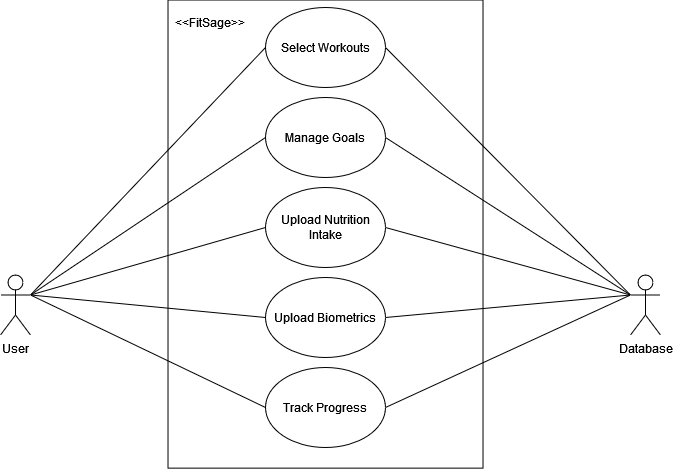
**Project Outline Document**

**Project Name:** FitSage

**Team Name:** Team 11

**Members:** Ryan Ferguson, Daniel Machado, Jim Pamplona, Jack Lee

**Use Case Diagram:**



**Use Case Descriptions:**

**UC1:** Select Workouts

**Description:** A user wishes to work out using the FitSage app. The user will have already uploaded their biometric information and selected a goal. The user will select muscle groups they wish to exercise, and the app will provide a list of recommended workouts involving those muscle groups. The recommended list will include information regarding that workout, and the user will be able to interact with workouts in the list.

**Actors:** User, Database

**UC2:** Manage Goal

**Description:** A user wants to select a goal to follow or change their current goal. The user may or may not have already uploaded their biometric information. The app will update the database with the user’s current goal.

**Actors:** User, Database

**UC3:** Upload Nutrition Intake

**Description:** A user wishes to upload nutrition information from food they ate. The user will have already uploaded their biometric information by creating their account. The app shall display the calories from the Nutrition Facts label so that the user can verify that the information was correctly interpreted. The calories will be added to the daily total in the database.

**Actors:** User, Database

**UC4:** Upload Biometrics

**Description:** A user wishes to upload or update their biometric information (age, height, weight), and the database stores the information. The user may or may not have already uploaded their biometric information. The app shall calculate the user’s Body Mass Index and Body Fat Percentage metrics. The app shall display them on the screen and upload them along with the biometric information to the database.

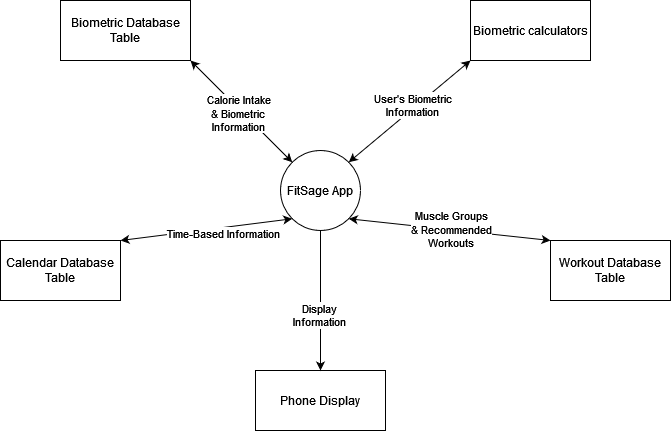
**Actors:** User, Database

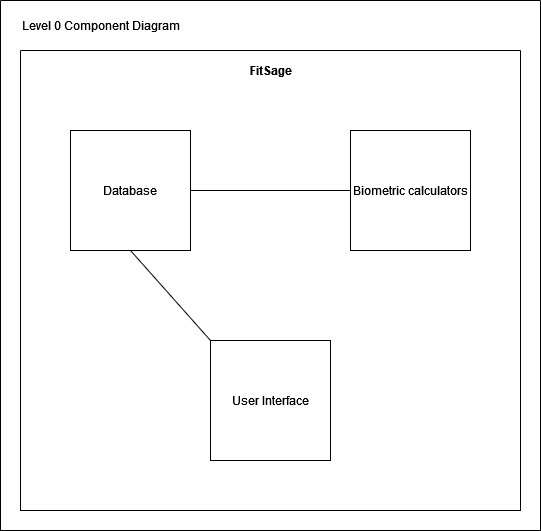
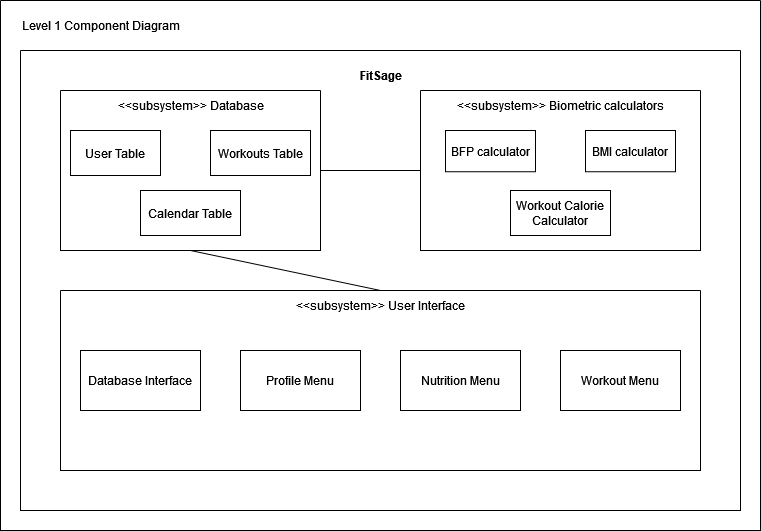
**UC5:** Track Progress

**Description:** A user wishes to see their progress towards their chosen goal. The user will have already uploaded their biometric information and selected a goal. The app will display the user’s daily calorie benchmark as well as their past history. The app will also display the user’s current streak and past streaks, if applicable. The information to be displayed will be accessed from the database.

**Actors:** User, Database

**Context Diagram:**



**Component Diagram:** 

**Component Diagram Descriptions:**

**Component:** BMI calculator

**Description:** Accesses the User Table in the Database for the user’s Age, Height, and Weight. Calculates the Body Mass Index (BMI) of the user and stores the value in the User Table of the Database.

**Component:** BFP Calculator

**Description:** Accesses the User Table in the Database for the user’s Age, Height, and Weight. Calculates the Body Fat Percentage (BFP) of the user and stores the value in the User Table of the Database.

**Component:** Workout Calorie Calculator

**Description:** Queries the Calendar, User, and Workouts Tables of the database to get information about each workout the user performed, such as the repetitions, amount of time the workout was performed for, workout name, MET value, and user’s weight. It then calculates the total calories that were burned from working out and subtracts the calculated value from the daily total calories before updating the Calendar table with the new daily total.

**Component:** Workouts Table

**Description:** Stores information about various workouts including the muscle group activated, workout name, difficulty, and MET value of the workout. This component is accessed by the Workout Menu to display information about relevant workouts to the user.

**Component:** User Table

**Description:** Stores information about the user including the user’s name, height, weight, age, BMI, BFP, goal, and goal calorie range. This component communicates with the BMI and BFP calculators as previously described. It also communicates with the Profile Menu to display and update the user’s information.

**Component:** Calendar Table

**Description:** Stores information about each day the user uses the app such as the date, total daily calories, workouts performed, number of times each workout was performed, how long each workout was performed for, and streak. It communicates with the Nutrition and Workout menus to update and access the daily total calories. It also communicates with the Workout Calorie Calculator to determine how many calories were burned from working out. It is also accessed by the Profile Menu to display the user’s current streak and their past streak history. Finally, it updates the age field of the User Table each year as the user ages.

**Component:** Database Interface

**Description:** Queries and updates the Database with data as the user uses the app. This component acts as the bridge between the Database and the rest of the User Interface.

**Component:** Profile Menu

**Description:** Allows the user to view and change information about their profile, including everything in the User Table of the Database and their current daily calorie total with streak and goal achievement information. Goal achievement is earned when the user’s daily calorie total is within the calorie range generated by the app for their selected goal.

**Component:** Nutrition Menu

**Description:** Scans Nutrition Facts labels from food the user consumes and updates the Calendar Table of the Database with the calories from the food.

**Component:** Workout Menu

**Description:** Allows the user to select the muscle groups they want to exercise and queries the Workouts Table of the Database for workouts matching the chosen muscle group. The user can sort the workouts by difficulty or intensity (MET value). Once a workout is chosen, a set of instructions will appear and the user will start exercising for a specified amount of repetitions or time. The workouts performed, repetitions, and how long each workout was performed is stored in the Calendar database. Then, the Workout Calorie Calculator is used to calculate the calories burned from working out and to update the daily total.

**Component Diagram Explanation:**

The three greater subsystems of Biometric Calculators, Database, and User Interface were chosen because each component is written in a different language and functions in a different way such that they comprise separate systems. While separate, the systems are designed to interface together to form a cohesive software application.

The Biometric Calculators access and store data in the Database, but do not communicate directly with the User Interface. Likewise, the User Interface accesses and stores data in the Database without communicating to the Biometric calculators. The User Interface will request the Biometric Calculators to calculate values, but no data will be sent from the UI to the Calculators. Instead, all relevant data will be stored in the Database, which acts as a bridge between different components and subsystems in the application.

The Biometric calculators are written in Java and perform three main functions, each represented as a separate component. The first calculator calculates the Body Mass Index (BMI) of the user. This value is used by the Workout Menu to determine which workouts to recommend to the user. The second calculator calculates the Body Fat Percentage (BFP) of the user. This value is used to determine the user’s body type, which is stored in the User Table of the Database and visible through the Profile Menu of the app. The BMI and BFP will change dynamically as the user makes progress towards their goal in the app. The final calculator is the Workout Calorie Calculator. This component will calculate the calories burned from working out given information about the workout from all three tables of the Database. Each calculator performs its own distinct function, which is why there is a component for each one.

The Database contains three tables, with each table storing information relevant to a different part of the app. For this reason, each table has been represented as a component in the Component Diagram. The User Table stores information about the user, such as their name, goal, and biometric information. The Workouts Table stores information about each workout, such as the name, associated muscle group, difficulty, and MET value. The Calendar Table stores information relevant to each day, such as the date, total daily calories, workouts performed, and streak. As you can see, each table serves a different purpose, and thus each table acts as a component in the application. The Database uses the SQLite database engine and SQL as the query language.

Finally, the User Interface (UI) serves as the means for the user to access all of the application’s functions. Each menu and interface contains actions the user can take to perform a specific set of functions that are mostly exclusive from each other. The only UI component that interacts with other UI components at a large scale is the Database Interface, which is used to query and update the Database as the user uses the app. The UI menus mostly display information to the user in an accessible format, and are focused on the appearance of the app. Because of this, most of the app’s actual functionality is delegated to the Database Interface, which serves as a boundary layer in between the UI and the Database. This separation allows the UI code to stay organized and design-focused. The UI is written in Dart using the Flutter framework for app development.