

# Requirements Analysis and Design

## Requirements analysis

### 1. Target

This application is mainly an on-line tennis training management system to provide a platform to the tennis training center, the instructors and the players. Its main function is to assist the instructors to design a scientific, reasonable and traceable training plan for the players and to evaluate the performance of the players periodically. Furthermore, the players can use the system to join into the tennis courses they are interested in and learn their performance during the period of training by using the feedback interface. Lastly, the training center can also extract the new users to join into the tennis classes it holds.

### 2. Business Content

According to the target of the tennis training system, there are 4 subsystems need to build.

#### (1) General User Management (GUM)

The GUM is to handle the common requests from both instructors and players such as login, register and so on.

#### (2) Training Plan Management(TPM)

The TPM is aim to mange the training plan life cycle from creating by instructors to publishing to the players. It should provide an editing interface to the instructor and a viewing interface to the players.

#### (3) Performance Evaluation Management(PEM)

The PEM is to provide an interface for both instructors and players to trace the player's performance periodically and let the players to learn how they performed and which part of skills they lack of. Also the instructors can use PEM to assist them to make a more reasonable and targeted training plan.

#### (4) Course Management(CM)

The CM is aim to manage the relevant training course information. It will provide an

function to training center to push their course information on-line, to assign the instructors to be responsible for. Further, the players can join some of the courses by using the system. The CM also provide the course reminder function to remind the instructors and the player the time and place of the future course they will attend.

## Features Analysis

### 1. The entrance of the system

Considering the user experience and the layout of the main page, we plan to apply “one entrance for different users” principle, which is that the instructors and the players can use a same interface to complete the register and login request. The back end will forward them to different pages according the their role from database.

Use case: Register

User	Instructors and Players
Work flow	When a user entered the system main page, it can click the “register” button at the navigation bar. The system will show it a page to require it select whether it is an instructor or a player, and to input its information such as username, password and email and so on. After clicking “submit”, a confirm email will be sent into the address the user input in to register page.
Preconditions	-
Result	After the user click the link in the email. The process of register will be completed and it will be able to login.

### 2. Personal Information Editing

No matter for the players or for the instructors, a full-sided personal information is important to let others to learn themselves. However, it is not realistic to require the user to input all information at the register phase. Thus, the system provide a function of editing personal information after the user login into the system. For the instructors, they upload their profile photo, the teaching experience and the reward certifications and so on. For the players they also can update their information such as adding their reward titles and their birthday and so on.

Use case: Edit user information

User	Instructors and students
Work flow	User click on the “Profile” button, then the controller will show him or her a new page that the user can update and modify their personal details such as uploading a profile photo or honor certificates. For the instructors , they also can update their qualification. When they click the “confirm” button, the system will store their information into the database and return them back to the trainer page or non-trainer page according to who are they.
Preconditions	User need to login into the system.
Result	The updated information of the users will be stored into the database, and will be displayed into the relevant pages. For example, the instructors’ will be contained into the training class information, when the students or their parent check the information of the their class and they will also can see the information of the instructors.

### 3. Main Functions For Instructors

After the instructors logged into the system, the system will forward them into their home page, which has different functions and layout with the players’ home page. The system will provide several functions to the instructors: displaying the class information, viewing the detail of the players , make new training plan, training evaluation and course timetable.

#### (1) Display the class information

The system will show the instructor a table of the classes that he or she is responsible for. And the instructor can view the detail of each class by clicking the title of each class.

Use case: Class information function for instructors

User	Instructors
Work flow	Each class name or title will be shown as a row of a table or in the format of slides. Clicking each class, the system will pop up a window containing the students list, historical training plan, making a new plan
Preconditions	User need to login into the system.
Result	This function is a intermediate link of a further movement and a displaying of the classes which the instructors are responsible for.

## **(2) Viewing the detail of the players**

After the instructor click a class title in the class information table. The detail information will be shown in a new page containing a list of the students of this class. The instructor can view a specific student information by click its name.

Use case: Student details for instructors

User	Instructors
Work flow	the system will display the students list of this class to the user. The user can also click the each student, then the system will show the detail information of the this student from the database including personal details, historical evaluation results and so on.
Preconditions	the user click the button or link of each class at the class information section
Result	This part is mainly to provide a interface to instructors to learn more about their students.

## **(3) Training performance evaluation**

Also at the student list table, there is a button “evaluation” at the end of each row for each student. The instructor can click the button when he or she want to assess the student performance.

Use case: Training performance evaluation

User	instructors
Work flow	the system will provide a format of performance evaluation to the instructor according to the tables in database. The instructor can choose what type of skills he or she want to assess for this student. The system will also provide some selection button to the instructor to record the counts that the student achieve during the assessing time. Also the instructor can type the comments into the evaluation table. After the instructor submit, a new performance evaluation report of this student will be created by the system
Preconditions	the user click the “evaluate” button
Result	the performance evaluation data will be stored into the database. And the student or their parents and the instructor can view the student’s performance data in the relevant pages.

## **(4) Make training plan**

After the instructor click a class title in (1) the Display the class information part, at the end of the page, there is a button “make a new plan”, the instructor can click it to

make a new training plan.

Use case: Make training plan

User	Instructors
Work flow	After the user click the “new plan” button, the system will draw out the training plan structure and the training types from database and show them to the user. The system will require the user to type the training time and place and allow him or her to select the types of training.
Preconditions	the user click the button or link of each class at the class information section and click the “new plan” button
Result	After the user click the “submit” button. The system will create a new training schema and store it into the database. Meanwhile, the time table and historical training plan sections for the instructor and each students in this class will be updated.

## (5) Timetable

The timetable will be shown at the home page of the instructor below the class information section. The instructor can view the future courses information such as training time, place and so on.

Use case: Timetable

User	Instructors
Work flow	The time table of the courses that the instructors will attend will be shown below the class information section. At the home page of the user, the time table will only display the most recent one or two courses, and the user can click the “more” link to see the more coming courses in the future. Also, when click each course, the training plan, the students, the place and the time will be shown in detail.
Preconditions	User need to login into the system.
Result	The instructors will see the coming courses that they need to attend. Also, the system has a reminder function to remind them the relevant information.

## 4. Main Functions for Players

After the players logged into the system, the system will forward them into their home page. The system will provide several functions to the players: displaying the class and course information, viewing the private training performance data, view training

plans. However, most of these functions are read-only.

### **(1) Class and course information function for students**

User	Students
Work flow	After the user login into the system, the class and course information will be shown below the navigation bar. The student can choose the course that they attended or will attend to see the details about this course or class such as training time and place, the instructor's information, the historical training plan.
Preconditions	User need to login into the system and has attend some course or class
Result	For the student section of the system, most of them are ready-only.

### **(2) Private training performance data**

User	Students
Work flow	The student can view his or her historical performance evaluation information including the performance data in the form of diagrams and the comments from the instructors.
Preconditions	User need to login into the system and has attend some course or class
Result	For the student section of the system, most of them are ready-only.

### **(3) Training plan viewing and the reminder**

User	Students
Work flow	The time table and the new training plan of the courses and class that the student will attend will be shown below Private training performance data section. Also, the system has a reminder function to highlight the most recent courses
Preconditions	User need to login into the system and has attend some course or class
Result	For the student section of the system, most of them are ready-only.

## **5. Course Join and Management**

The system will provide the interfaces for both the new players or the players who want to join into a different course and the training center to open a new training class or update the information of the training courses.

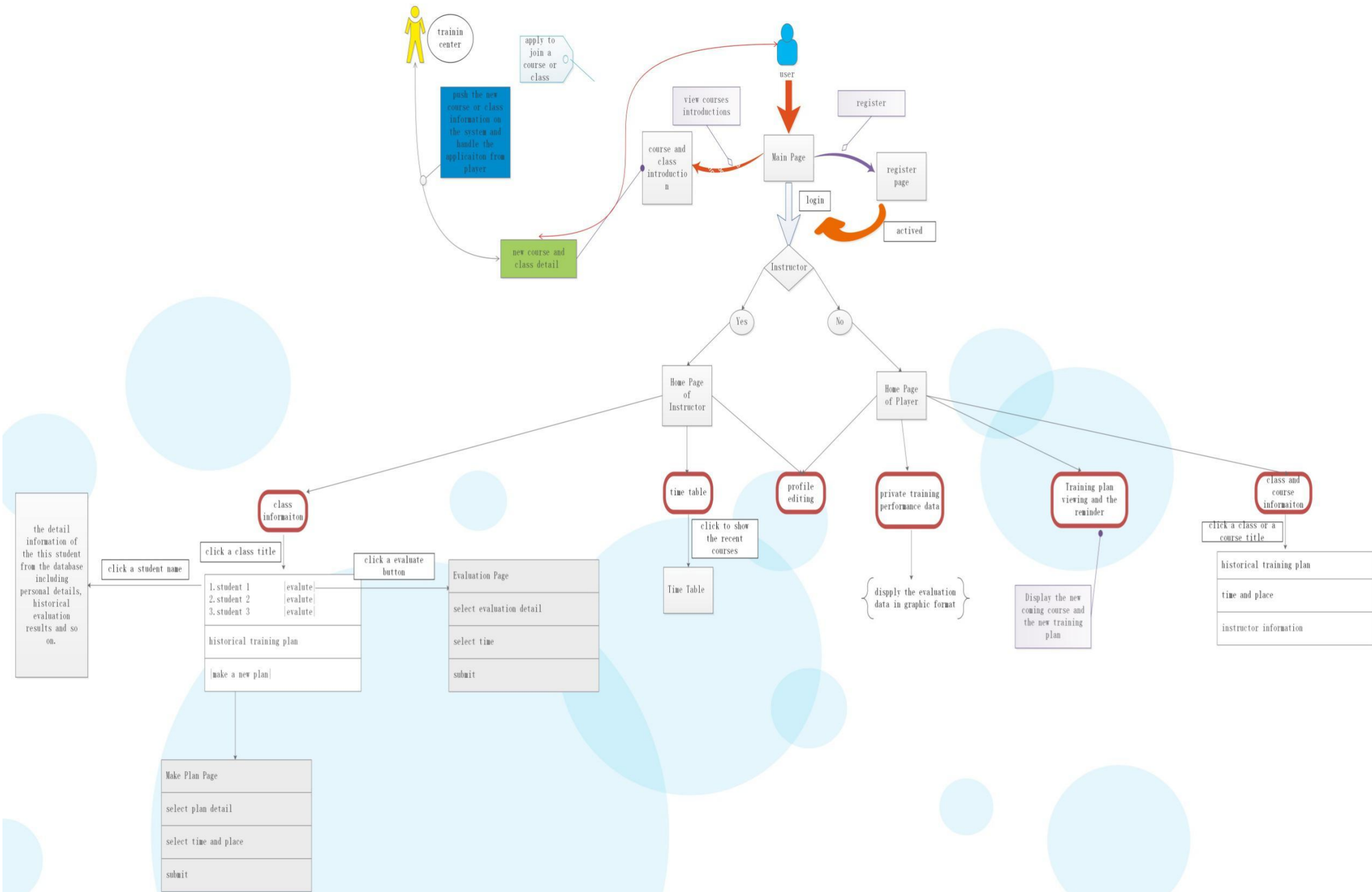
## 1. Class and Course management

User	Training center
Work flow	The system will provide a platform to the training center so that it can update the training courses. The administrator of the training center can open a new courses by click the “open new course” button, and select the instructor and type the course detail into the relevant forms.
Preconditions	the username and password to login has to be authorized.
Result	After submit , the system will create a new course into database and the course information will be display into the main page of the system so that the user can join into the new course if they are interested in.

## 2. Join a course

User	Players
Work flow	At the main page of the system, the players can view the course introduction provided by the training center. After they click the “join” button, the request will be passed by the system to the training center.
Preconditions	To view the course instruction at the main page, the user can remain logout state, but if the user click the “join” button, the system will require it to login.
Result	After the training center approved, the player can view the relevant detail of the course including training plan, training time and place at its home page.

**Whole Picture for the work flow(see next page)**





# System Design

## 1. Front End

### 1.1 Developing Environment

(1) Language: HTML5+Less+JavaScript

(2) Version:

- a) AntD: 3.8.2
- b) Axio: 0.18.0
- c) Babel-loader: 7.1.2
- d) React: 16.4.2
- e) Redux: 4.0.0
- f) React-route-dom: 4.3.1
- g) Webpack: 3.1.8
- h) Less: 2.7.3

### 1.2 Developing Concept

#### 1.2.1 Design Mode

According to the thinking pattern of React, front-end team use components to separate some functional sections, interact parameters among those components. Based on user experiences, build delicate user interfaces and render the web pages after receiving relevant data from the back-end.

#### 1.2.2 Structure Mode

We use React to build encapsulated components that manage their own state, then compose them to make complex UIs. To make the entire data-flow much clear, we also apply Redux to store the state of each components and also dispatch some actions to update state. Since components are the heart of React's powerful, declarative programming model. We use React Router as a collection of navigational components that compose declaratively with our application.

## 1.3 Function Module

### (1) Welcome Page (WP)

**Functions:** display main functions of project

**Developing flow:**

- 1. Design the user interfaces of this page;
- 2. Using React-route to navigate the link to other page;

### (2) Log in Page (LIP)

**Functions:** handle log in function of users

**Developing flow:**

1. Collect the value of log in form;
2. Use Redux to dispatch the log in action;
3. Use axios to fetch the log in message from back-end with form value;
4. Deal with the error message if exists;

**(3) User Main Page (UMP)**

**Functions:** display the main page of users

**Developing flow:**

1. Judge the user type based on the data from back-end;
2. Build Student page which contains their course information and their persona ability data;
3. Build Trainer page which contains their courses information;

## **2. Back End**

### **2.1 Developing Environment**

(3) Language: java

(4) Version:

- a) Jdk 1..8
- b) Maven 3.5.2
- c) Spring 4.1.3
- d) Mybatis 3.2.8
- e) Tomcat 7

### **2.2 Developing Concept**

#### **2.2.1 Design Mode**

Apply the MVC mode as developing principle including using springmvc framework as the controller to handle the request and the response, using the relevant java class as the model to transmit, modify and store the data, and using the front pages as the view to interact with the users.

#### **2.2.2 Structure Mode**

Since using java to develop the system of back end, we apply the three structure mode to build the system. First, springmvc is responsible for interacting with the pages and invoke other methods of the relevant java class. Second, the spring framework is in charge of managing the javabeans and transactions. Lastly, Mybatis framework is responsible for interacting with database.

### **2.3 Function Module**

**(4) General User Management (GUM)**

**Functions:** login, register, active the account, edit the personal profile

**Developing flow:**

1. Create the User POJO and mapping it into the user table;
2. Create the “UserController”, complete the relevant methods;
3. Complete the methods of the UserService and UserMapper, which the Controller needs

**(5) Training Plan Management(TPM)**

**Functions:** query historical plan, make a new training plan

**Developing flow:**

1. Create the training plan POJO and mapping it into the relevant tables;
2. Create the “PlanController”, complete the relevant methods;
3. Complete the methods of the PlanService, PlanMapper, CourseMapper, and so on, which the Controller needs

**(6) Performance Evaluation Management(PEM)**

**Functions:** query historical performance data of a student, make a new evaluation

**Developing flow:**

1. Create the relevant evaluation POJOs and mapping them into the relevant tables;
2. Create the “EvaluationController”, complete the relevant methods;
3. Complete the methods of the EvaluationService, UserMapper, EvaluationMapper, and so on, which the Controller needs

**(7) Course Management(CM)**

**Functions:** display the course introduction, detail and provide an interface to the training center to open a new course. Handle the student application to join a specific course or a class.

**Developing flow:**

1. Create the relevant course POJOs and mapping them into the relevant tables;
2. Create the “CourseController”, complete the relevant methods;
3. Complete the methods of the CourseService, UserMapper, CourseMapper, and so on, which the Controller needs