# **CSC 431** Happy fitting System Architecture Specification (SAS)

**Team 02**

|  |  |
| --- | --- |
| Zhenyang Guo | Documenter |
| Zipei Chen | Coding |
| Yi Rong | Market Analyzing |
|  |  |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1 | 4.1 | The whole team | First draft |
| 2 | 4.13 | The whole team | Second draft |
| 2.1 | 4.15 | The whole team | Revised second draft |
| 3 | 5.5 | The whole team | Final draft |

# Table of Contents

[1. System Analysis 5](#_Toc412746586)

[1.1 System Overview 5](#_Toc412746587)

[1.2 System Diagram 5](#_Toc412746588)

[1.3 Actor Identification 5](#_Toc412746589)

[1.4 Design Rationale 6](#_Toc412746590)

[1.4.1 Architectural Style 6](#_Toc412746591)

[1.4.2 Design Pattern(s) 6](#_Toc412746592)

[1.4.3 Framework 6](#_Toc412746593)

[2. Functional Design 7](#_Toc412746594)

[2.1 System diagram 7](#_Toc412746595)

[3. Structural Design 8](#_Toc412746596)

# List of Figures

1. System diagram 5
2. Sequence diagram 7
3. Happy fitting class diagram 9

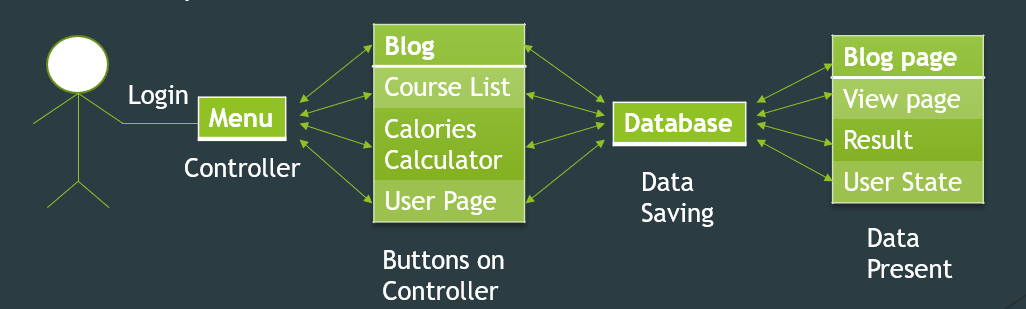
### System Analysis

#### System Overview

The System will be comprised of four main parts: menu, courseList, Blog system, and user page. The menu contains two services which is the courseList and Blog. Users could choose the according services in the Menu. The courseList page would offer user a list of fitting course videos stored in server. Blog system and user page will be directly connected to the database. Data will be stored in the database and sorted / retrieved according to the user’s needs. The database will store data via firebase using no-SQL database format.

This pipeline for courselist is as follows: (1) the user submits courselist opening request which opens the courselist service on the menu. (2) the user start a course and the timer begin to work. (3)the user close the course then the exercise time of the user will upload to the database (4) the server would calculate the calorie lost based on the exercise time and course information (5) the database will add the exercise time to the time the user already exercised, calorie lost to the total calorie lost and feedback the total exercise time and calorie lost of the user.

#### System Diagram



#### Actor Identification

There are three type of human Users. First is user has account created, Second is user doesn’t have account created, Third is administrator. For users with their accounts created, they would be able to use all functions this app provides. However, for users don’t have accounts, they would not be able to use these functions and would be asked to create new accounts. Administrator would be able to use all of these functions without the login process , delete illegal topics on the blog, add or delete videos on Server. This system would not support non-human actors.

#### Design Rationale

##### Architectural Style

We use Model-View-Controller as our architecture style. Model-View-Controller is the architecture style which has three layers. One is where you have the data, in our product, is our database. The other one is where the users actually view the data, in our product, is the page to show the courses. Final is the middle layer which has the ability to communicate the one layer with the other. In our product, is the client-based tier, it works as a controller that allows the users to easily get the data they want.

##### Design Pattern(s)

In our product, the most important design pattern is Factory Method. The advantage of using factory method is that it can easily create a lot of objects. In order to increase the competitiveness of our app in the market, we need to add new courses into our app very often. So that’s reason why we choose Factory Method for our system.

##### Framework

The web application will run with Responsive Web because it’s lightweight and efficient.

### Functional Design

#### Sequence Diagram

A screenshot of a computer

Description automatically generated with low confidence

* When the user opens the application, the login screen opens.
* After submitting the login page, the login is verified, and user data is loaded if the login was a success.
  + If the login was not a success, login is denied.
* The menu screen is then opened.
* Two options would be available on the menu screen.
* First is the Course List, Once the Course List is selected, the course selecting screen is opened.
* Once a course is selected, the course screen is opened. At the same time, the timer starts working.
* Once close the course, the exercise time of the user will upload to the database
* the server would calculate the calorie lost based on the exercise time and course information
* the database will add the exercise time to the time the user already exercised, calorie lost to the total calorie lost and feedback the total exercise time and calorie lost of the user.
* The second option is the Blog.
* Once a topic is created on the blog, the according information would be stored in the database.
* Once a topic title is being searched, the Server would sort and retrieve according topic in the database
  + If the following topic title is founded, the database would return a list of finding result.
  + Else, a failure message would be represented.
* At the end of use, the application is closed (automatic logoff).

### Structural Design

Happy fitting class diagram

Timeline

Description automatically generated