete the exercise within the user's workout.