Group E / Iteration 3

Fitness Mobile

Iteration 3 Improvements

1. UI Improvements
    a. Bottom Bar For Navigation
    b. New icon
2. Sleep Tracking
    a. Allows the user to track sleep amounts for each day
3. Google Fit Integration
    a. Pulls Calorie Burned and Step Count information on Google Fit into the app
4. Initial Home Page User Interface
    a. Provides a centralized dashboard for display of all of the information within the app
    b. Initial code only, further work will occur in Iteration 4

Added Databases

```dart
import 'dart:io';

import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';
import 'package:sqflite/sqflite.dart';
import '../sleep_log.dart';

class SleepDatabase {
  // DB info
  static final _databaseName = "dbsleep.db";
  static final _databaseVersion = 1;
  static final tableName = "sleep";

  // DB colums
  static final columnId = '_id';
  static final columnStartTime = 'start_time';
  static final columnEndTime = 'end_time';

  // beginning with an '_' means private constructor
  SleepDatabase._privateConstructor();

  // Singleton ~~ only one instance
  static final SleepDatabase instance = SleepDatabase._privateConstructor();

  // Database reference
  static Database _database;

  // get database function
  Future<Database> get database async {
    if (_database != null) return _database;
    _database = await _initDatabase();
    return _database;
  }

  // initialize database helper function for opening and creating database
  _initDatabase() async {
    Directory documentsDirectory = await getApplicationDocumentsDirectory();
    String path = join(documentsDirectory.path, _databaseName);
    return await openDatabase(path,
        version: _databaseVersion, onCreate: _onCreate);
  }

  // helper function to create SQL table
  Future _onCreate(Database db, int version) async {
    await db.execute('''
        CREATE TABLE $tableName (
            $columnId INTEGER PRIMAY KEY
            $columnStartTime INTEGER NOT NULL,
            $columnEndTime INTEGER NOT NULL,
        )
    ''');
  }
}
```

```dart
import 'dart:io';

import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';
import 'package:sqflite/sqflite.dart';
import '../sleep_log.dart';

class UserDatabase {
    // DB info
    static final _databaseName = "dbuser.db";
    static final _databaseVersion = 1;
    static final tableName = "user";

    // DB colums
    static final columnId = '_id';
    static final columnUsername = 'username';
    static final columnAge = 'age';
    static final columnHeight = 'height';
    static final columnWeight = 'height';
    static final columnBMI = 'bmi';
    static final columnTimestamp = 'timestamp';

    // beginning with an '_' means private constructor
    UserDatabase._privateConstructor();

    // Singleton ~~ only one instance
    static final UserDatabase instance = UserDatabase._privateConstructor();

    // Database reference
    static Database _database;

    // get database function
    Future<Databse> get database async {
        if (_database != null) return _database;
        _database = await _initDatabase();
        return _database;
    }

    // initialize database helper function for opening and creating database
    _initDatabase() async {
        Directory documentsDirectory = await getApplicationDocumentsDirectory();
        String path = join (documentsDirectory.path, _databaseName);
        return await openDatabase(path, version: _databaseVersion, onCreate: _onCreate);
    }

    // helper function to create SQL table
    Future _onCreate(Database db, int version) async {

        await db.execute('''
            CREATE TABLE $tableName (
                $columnId INTEGER PRIMAY KEY
                $columnBMI REAL NOT NULL,
                $columnTimestamp INTEGER NOT NULL,
            )
        ''');
    }
```

Interface to database

```
class SleepLogStorageProvider extends StorageProvider<SleepEvent> {
  @override
  void delete(SleepEvent item) async {
    //FoodDatabase.instance.deleteByTimestamp(item.time);
  }

  @override
  void write(SleepEvent item) {
    //FoodDatabase.instance.insert(item);
  }

  @override
  void writeAll(List<SleepEvent> items) {

  }

  @override
  Future<List<SleepEvent>> read(DateTime date) {
      Future<List<SleepEvent>> temp;
      return temp;
  }

  static SleepLogStorageProvider get instance => SleepLogStorageProvider();
}
```

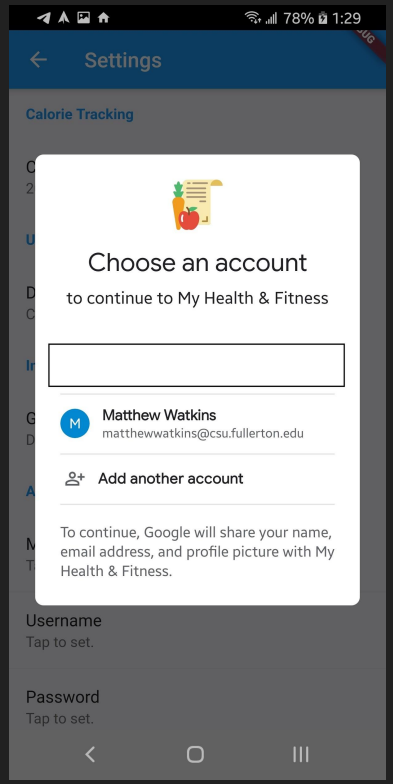
Homepage & Google Fit Integration

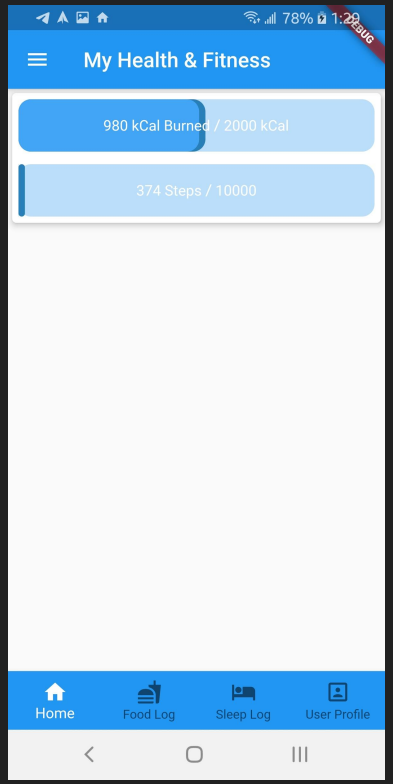
-Google Fit integration is functioning

- Users can connect to Google Fit, and their information will automatically be updated in the app.

- Initial homepage code and interface was added to demonstrate integration, further data will be added in the next iteration

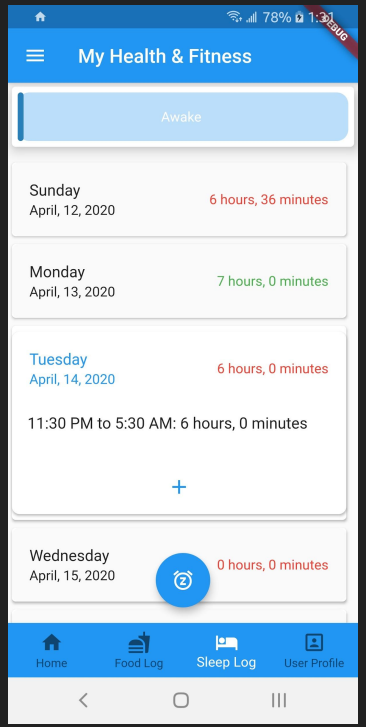
-UI was also updated with a bottom bar and a new icon (displayed to the left)

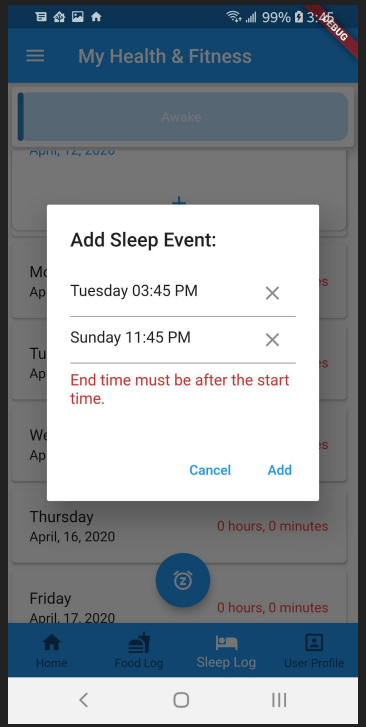




Sleep Tracking

- Users can track their sleep per day
  - Users can add sleep information through a dialog or with a start and stop button
- Sleep amounts are displayed and unhealthy sleep amounts are highlighted in red
- Top bar keeps track of how much sleep has accumulated when using the sleep stopwatch
- Add sleep dialog properly handles error conditions





About Us / Help Center

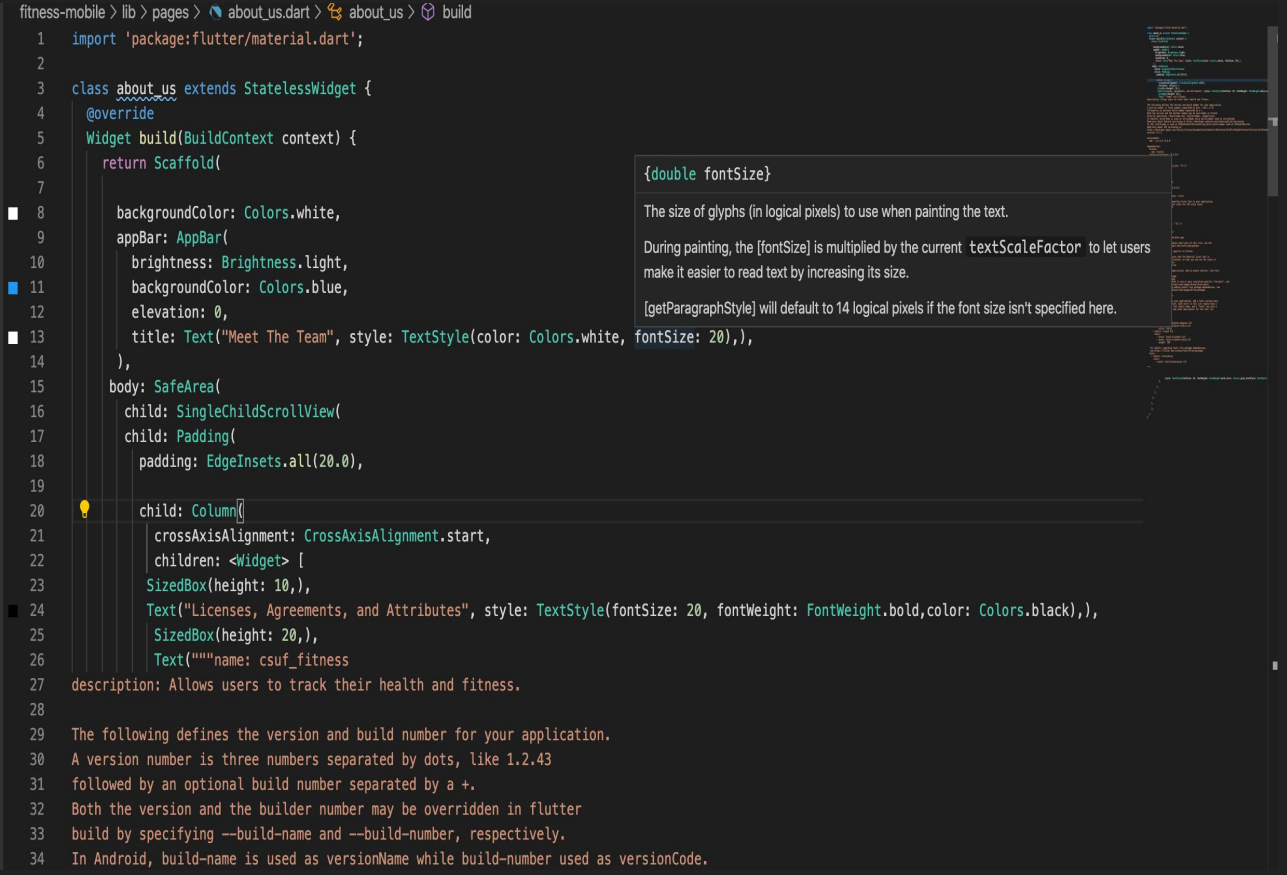
ss-mobile > lib > pages > user_page.dart > ...

```
  }

  Widget _buildUserPage() {
    return Center(
      child: Column(
        children: <Widget>[
          EditUserInfo(),
          FlatButton(
            child: Text("About"),
            onPressed: () {
              Navigator.push(context, MaterialPageRoute(
                builder: (BuildContext context) {
                  return Scaffold(body: about_us());
                },
              ));
            },
          ),
          FlatButton(
            child: Text("Help Center"),
            onPressed: () {
              Navigator.push(context, MaterialPageRoute(
                builder: (BuildContext context) {
                  return Scaffold(body: help_center());
                },
              ));
            },
          )
        ],


      ),
    );
  }
}
```

To start off, I had to link the about us and help center to the rest of the application.

So I made a Widget, which was a flat button to redirect to the new U/I page. This was done in the user profile.

About Us / Help Center



The about us is consisted of an app bar at the top. The app bar is used to let the user know what page they are on at all times. After I established that, I had to take the context of the .yml and add it under the licenses and attributes section. In order to add multiple line text, I had to use the multiple quotes. For example,

”””” Dependencies and licenses ””””

Also, had to use a scroll down widget to not overload the U/I page with pixels.

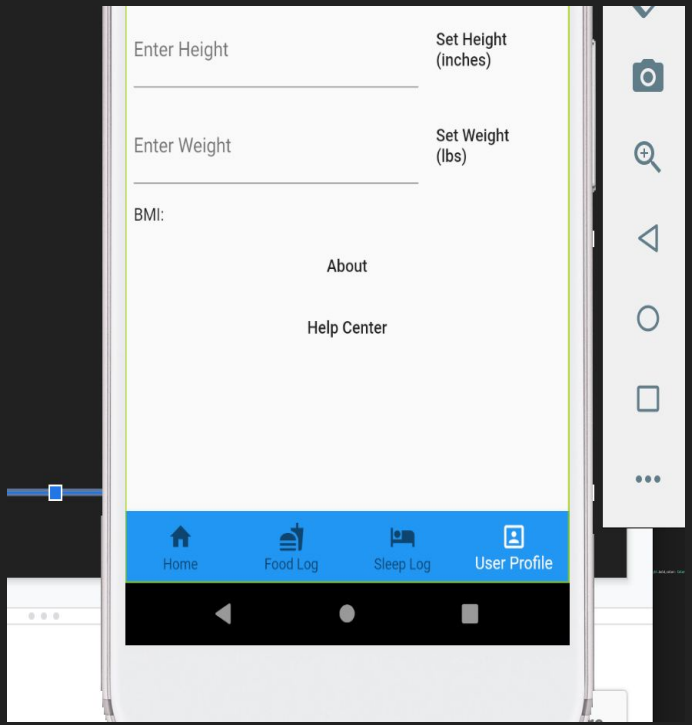
About Us / Help Center

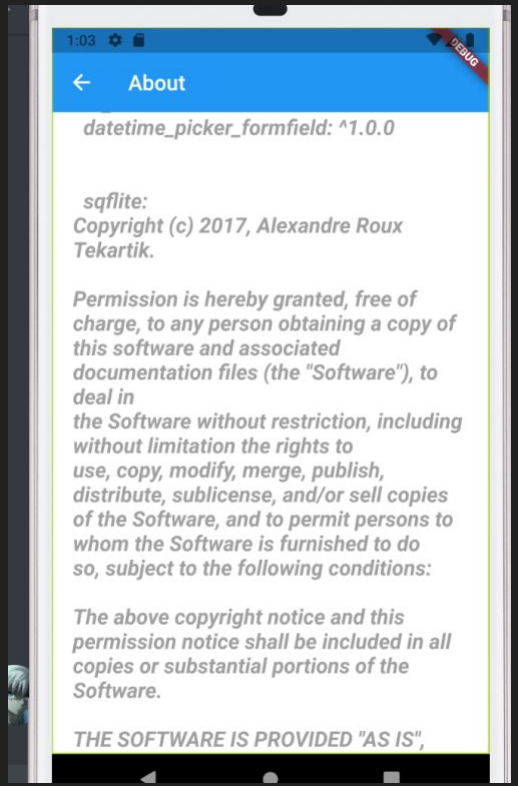
fitness-mobile > lib > pages > help_center.dart > help_center

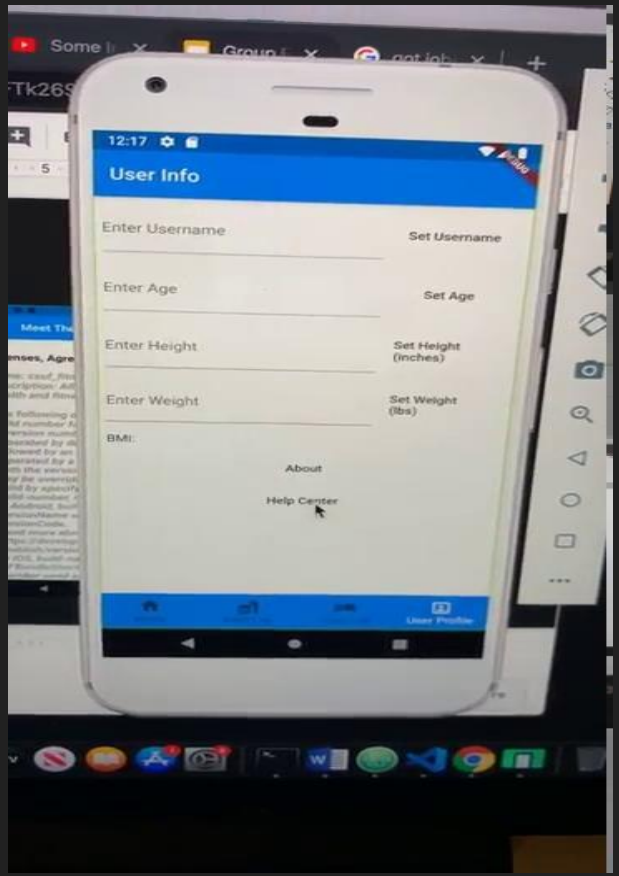
```dart
import 'package:flutter/material.dart';

class help_center extends StatelessWidget{
  Tween<double> _scaleTween = Tween<double>(begin: 1, end:2 );
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome To The Help Center ',
      home: Scaffold (
        body: Center (
          child: TweenAnimationBuilder(
            tween: _scaleTween,
            duration: Duration (seconds: 1),
            builder: (context, scale, child) {
              return Transform.scale(scale: scale, child: child);
            },

              child: Text(
              'Welcome To The Help Center ',
              style: TextStyle(fontSize: 9, fontWeight: FontWeight.bold,color: Colors.black,height: 0),
            )


          ),

      ),
      ),
      );
  }
}
```

After I finished the about us page, I moved on to the help center. The design of this page is a little different than the about us. The goal was to somehow mimic some of the artwork that goes into Twitters UI design. For that, an animation was implemented. To bring a more of a welcoming interaction with the user after being redirected to the help center.

About Us / Help Center







Iteration 4 Goals

- Weight tracking
- UI refinements
- Update the homepage with more user information
- Suggestions for daily calorie counts based on user profile