A Software Design Specification for the Fitness Paradise App (FPA)

1. Purpose

This *Software Design Specification* is made to provide a blueprint of the software design of the FPA. This document provides a textual description to help developers to get a deeper insight into the structure of the software.

2. Intended audience

This document is directed towards the development team and the customer (i.e., the fitness paradise center's owner).

3. System Overview

The FPA is a software that facilitates the booking and use of the facilities of the fitness paradise center. Also, the FPA allows its members to visualize information on the booked facilities and activities, and to share performance data with other members.

4. System Design

The software of the app is designed according to the Model-View Controller (MVC) design pattern to have a: (i) faster development process: MVC supports rapid and parallel development, (ii) ability to provide multiple views: in the MVC Model, you can create multiple views for a model. (iii) better support for maintenance: modification does not affect the entire model because model part does not depend on the views part.

The Model is the central component of the pattern. It is the application's dynamic data structure, independent of the user interface. It directly manages the data, logic and rules of the application. The Model part consists of several entities:

- **Member**: can be a club or an individual. A member has information on user account, email and the list of bookings that he/she did. The member can do bookings and exchange performance data. One Member can do zero or many Bookings. One Member owns zero or many FitnessBands. The Member entity is used by the PerformanceTracker.
- **FitnessBand:** provides information on the performance data of the member. One FitnessBand belongs to one Member.
- **Booking:** provides information on the booking number and the payment method. One Booking is done by one Member. One Booking has zero or one Invoice. The Booking entity is used by the Booking Controller.
- **Invoice:** provides information on the invoice number. One Invoice is related to one Booking.
- **Facility:** provides information on the list of activities conducted in one facility. One Facility can host one or many Activities. The facility can be a football pitch, tennis court, swimming pool or gym. In the gym, there is at least one machine. The machine can be a: weight, rowing or cycling machine. The Facility entity is used by the Booking Controller.

- **Activity:** can be training, yoga classes, physiotherapy sessions, or massage sessions. One Activity has one Schedule. One Activity is conducted in one Facility. The Activity entity is used by the Booking Controller.
- **Schedule:** provides information on the schedule of the activities. One Schedule belongs to one Activity.

The Controller part consists of three controllers that accepts input and converts it to commands for the model or view:

- **BookingController:** this controller uses the data from the Booking, Facility and Activity entities. This controller updates the Booking View. The functionality of this controller is to change the booking, cancel the booking, get a facility, get an activity, manage the payments, and send the invoices.
- **ActivityController:** this controller uses the data from the Activity entity. This controller updates the Activity View. The functionality of this controller is to manage the activities, update the schedule of the activities, and get the facilities where the activities are taking place.
- **PerformanceTracker:** this controller uses the data from the Member entity. This controller updates the Performance View. The functionality of this controller is to manage performance data of the member.

The view can be any output representation of information. The view part includes three views that each implement a *ViewComponent* interface:

- **BookingView:** this view provides options to visualize the booking, visualize the payment and visualize the member.
- **ActivityView:** this view provides option to visualize the activity, visualize the facility and visualize the schedule of the activities.
- **PerformanceView:** this view provides option to visualize the covered distance and time, the burned calories, but also aggregates over a week and month.