Software Design Description Detail

MyCampusGym

6.3 Detailed CSC and CSU Descriptions Section

The Computer Software Components (CSCs) of MyCampusGym include all of the primary feature modules of the application such as:

- MyFeed
- MyJournal
- MyPlan
- MyHealth
- MyProfile (titled “Profile & Settings”)
- Authentication
- MyFitness
- MyBuddies [to be implemented]
- MyMentor [to be implemented]
- MyRewind [to be implemented]
- Admin

MyFeed is comprised of the following Computer Software Units:

- MyFeedView
- FeedService

MyJournal is comprised of the following Computer Software Units:

- MyJournalView
- AddExerciseView
- Workout
- WorkoutDetailView
- WorkoutView
- JournalService

MyPlan is comprised of the following Computer Software Units:

- MyPlanView
- PlanService

MyHealth is comprised of the following Computer Software Units:

- MyHealthView

MyProfile is comprised of the following Computer Software Units:

- ProfileSettingsView
- ProfileView
- ProfileRowView
- ProfileService

Authentication is comprised of the following Computer Software Units:

- AuthViewModel
- LoginView
- ProfileView
- RegistrationView
- User

MyFitness is comprised of the following Computer Software Units:

- MyClassesView
- OccupancyView
- FacilityRowView
- MyFitness

Admin is comprised of the following Computer Software Units:

- AdminView
- AdminFacilityView
- AdminFacilityEditView
- AdminUserView

Each of these CSUs will be described in detail in the subsequent subsections of the document.

6.3.1 Detailed Class Descriptions Section

Each CSU is composed of a struct or class that performs a specific function in MyCampusGym.

MyFeedView

MyFeedView is responsible for rendering the Campus Feeds and providing navigation between the Campus Feed and Builder Feed using a Swift TabView.

FeedService

The FeedService is responsible for retrieving and saving feed messages to and from the firestore database. The class contains a method for creating a new feed conversation, creating a feed conversation, retrieving the messages of a conversation, and creating a new message in the conversation.

MyJournalView

MyJournalView is responsible for rendering the list of a user's workouts and contains a member array containing the respective workouts, an AuthViewModel instance as an environment variable, an instance of JournalService for managing journal data, and boolean to control showing the new workout sheet.

AddExerciseView

The add exercise view is responsible for rendering the exercise entry form on the Workout View sheet.

Workout

The workout struct contains the member variables that comprise a workout. These include the start and end date and time, the string values for breakfast, lunch, dinner, and snacks, as well as the array of exercise.

WorkoutDetailView

The workout detail view is responsible for rendering the form view of the workout and contains a member variable for the workout struct.

WorkoutView

The Workout View is responsible for rendering and displaying the individual rows representing a workout on the My Journal View. It contains a member boolean variable to determine if the user is creating a new workout, a binding variable member to contain the workout struct, a binding variable of the Journal Service interface, and an AuthViewModel instance environment object.

MyPlanView

The My Plan View is responsible for displaying the My Plan View information of a user. It contains member variables to contain the goal weight, the daily frequency, weekly frequency, and the array of booleans for workout days.

MyHealthView

The My Health View is responsible for displaying the user’s current weight, height and body mass index and renders the form for entering all three values. It contains a member variable for the feet, inches, body mass index, and weight of the user.

ProfileSettingsView

The Profile Settings View is responsible for rendering the various data points of a User’s profile and contains an environment variable member for the AuthViewModel instance.

Profile

The Profile struct contains the member data structure containing the member variable for the user’s id, birthday, campus name and ID, and user ID.

ProfileRowView

The Profile Row View is responsible for rendering a row for each data point in the profile struct on the Profile Settings View.

LoginView

The login view is responsible for rendering the login page of the MyCampusGym application, including the username field and the password field, as well as the login button and navigation link to register as a new user.

ProfileView

The Profile View is responsible for rendering a user's basic profile information including the users name and email and profile settings. It contains a member instance of the AuthViewModel to access the user's data.

RegistrationView

The registration view is responsible for rendering the registration page of the MyCampusGym application, including the email field, password field, name field, and create user button.

User

The User contains an instance of Firebase User data in the form of a member variable for the ID, display name, and email of a user.

MyClassesView

The my classes view is responsible for rendering a list of classes that are coming up for the user's campus.

OccupancyView

The occupancy view is responsible for rendering a list of the facilities that are available at the user's campus along with their current open state and the current occupancy level.

FacilityRowView

The facility row view is responsible for displaying the name of a facility, its open state, and the current occupancy level. This facility row view is used to as list items on the occupancy view.

MyFitness

The my fitness view is responsible for displaying options to the user to view my classes or the occupancy data for their campus.

AdminView

The admin view gives campus admins the ability to access their facilities for editing and their users for editing, including making them admins.

AdminFacilityView

The admin facility view is responsible for rendering a list of the campuses facilities for an admin to add to or select for editing.

AdminFacilityEditView

The admin facility edit view is responsible for rendering editable fields to the admin users so they can modify the details of a facility. This inclues the facility's name, its place id for the google maps api, as well as the postal code for the google api search.

AdminUserView

The admin user view is responsible for rendering a list of the users for an admin to review and select to update. This view displays all the users associated with a campus to an admin in alphabetical order by first name.

6.3.2 Detailed Interface Descriptions Section

JournalService

The Journal Service is responsible for interfacing between the Journal View and the Cloud Firestore Database. It contains a member variable for the Firestore Database instance, as well as a member function to retrieve the workout documents of a user as well as a one to create a workout for a user.

AuthViewModel

The AuthViewModel is responsible for interfacing between the Login View or Registration View, and the cloud Firebase Authentication application. It contains a method for creating a user in Firebase, for initiating a login for a Firebase Authentication user, for signing out a user session, and updating the name of a Firebase User.

Profile Service

The Profile Service is responsible for interfacing between the Profile View and the Cloud Firestore Database application. It contains a member function for retrieving a member document from the Firestore Database, as well as one for updating or creating a user's profile document in the Cloud Firestore Database.

Plan Service

The Plan Service is responsible for interfacing between the My Plan view and the Cloud Firestore Database application. It contains a member function for updating a user's plan document in the database, a member function for retrieving a user's plan document in the database, and a member function for creating a user's initial plan document in the database.

Facility Service
The Facility Service is responsible for interfacing between the MyFitness and the Cloud Firestore Database application, as well as between AdminFacilityEditView and the Cloud Firestore Database application. It contains a number of member function for updating a Facility

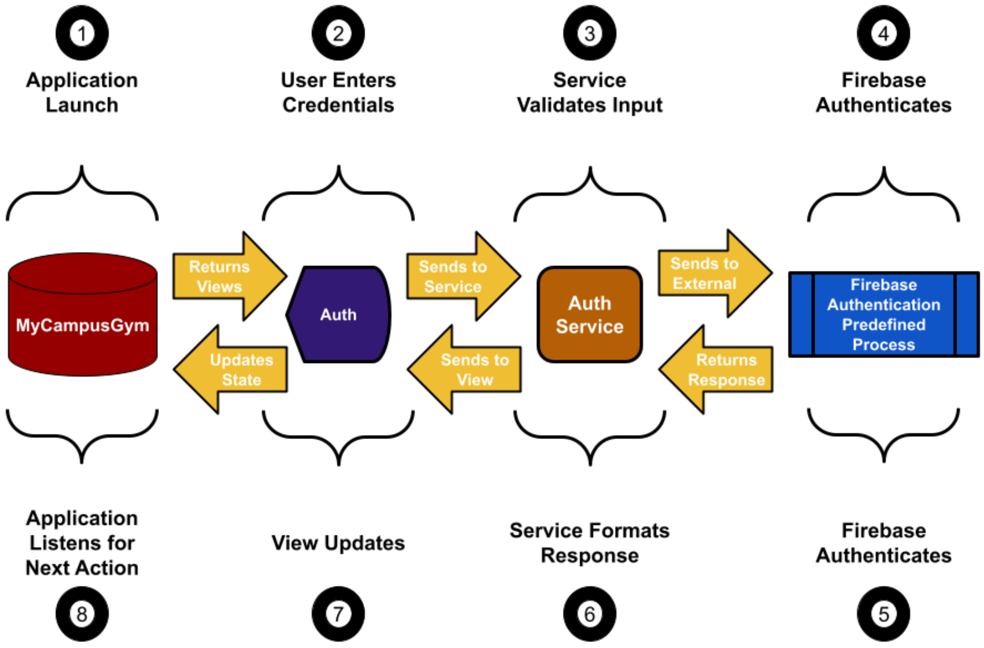
as well as retrieving one facility document or a list of facility documents, as well as one to create a new facility.

6.3.3 Detailed Data Structure Descriptions Section

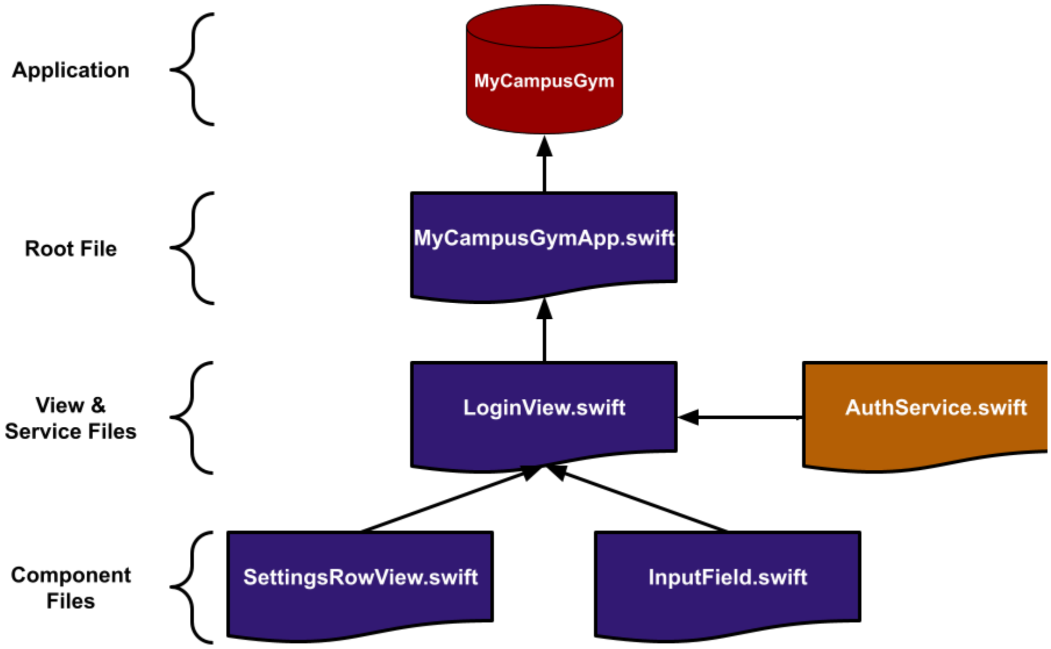
As MyCampusGym is a SwiftUI project, all data structures are composed in the context of the computer software unit classes described in Section 6.3.1 above.

6.3.4 Detailed Design Diagrams Section

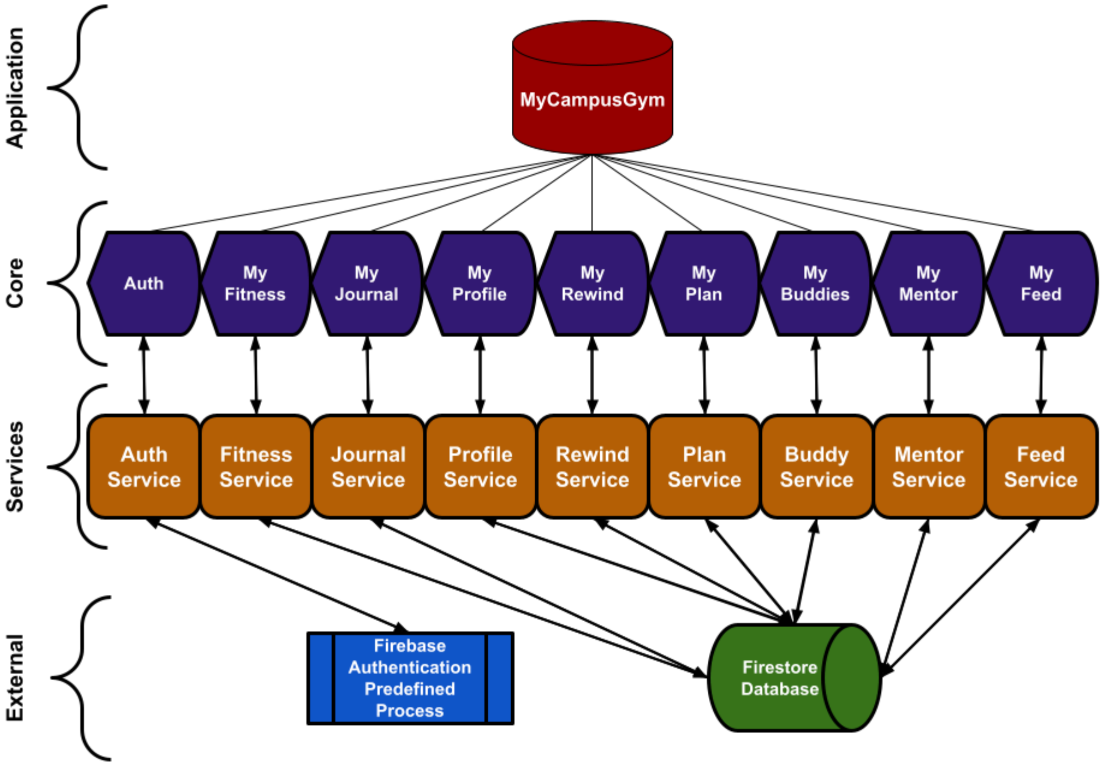
Data Flow Diagram



Implementation Diagram



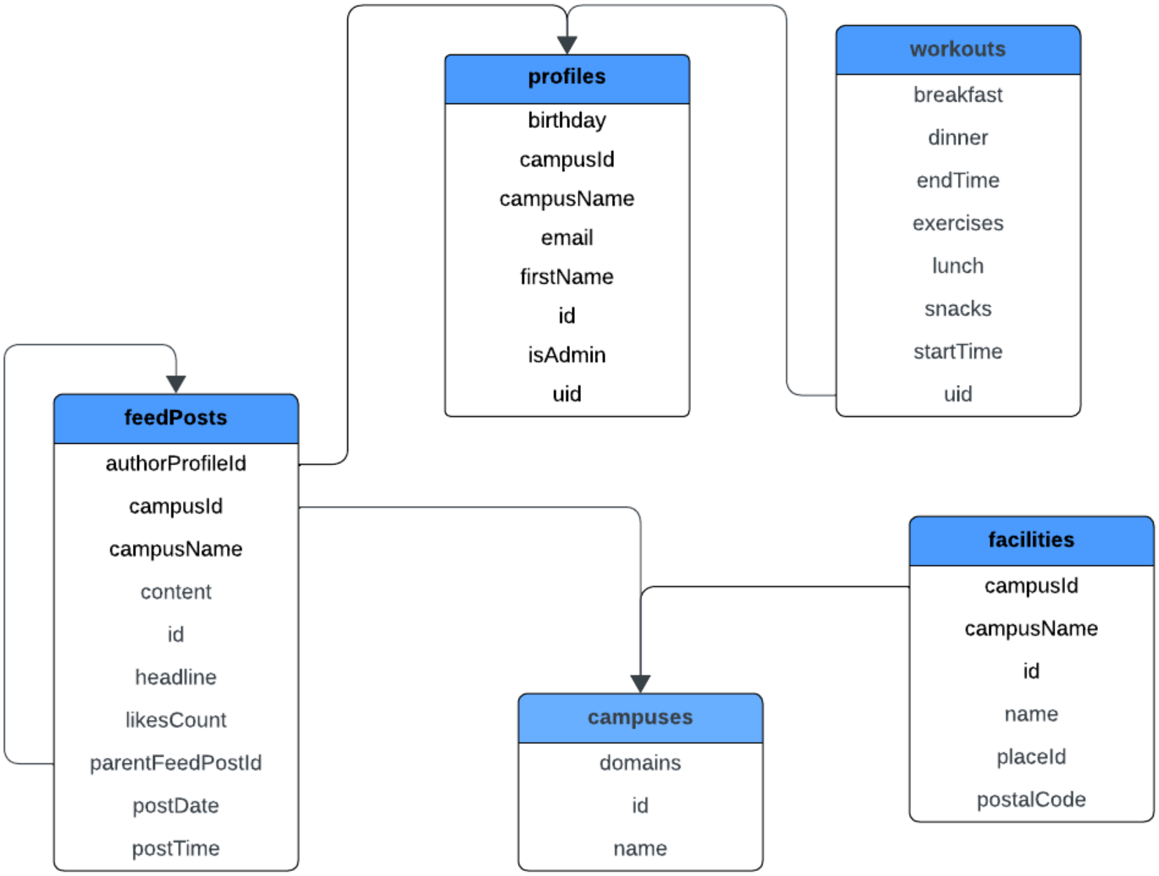
Top-Level Diagram



6.4 Database Design and Description Section

Firestore database collections are based on each feature of MyCampusGym. Each document in collections related to user data contain an Id field to reference the related User’s profile. The Profile collection contains additional User data not handled natively by the Firebase Authentication User.

6.4.1 Database Design ER Diagram Section



6.4.2 Database Access Section

Firestore Database collections are access by dedicated Service classes, respectively created for each Collection. See Section 6.3.2 for information regarding the interface of the Views with the Database.

6.4.3 Database Security Section

Database Security is handled by native Firestore Database Security protocols. MyCampusGym is configured to require an authenticated query or change prior to granting access to the database. In addition, Edit Permissions are restricted to users with a Profile that is an Admin and Read Permission are granted to users who have a matched Campus in their profile. This ensures that bad actors are unable to modify database data and restricts any data from other campuses to be visible.