[Capstone Project]

|  |  |
| --- | --- |
| **Reporting Date** | June 9, 2019 |
| **Team members** | 20145271 금강현  20144367 이동현  20133950 공찬형 |

**Capstone Project Final Report**

**Life Sports Total Solution**

Index

[1. Project Overview 3](#_Toc11021092)

[**1.1** **Project Topic Selection** 3](#_Toc11021093)

[**1.2. Main Features** 4](#_Toc11021094)

[**1.3 Direction of Implementation** 7](#_Toc11021095)

[**1.4 Project Schedule** 8](#_Toc11021096)

[2. Web UI : UI for Administrator 9](#_Toc11021097)

[**2.1 Purpose** 9](#_Toc11021098)

[**2.2 Functionality** 9](#_Toc11021099)

[**2.3 Development Environment** 11](#_Toc11021100)

[**2.4 Implementation Details** 11](#_Toc11021101)

[3. App UI : UI for Users 13](#_Toc11021102)

[**3.1 Application Server** 13](#_Toc11021103)

[**3.2 Application UI** 13](#_Toc11021104)

[**3.3 Function** 14](#_Toc11021105)

[4. DB and Data Analysis 17](#_Toc11021106)

[**4.1 DB Structure** 17](#_Toc11021107)

[**4.2 DB Data Generation** 23](#_Toc11021108)

[**4.3 Data Analysis** 25](#_Toc11021109)

# Project Overview

**<Git Address>** [**https://github.com/cau-capstone-team11**](https://github.com/cau-capstone-team11)

## **Project Topic Selection**

1. **Project Topic**

The project's theme, Lifesports total solution, provides both users and administrators with a holistic solution for life and sports. For the user, the overall services related to the game, such as the process to be preceded by the competition, such as joining the team, checking the information on the sports facilities that can be played, and registering the schedule, It provides a comprehensive range of services to help manage gyms such as schedule registration and tournament registration.

1. **Background for Selecting Project Topic**

Experimental factors and social factors influenced the selection of project themes. The reasons for each are as follows.

1. Experimental Factors

Suppose you book a real football game. Unless there is a club that regularly participates, it is almost impossible to play a game. It will be a challenge to find and book a playable gym. You can call the gym that has a football stadium directly or check the availability of a reservation on the date and time by accessing the gym homepage. Fortunately, it is not easy to recruit a minimum of 22 people for the game, due to the nature of the soccer game, even if you have reserved a suitable time to find a playable gym. Therefore, it is necessary to recruit opponents who can participate in the match with high frequency, Even if a new person is sought according to the number of people, the team balance should be adjusted as the game progresses because the player does not know his / her ability. I simply want to play soccer with my friends in the spare time zone, but it is not easy to play a soccer game that is satisfactory in the present system because we need the complicated process as described above and do not know the skill of the people who participated in the game. This is not limited to soccer, but is applied to various ball games such as basketball and baseball.

1. Social Factors

According to KOSIS's survey data on participation rate of living sports in Korea, the number of people participating in living sports is steadily increasing for all ages. As of 2015, more than 40% of the population of all ages are enjoying sport for everyday life, and related infrastructure is needed as the rate of participation in daily sports increases. On the other hand, the most frequent respondents answered 'Lack of time (47%)', 'Health problems (15%)', 'Lack of interest in physical activity' (9%), ‘Low accessibility to sports facilities (6%)’, and ‘Feeling a burden on expense (5%).’ 17% of the factors directly influenced the lack of infrastructure related to daily life sports (lack of access to physical facilities, cost burden, lack of relevant information, lack of peers to enjoy sports, lack of PE program) (Due to lack of time) reached 47%.

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

As we can see from the above two factors, due to lack of platform related to sports for the living, people feel uncomfortable and the demand for them is sufficient due to the steady increase in the physical population.

## **1.2. Main Features**

Through the Total Sports Solution for Daily Life, administrators can easily manage gym facility and schedule through the web, and users can provide various functions and information necessary for the progress of the game through the app. To summarize the services provided through the platform in two sentences :

1. **Just focus on the game itself**

By providing various functions necessary for the progress of the game through the platform, administrators can make easier facilities and schedule management, and users can concentrate on the game itself. The detailed functions implemented for this purpose are as follows.

1. Divide the types of schedule in detail

For a user enjoying the game, the type of game to be enjoyed may vary depending on the user's circumstances. For example, if you have a team that enjoys a game on a given day or time, you may need a "home stadium". You should be able to schedule certain sports facilities on regular days and times of the day and be able to play regularly. On the other hand, users who do not play regularly but who want to participate in the game lightly on weekends or during off-hours may be more likely to open-match to recruit a large number of participants.

  As such, it provides three types of schedules to meet the needs of various users. 'Regular booking' which is a regular reservation of team unit, 'open matching' where various users who do not belong to the same team can participate in the game, 'tournament' schedule which enables the gymnasium prize money and gymnasium occupation do. The user can easily register the schedule according to his / her circumstances through the application. From the manager's viewpoint, the manager can notify the schedule registered through the web through the application, and the utilization rate of the gym can be increased through various schedule registration.

1. Manager : Facilities and schedule management

Administrators managing facilities at each gym are provided with a web UI for more convenient facilities and schedule management. It is possible to manage the facilities belonging to the gymnasium, and to register and manage the above-mentioned three schedule types and closed days for each facility. The registered schedule is displayed to the user through the app, and the user's reservation status can be confirmed through the web.

1. Users : Reserve matches

Users can check and register the facilities registered at each gym through the app. The reservation is made by selecting the items to be booked, the gym, and the schedule in order. Especially, in the case of "regular reservations," the whole team can use all the gyms, but by activating the "intercept" option, they can play against other teams. Regular reservations and tournaments are booked on a team basis, and open matches are booked on an individual basis.

1. Users : Match records for Team and Individuals

The application provides users with full information about the individual and the team. In total, you will see the number of matches played by teams and individuals to date, the odds of the last 10 games, and the latest game scores.

1. Users : Recommendation System

Three recommended features are provided for smooth progress. First of all, except for Open Matching, most matches are done on a team basis. Therefore, the team recommends teams based on the placement test or existing score, distance and age for the first-time user or no-team user. Also, if you belong to a specific team, it recommends an opponent team for regular bookings. Likewise, you will recommend the right team for your match based on your team's total, distance and age. Finally, in the case of open matching, we do not know the team's ability to play the match, so we divide the participants into open teams based on the existing open matching. Through this, participants with various skills can be placed evenly on both teams, enabling a balanced game progression.

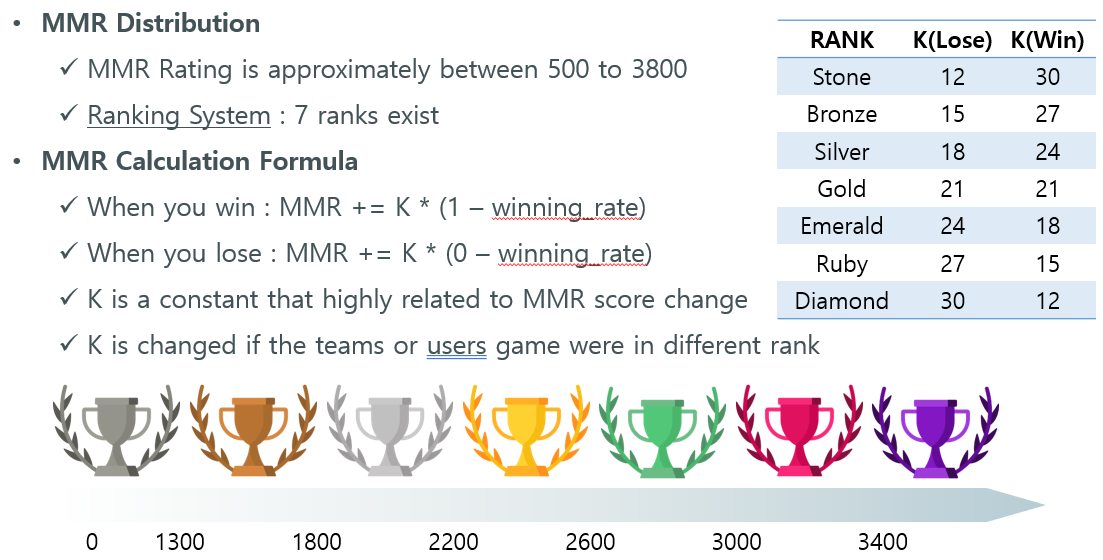
1. **Play the game like ‘a Game.’**

Add ‘gaming elements’ to the game to make it easier for the user to use the platform and promote the desire to participate in the game. The following functions are provided for this purpose.

1. Ranking System

The MMR system is introduced based on the users' individual and team totals for each event to quantify the user's ability. We determine team MMRs for each event using team matching through regular reservations for each sport (soccer, baseball, basketball, and badminton) provided by the platform, and determine the individual MMR by using the results of open matching conducted by the user. Build. Placement tests are given to users and teams who do not have a match record for a particular sport. The initial score is set at 2000 points, and the score variation according to the winner is increased so that the team / user can have the appropriate MMR score.

There are 7 'tiers' according to the score interval of MMR, and each tier' s name and score change width are as shown below. By using the odds ratio roughly proportional to the MMR score and the constant K applied to each tier, ultimately the overall MMR distribution can follow the Gaussian distribution based on the average score of 2000 points.



1. Badge / Achievement System

Through the introduction of the achievement system, the user is encouraged to participate in the game and provides additional information to indirectly understand the competence of the matching opponent in tournaments or regular bookings. Similarly, in open matching, additional information is provided to identify individual users' personal information assigned to the same team or opponent team. You can achieve achievements based on the number of matches played by the current user, the odds of the last 10 games, and the level of MVP points.

1. MVP

The participant will select the MVP of the match by voting. When selected as an MVP, MVP points will be accumulated for each individual event, which is used to achieve achievement and provides additional information to indirectly understand the individual's ability.

## **1.3 Direction of Implementation**

As mentioned earlier, the web UI for managing facilities and facilities in gymnasium, app UI for increasing user convenience and motivation to participate, storing data generated at the time of reservation, and recommending analysis based on collected data We will implement DB and Hadoop clusters to provide. The functions implemented in each field are as follows.

1. **Web UI**

Provides a UI for administrators. Register new facilities for managed gymnasiums, manage scheduled reservations based on registered facilities, organize tournaments, and recruit users and teams.

1. **App UI**

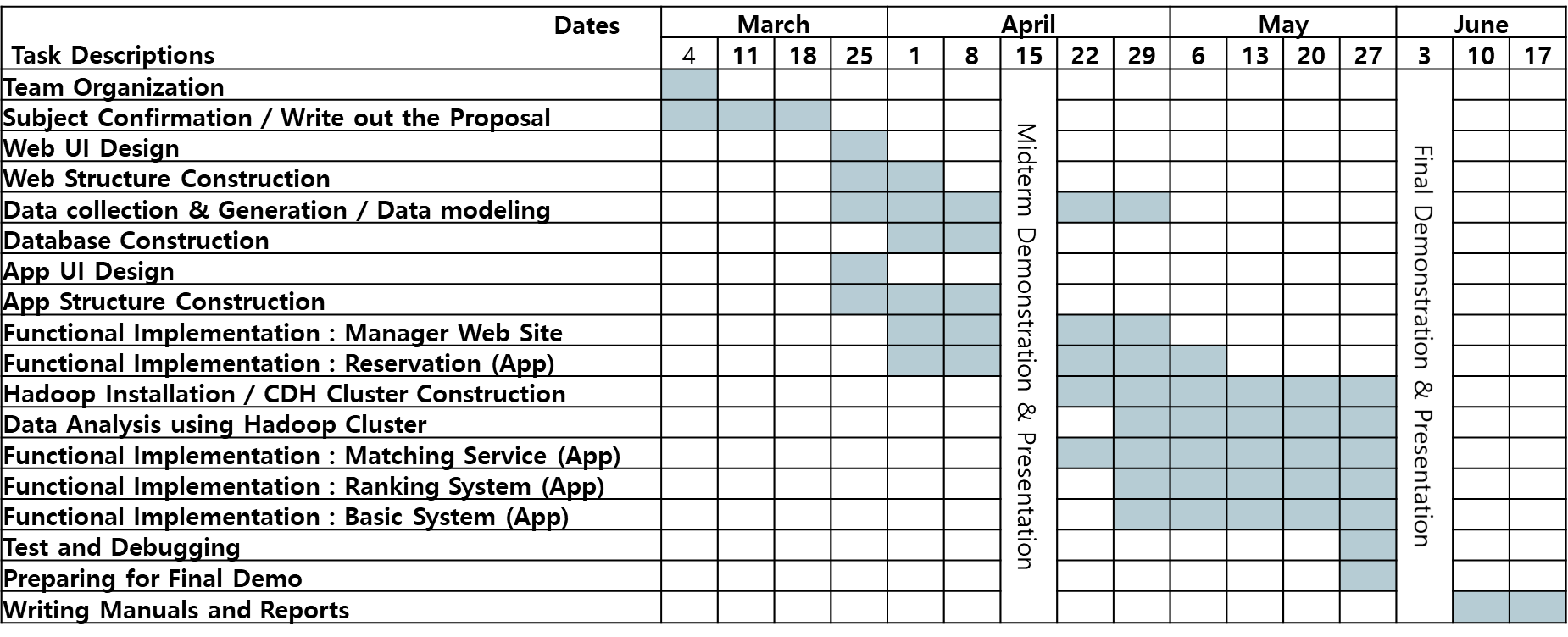
Provides a UI for the user. The user's personal history, the history of the team to which the user belongs, the schedule related to the user's reservation, the schedule related to the open matching and tournament, the place and the team of the opponent team, team information and team recommendation.

1. **DB structure & Hadoop Cluster**

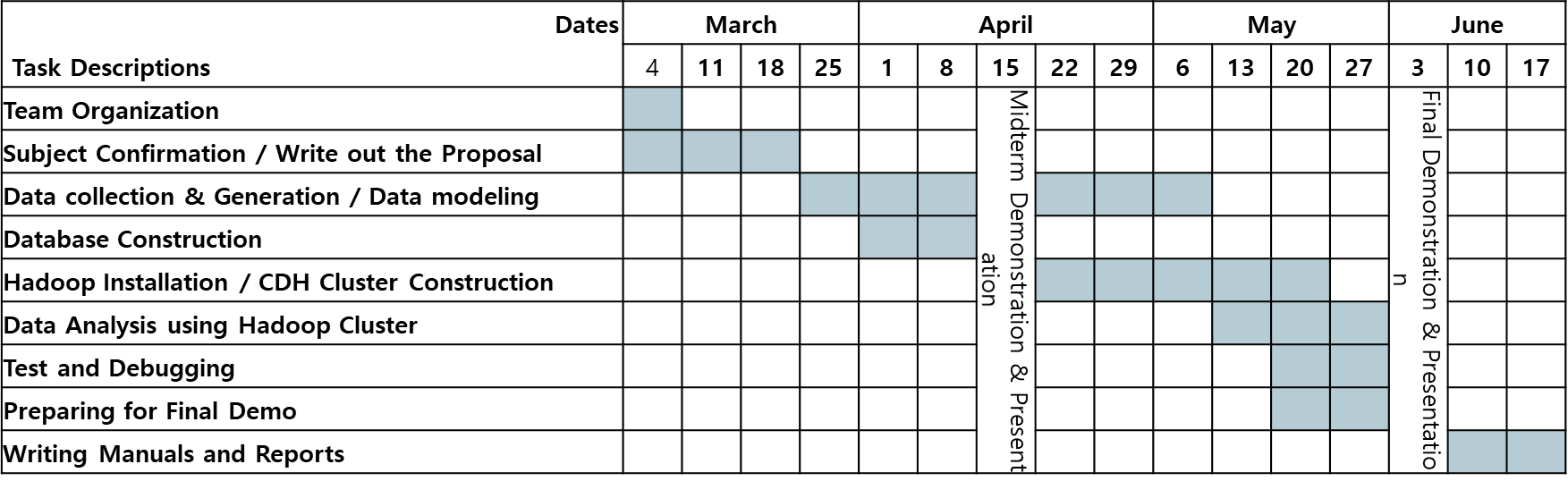
You need an RDB to manage the data generated by the app and the web. The RDB will store real-time data on facilities, schedules, and competition records. By building a Hadoop cluster, we import the related data using the early morning hours with a small number of users, analyze it using the data, and provide various recommended functions based on the analyzed data.

## **1.4 Project Schedule**

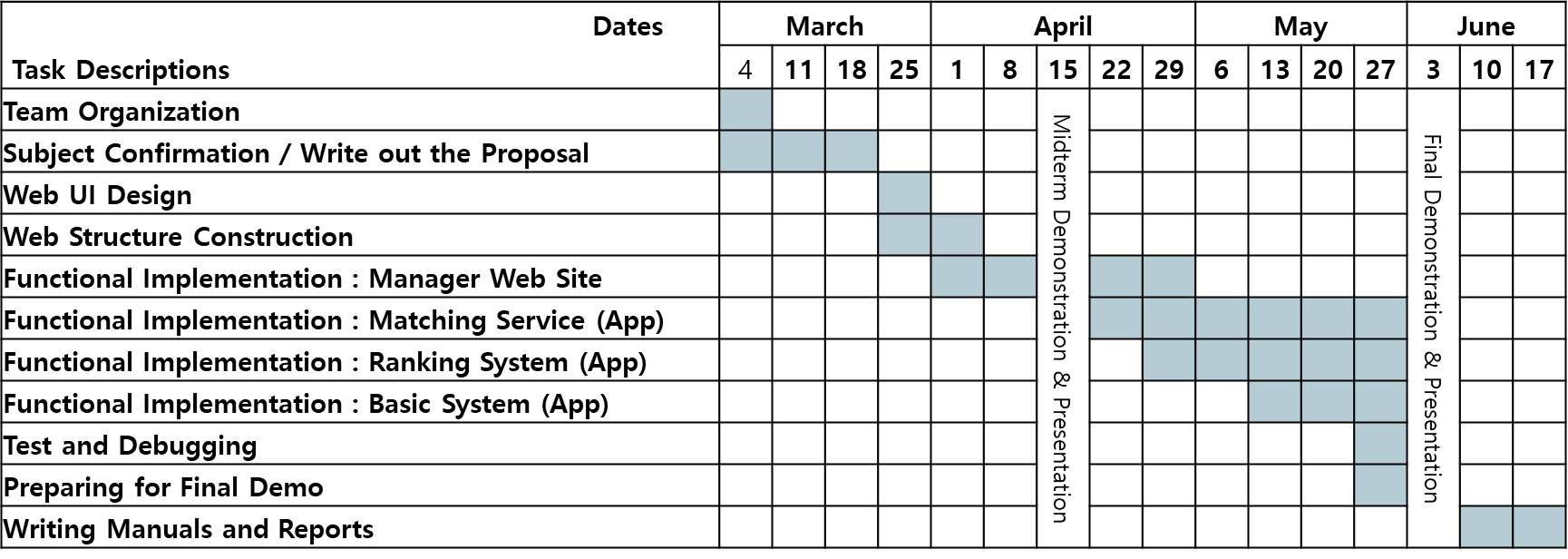
1. **Team Schedule**



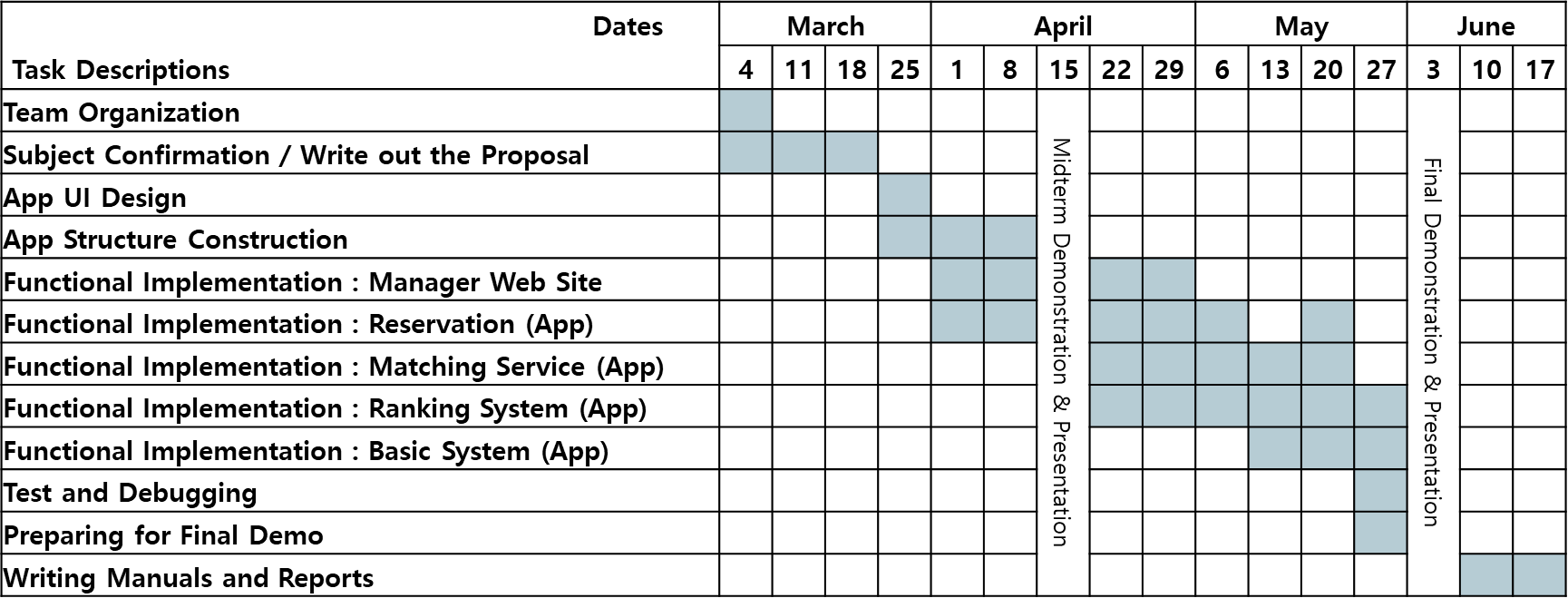
1. **Personal Schedule: DB Construction & Data Analysis, Recommendation (Lee, Donghyun)**



1. **Personal Schedule : Web Structure construction, Web & app implementation (Kong, Chanhyung)**



1. **Personal Schedule : App Structure construction, Web & App implementation (Geum, Kanghyeon)**



# Web UI : UI for Administrator

## **2.1 Purpose**

The management web page of the Integrated Sports Solution for Daily Life aims to provide gym facility leasing, personnel management and schedule management functions so that the gym manager can integrate and manage the above items. The web page allows the gym manager to set up the initial gym registration, facility registration, and holiday setting or schedule, open match, and tournament schedule, and the user has to participate in the schedule through the application-based platform.

## **2.2 Functionality**

Typical functions provided by web pages are as follows.

1. **Login Function**

The login function is realized through the ID and password given to the system administrator. After logging in, you can see information about the gym and the status of your reservation.

1. **Gym Registration Function**

A user who does not register a gym can register a gym to be managed through the gym registration function. To register a gym, you need the name of the gym, the location of the gym, the main items of the gym, and a photo of the gym.

1. **Facility registration function**

You have not registered your facility in the gym, or you may use the facility registration feature to add or delete it. For each facility registration, the name of the facility, the type of facility, the maximum number of people who can accommodate the facility, the minimum number of people to reserve the facility, and the time available for the facility are required.

1. **Closed day registration function**

You can use the holiday registration function to set the holiday day of the gym and set a holiday or a holiday to make it unavailable. Closed days are set on a daily basis, and you can set holiday days on the Closing tab of the Web page.

1. **Reservation, open match, tournament schedule registration function**

You can use the schedule registration function to register each facility-specific reservation, open match, and tournament schedule. In the schedule registration tab of a web page, a user can register a schedule through a mouse input into a schedule table displayed on a weekly basis. Each schedule is divided into a reservation / open match / tournament, and can be distinguished by different colors of each schedule.

The registered schedule is permanently stored in the database server, and can participate in the registered schedule through the application-based platform described later, or can select the opponent to proceed the competition.

1. **Current Reservation Status and Participant Recall Function**

You can use the schedule inquiry function to inquire the list of the team or individuals participating in the currently registered schedule, the number of the participating persons, and the like. The gym manager can change the schedule for the next month or modify it efficiently based on the current booking conditions.

1. **Tournament post registration function**

The administrator can notify the tournament holding information through tournament posting registration function before and after registering the tournament schedule. The post registration function is provided in the form of a bulletin board, and the title, the date of creation, the contents of the tournament, and the like can be attached and registered with the photograph.

## **2.3 Development Environment**

The development environment of web service for gym manager of integrated sports management management solution is as follows.

Web Stack: Spring MVC

Sql translater: mybatis

View Resolution: JSTL, JQuery

WAS: Tomcat 6.0

Source control: go

Db connector: mysql-connector-java

Deployed Server: Centos 7.0

Project control: Maven

Java Version: jre7 or jdk7

## **2.4 Implementation Details**

1. **Spring MVC**

The behavior of the Spring MVC framework is shown in the figure below.

<Figure 2-1>

A URL-based request is sent to the web server via the user's web browser, and the Handler Mapper passes the request to the appropriate controller. The controller performs the database work or the business logic work through the incidental Json information that comes with the request. The View Resolver is created through the result of the work performed, and the View page is completed. And then processes the user's request by responding through the completed page.

1. **Bean**

To use Dependency Injection (DI) provided by the Spring Framework, we used the following annotation.

@RequestMapping: Maps each controller to the request.

Declare @Controller: Controller, and let Spring Framework manage dependencies.

We declare @Service: Service and inject dependencies by requesting that the Spring Framework manage the objects, not the code, on request.

@DAO: Declare a data access object.

These annotations are registered as spring beans, created as objects managed on the spring framework, and dependencies are managed.

1. **Mybatis**

I used the mybatis module to read and store data through the database. Each query used is stored in xml format, and specifies the workspace and name when saving. Each stored SQL query is registered as a bean in the same way as the controller, and the dependency is managed in the spring framework.

1. **JSTL**

JSTL was used to reflect the results of the database and business logic on the web page. JSTL plays the role of View Resolver that completes the page by inserting appropriate model data into the middle inserted tag of the web page written in Html in the process of combining the data calculated in the Spring server, that is, the template and the template of the basic web page.

1. **Publishing**

We used the FullCalander API and the web page template provided by W3School to improve the UI to implement the schedule function of the web page.

The FullCalander API provides monthly and weekly calendars, with monthly calendars set for holiday days and weekly calendars for calendar entries. We also modified the code of the API in order to separately register the cause of absence, schedule, reservation, open match, and tournament.

1. **WAS**

We built Web Application Server based on Tomcat 6.0 version and set it to operate using port 80 basically. During distribution in the local development environment, I found port related problems and Maven errors, and failed to solve them, so I performed demonstration and operation tests in the local environment.

# 3. App UI : UI for Users

## **3.1 Application Server**

1. **Purpose**

The application server is developed to perform the data processing and operation necessary for the screen configuration and the function operation in the application platform. The application server performs database inquiry, registration and deletion through queries, and data conversion and processing for use in applications .

The application server is written in NodeJS and operates through the PM2 module. Project management required for server operation is managed through NPM.

1. **Development environment**

platform : NodeJS

WAS : PM2

Database connector : mysql-connector

OS : centos 7.0

## **3.2 Application UI**

1. **Development environment**

Platform : React Native

Emulator : Android Studio / Nexus 5X API 28

Phone : Expo / Galaxy S8

Target : Android 9.0 (**Google Play**)

Dependencies :

"react-native-3d-swiper": "^2.0.7",

"react-native-calendars": "^1.112.0",

"react-native-chart-kit": "^2.6.1",

"react-native-elements": "^1.1.0",

"react-native-maps": "^0.24.2",

"react-native-snap-carousel": "^3.8.0",

"react-native-vector-icons": "^6.4.2",

"react-navigation": "^3.9.1"

## **3.3 Function**

The functions performed by the application server are as follows.

1. **Login**

Click the button on the upper left to move. The user can enter the ID and PWD to log in. If you do not have an ID, you can register through the registration button at the bottom. The information of the logged-in user is stored in a safe storage inside the app, and automatic login is executed every time the app is run.

1. **Register Member**

The user proceeds to check the ID duplication and prevents joining with the same ID. In addition to this, you can select your ID, password, name, gender and preference. Enter the details according to your preference item. Currently, we only go for soccer. You will complete your registration by choosing your main foot, main position, skills and experience.

1. **My Page**

Click the button on the upper left to move. If you are logged in, go to My Page. The profile picture of the currently logged in member, the rank and MMR, the total number of games, the latest winning percentage, and the MVP point. The number of monthly games for the last five months is shown on the chart, showing how many are their rankings for all members. This information is displayed in the same way on all member profile screens, and the logout button is activated when entering the profile of the logged-in user.

1. **Main Screen**

It consists of 3 tabs : Individual Information / Team Information / Team Recommendation

* Individual Information Tab : Reservation and Reservation Status, Matching Status, Result Registration
* Team Information Tab : Team registration, Team profile verification
* Team Recommendation Tab : Team recommendation and search

1. **Reservation**

Select a schedule by Type / Event / Place / Time order.

* Type: The type is a team-based reservation, the individual-based matching, and the reservation is only for the team leader.
* Sports: Supports four categories of soccer, basketball, baseball and badminton, and the current data is generated for the football stadium.
* Place: The gym will appear on the map to support the selected event. You can choose your own gym on the map, or register your favorite gym to quickly select it.

Schedule: Reservation can be made up to 7 days after the date. There are 5 types of reservation schedule.

* Empty Schedule: You can select a team you want to reserve, because you can be the team leader for multiple teams at the time of booking.
* Reservation completed, allow participation X
* Schedule Completed, Allow Participation O, Pending Participation: As with the empty schedule above, you must select a team and see the profile of the team that made the reservation.
* Complete the reservation, allow participation, complete the opponent team participation
* Closed gym

1. **Matching**

* The matching schedule is available only when the minimum number of persons and the maximum number of persons are exceeded and the number of applicants exceeds the minimum number of persons.
* As with scheduling, select a schedule by type / event / place / time order.

Schedule: Matching is possible only up to 7 days after the date. There are four types of matching schedules.

* Minimum number of people
* Meet minimum requirements
* Deadline for allowance
* Closed gym

By selecting an available event, you can see a list of people who have applied for the event, and you can see the applicant's MMR and profile.

1. **Check Reservation Status**

* You can see the details of the schedule you have booked. Gym name, facility name, address, date and time of booking.

1. **Check Matching Status**

* You can see the details of the matching schedule you have booked. Gymnasium name, facility name, address, date and time of reservation, event and status of current applicants, list of participants.
* The Participant List automatically organizes teams according to their ability.

1. **Input Game Result**

* After the game is over, the user can enter the results of the match. In the case of reservations, only the team leader who booked the schedule can enter the results. In the case of matching, all the participants enter the results and analyze the results to finally reflect the results of the competition.

1. **Register Team**

* You can register a new team by clicking the Team Registration button on the Team Info tab. Proceed with a duplicate check to prevent the occurrence of a team of the same name. It is possible to select the main item. The user who registered the team automatically becomes the team leader of the team.

1. **Team Profile**

* On the Team Info tab, select a team to view the team's profile. The team's profile picture, the rank and MMR, the total number of games, the recent winning percentage, and the number of team members. The charts show the number of monthly games played in the last five months, and how many of their rankings are among the team members. If you are not a team you are a member of, you will be shown first team members in the team.
* In the pre-match tab you can see the team's odds and the most recent match. When, where, where, and with which team the score and the MVP of the match are displayed.
* On the Team Members List tab, you will see a list of team members who have joined the team, search for team members, and click to view your personal profile.

1. **Team recommendation and search**

* The Team Referrals tab suggests teams with similar skills based on the MMR of the currently logged in user. If you do not enter anything in the top search bar, it shows the team recommendation screen. If you enter a search term, the team name shows the team that contains the search term. You can select a team to view your team profile

# 4. DB and Data Analysis

We used RDBMS and Hadoop clusters to handle the data generated by the web and the app. Two tools were used to distinguish between OLTP and OLAP. The RDBMS was used to handle the data on facilities, schedule, reservations and competition results generated in real time. Hadoop clusters were used for analysis and recommendation. AWS EC2 instance is used to enable external connection. Detailed specifications of RDBMS and Hadoop cluster are as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| **RDBMS** | | | |
| **Server Spec** | **Server OS** | **RDBMS** | **Ver.** |
| AWS EC2 t2.micro (vCore : 1, RAM : 1GB, Disk : 30GB) | RHEL 7.6 | MariaDB | 5.5.60 |
| **Hadoop Cluster** | | | |
| **Hosts Spec** | **Server OS** | **Hosts #** | **CDH Ver.** |
| AWS EC2 m5.xlarge(vCore : 4, RAM : 16GB, Disk : 100GB) | CentOS 7.6 | 5 | 5.16 |

## **4.1 DB Structure**

Data is managed with a total of 28 tables. Adjacent tables painted in the same color have similar properties. The table can be classified into user, user 's overall sport, team, game reservation and result, gymnasium, sport, achievement, and tournament.

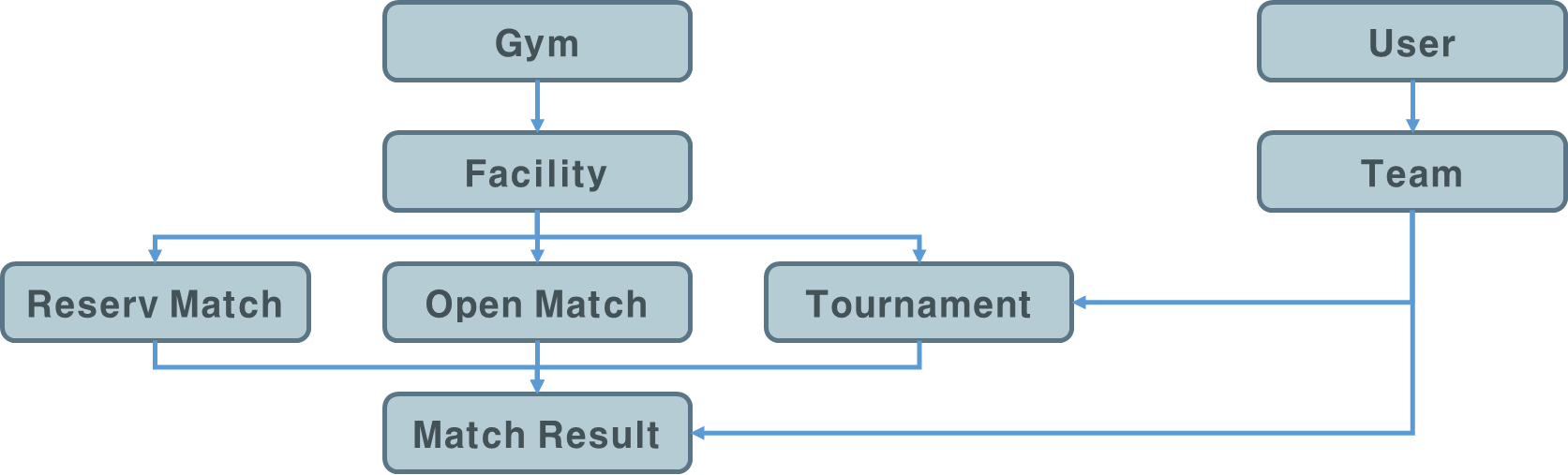
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **테이블 명** | **Table\_name** |  | **테이블 명** | **Table\_name** |
| 사용자 | User |  | 경기결과 | match\_result |
| 사용자별관심종목 | pref\_subj\_per\_user |  | 경기참여자명단 | match\_participant |
| 사용자별 업적 | badge\_per\_user |  | 예약경기 | reserv\_matches |
| 축구전적 | soccer\_record |  | 오픈매칭 | open\_matches |
| 야구전적 | baseball\_record |  | 오픈매칭 참가자 | open\_match\_participant |
| 농구전적 | basketball\_record |  | 체육관 관리자 | gym\_admin |
| 배드민턴전적 | badminton\_record |  | 체육관 | gym |
| 사용자별선호체육관 | pref\_gym\_per\_user |  | 시설정보 | fac\_info |
| 사용자별선호체육시설 | pref\_fac\_per\_user |  | 시설 일정 | fac\_schedule |
| 팀 | team |  | 종목 | subject |
| 팀별 업적 | badges\_per\_team |  | 업적 | badge |
| 팀별선호체육관 | pref\_gym\_per\_team |  | 토너먼트 게시판 | tournament\_board |
| 팀별선호체육시설 | pref\_fac\_per\_team |  | 토너먼트 | Tournament |
| 팀별팀원목록 | team\_user\_list |  | 토너먼트 일정 | tournament\_schedule |

The schema for each table is shown below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table(Eng)** | **Schema** | **Schema(Eng)** | **Datatype** | **KEY** | **NOT NULL** |
| User | UDID | UDID | INTEGER | PK | ◎ |
| ID | ID | VARCHAR(50) | UK | ◎ |
| PWD | PWD | VARCHAR(50) |  | ◎ |
| 프로필 사진 | profile\_fig | MEDIUMBLOB |  |  |
| 이름 | name | VARCHAR(50) |  | ◎ |
| 성별 | gender | VARCHAR(50) |  | ◎ |
| 생년월일 | birth | DATE |  | ◎ |
| e-mail | email | VARCHAR(50) |  |  |
| 폰번호 | phone | VARCHAR(50) |  |  |
| 키 | height | FLOAT |  |  |
| 몸무게 | weight | FLOAT |  |  |
| 지명 | location | VARCHAR(200) |  |  |
| pref\_subj\_per\_user | UDID | UDID | INTEGER | FK | ◎ |
| 종목명 | subj\_ID | INTEGER | FK | ◎ |
| badge\_per\_user | UDID | UDID | INTEGER | FK | ◎ |
| 업적명 | badge\_ID | INTEGER | FK | ◎ |
| soccer\_record | UDID | UDID | INTEGER | FK,PK | ◎ |
| 주 발 | main\_foot | VARCHAR(50) |  |  |
| 주 포지션 | main\_position | VARCHAR(50) |  |  |
| 주관적인 실력 | subjective\_skill | VARCHAR(50) |  |  |
| 경력(연차) | career | INTEGER |  |  |
| MMR | MMR | INTEGER |  |  |
| mvp 포인트 | mvp\_point | INTEGER |  |  |
|  | Implement later… |  |  |  |
| baseball\_record | UDID | UDID | INTEGER | FK,PK | ◎ |
|  | Implement later… |  |  |  |
| basketball\_record | UDID | UDID | INTEGER | FK,PK | ◎ |
|  | Implement later… |  |  |  |
| badminton\_record | UDID | UDID | INTEGER | FK,PK | ◎ |
| 단식/복식 | pref\_doubles | BOOLEAN |  |  |
| 주 손 | main\_hand | VARCHAR(50) |  |  |
| 주관적인 실력 | subjective\_skill | VARCHAR(50) |  |  |
|  | 경기 분석 결과 추가 |  |  |  |
| pref\_gym\_per\_user | UDID | UDID | INTEGER | FK | ◎ |
| 체육관 ID | gym\_ID | INTEGER | FK | ◎ |
| pref\_fac\_per\_user | UDID | UDID | INTEGER | FK | ◎ |
| 시설 ID | fac\_ID | INTEGER | FK | ◎ |
| team | 팀 ID | team\_ID | INTEGER | PK | ◎ |
| 팀 명 | team\_name | VARCHAR(50) |  | ◎ |
| 팀장 UDID | team\_leader\_UDID | INTEGER | FK | ◎ |
| 팀 MMR | team\_MMR | INTEGER |  |  |
| 팀 주종목 | team\_main\_subj | VARCHAR(50) | FK |  |
| 승률 | winning\_rate | FLOAT |  |  |
| badges\_per\_team | 팀 ID | team\_ID | INTEGER | FK | ◎ |
| 업적 ID | badge\_ID | INTEGER | FK | ◎ |
| pref\_gym\_per\_team | 팀 ID | team\_ID | INTEGER | FK | ◎ |
| 체육관 ID | gym\_ID | INTEGER | FK | ◎ |
| pref\_fac\_per\_team | 팀 ID | team\_ID | INTEGER | FK | ◎ |
| 시설 ID | fac\_ID | INTEGER | FK | ◎ |
| team\_user\_list | 팀 ID | team\_ID | INTEGER | FK | ◎ |
| UDID | UDID | INTEGER | FK | ◎ |
| match\_result | 경기 ID |  | INTEGER | FK | ◎ |
| 승자팀 ID | win\_team\_ID | INTEGER | FK | ◎ |
| 패자팀 ID | lose\_team\_ID | INTEGER | FK | ◎ |
| 체육관 ID | gym\_ID | INTEGER | FK | ◎ |
| 시설 ID | fac\_ID | INTEGER | FK | ◎ |
| 종목 ID | subj\_ID | INTEGER | FK | ◎ |
| 경기 결과 | score | VARCHAR(50) |  |  |
| mvp | MVP\_UDID | INTEGER |  |  |
| match\_participant | 경기 ID | match\_ID | INTEGER | FK | ◎ |
| 사용자 ID | UDID | INTEGER | FK | ◎ |
| 팀 ID | team\_ID | INTEGER | FK | ◎ |
| 승패여부 | is\_win | INTEGER |  | ◎ |
| reserv\_matches | 예약 ID | reserv\_ID | INTEGER | FK | ◎ |
| 예약 종목 | subj\_ID | INTEGER | FK | ◎ |
| 예약 팀 ID | reserv\_team\_ID | INTEGER | FK | ◎ |
| 상대 팀 ID | opponent\_team\_ID | INTEGER | FK |  |
| 팀vs팀 여부 | is\_solo | INTEGER | FK | ◎ |
| open\_matches | 예약 ID | reserv\_ID | INTEGER | FK | ◎ |
| 예약 종목 | subj\_ID | INTEGER | FK | ◎ |
| 사용자 UDID | UDID | INTEGER | FK | ◎ |
| gym\_admin | 관리자 UDID | UDID | INTEGER | PK | ◎ |
| 관리자 ID | ID | INTEGER | UK | ◎ |
| 관리자 PWD | PWD | VARCHAR(50) |  | ◎ |
| 관리자 이름 | name | VARCHAR(50) |  | ◎ |
| 관리자 성별 | gender | VARCHAR(50) |  | ◎ |
| 관리자 생년월일 | birth | VARCHAR(50) |  | ◎ |
| e-mail | email | VARCHAR(50) |  |  |
| 핸드폰 번호 | phone | VARCHAR(50) |  |  |
| 체육관 ID | gym\_ID | INTEGER | FK |  |
| gym | 체육관 ID | gym\_ID | INTEGER | PK | ◎ |
| 체육관 이름 | gym\_name | VARCHAR(200) |  | ◎ |
| 체육관 사진 | gym\_fig | MEDIUMBLOB |  |  |
| 체육관 위치 | gym\_location | VARCHAR(200) |  |  |
| 체육관 위도 | gym\_latitude | FLOAT |  |  |
| 체육관 경도 | gym\_longitude | FLOAT |  |  |
| 이용가능 시작시간 | avail\_starttime | TIME |  |  |
| 이용가능 종료시간 | avail\_endtime | TIME |  |  |
| 체육관 추가정보 | gym\_info | VARCHAR(500) |  |  |
| 체육관 관리자 ID | gym\_admin\_ID | INTEGER | FK | ◎ |
| 종목정보 | subj\_info | VARCHAR(500) |  |  |
| fac\_info | 체육관 ID | gym\_ID | INTEGER | FK | ◎ |
| 시설 ID | fac\_ID | INTEGER | PK | ◎ |
| 이용가능 시작시간 | avail\_starttime | TIME |  |  |
| 이용가능 종료시간 | avail\_endtime | TIME |  |  |
| 수용인원 | avail\_participant | INTEGER |  |  |
| 시설 이름 | fac\_name | VARCHAR(200) |  | ◎ |
| 시설 종목 | subj\_ID | INTEGER | FK | ◎ |
| subject | 종목 ID | subj\_ID | INTEGER | PK | ◎ |
| 종목 이름 | subj\_name | VARCHAR(50) | UK | ◎ |
| badge | 업적 ID | badge\_ID | INTEGER | PK | ◎ |
| 업적 이름 | badge\_name | VARCHAR(200) | UK |  |
| 업적 타입 | badge\_type | INTEGER |  | ◎ |
| 업적 정보 | badge\_info | VARCHAR(200) |  |  |
| fac\_schedule | 일정 ID | schedule\_ID | INTEGER | PK | ◎ |
| 체육관 ID | gym\_ID | INTEGER | FK | ◎ |
| 시설 ID | fac\_ID | INTEGER | FK | ◎ |
| 종목 ID | subj\_ID | INTEGER | FK | ◎ |
| 일정 명 | schedule\_name | VARCHAR(200) |  | ◎ |
| 일정 설명 | schedule\_detail | VARCHAR(500) |  |  |
| 일정 타입 | schedule\_type | INTEGER |  | ◎ |
| 일정 시작시간 | starttime | DATETIME |  | ◎ |
| 일정 종료시간 | endtime | DATETIME |  | ◎ |
| 최소 참여인원 | min\_participant | INTEGER |  | ◎ |
| 최대 참여인원 | max\_participant | INTEGER |  | ◎ |
| tournament\_board | 토너먼트 ID | tournament\_ID | INTEGER | PK |  |
| 제목 | title | VARCHAR(200) |  |  |
| 내용 | contents | MEDIUMBLOB |  |  |
| Tournament | 토너먼트 ID | tournament\_ID | INTEGER | FK |  |
| 이름 | title | VARCHAR(200) |  |  |
| 시작기간 | starttime | DATE |  |  |
| 종료기간 | endtime | DATE |  |  |
| tournament\_schedule | 토너먼트 ID | tournament\_ID | INTEGER | FK |  |
| 일정 ID | schedule\_Id | INTEGER | FK |  |

## **4.2 DB Data Generation**

In order to implement the recommendation function, which is one of the main functions of the platform, data on existing matches to be used for recommendation are needed. The recommendation is to utilize the stored data through the competition or facility and schedule registration. Data was generated only for soccer events among the four events, and data was generated in the following order.



* 1. **Gym**

We crawled 'Seoul City Public Sports Facilities Status (2018)' data provided in 'Seoul Open Data Portal' and registered 207 public sports facilities in Seoul city.

* 1. **Facility**

Based on the information on public sports facilities, only one or two soccer field facilities were registered in the gymnasium that supports soccer games. For the registered soccer field, schedule was created from April 1 to June 20 at the following ratio.



First, the probability that an administrator opens a schedule on a certain date is 60%. We can create more schedules, but we set it up considering the load of DB server. For 60% of the dates for which you want to generate data, 5% is the closing date. For the remaining dates, create the schedule in the same ratio as above.

* 1. **User**

In the case of user data, the name, date of birth, gender, key, and weight may be utilized, so that data is generated with certainty. We created the user's name through a combination of about 50 names and names, and created the user's key and weight at a similar rate. A total of about 5000 users were created.

* 1. **Records**

And has a table for storing information necessary for each sport for each of the four sports provided by the corresponding platform. In the case of soccer, ancillary data on the user's performance constituted by the initial input of the user is generated, and data is generated so as to follow the Gaussian distribution with an average of 2000 points in MMR. Therefore, the MMR between 540 and 3719 points was constructed for soccer.

* 1. **Team**

Based on the user information we created, we created a team of at least 22 to 50 people. The minimum number of players is 22, because the minimum number of players who can play soccer is 22. Actually, the number of team members who can belong to one team can be changed while operating the platform. A total of 150 teams were created.

  In the case of MMR, the Gaussian distribution is used as well as the record of each event. The score of each team is composed of MMR between 815 and 3087 points.

* 1. **Matches**

Based on the team and schedule information we created, we generated data for booking matches and open matching. It means that a team or an individual has reserved a specific game through the application in a time zone in which the manager can register on the web. Details of the reservation type are as follows.

1. Reserved Match

Assuming that the actual platform is operational, it is assumed that a particular team would have played a lot in a particular facility. In the beginning, the concept of reservation game had many concepts similar to 'home stadium'. Therefore, it was decided to schedule frequent teams for each facility and decide whether to play with other teams by allowing the team members or interrupts. Therefore, each team has a high number of competitions in a specific area or specific facility.

1. Open Match

In case of open matching, since the open match is not represented by one representative, the open match schedule registered in the facility must be mapped to the UDID of the participant. When the users perform the open matching reservation in real time, the team is automatically configured referring to the MMR information of the reserved user.

* 1. **MatchResult**

Randomly record the race results for the team or individual who registered the schedule. Based on the recorded results, it will be reflected in individual and team excellence.

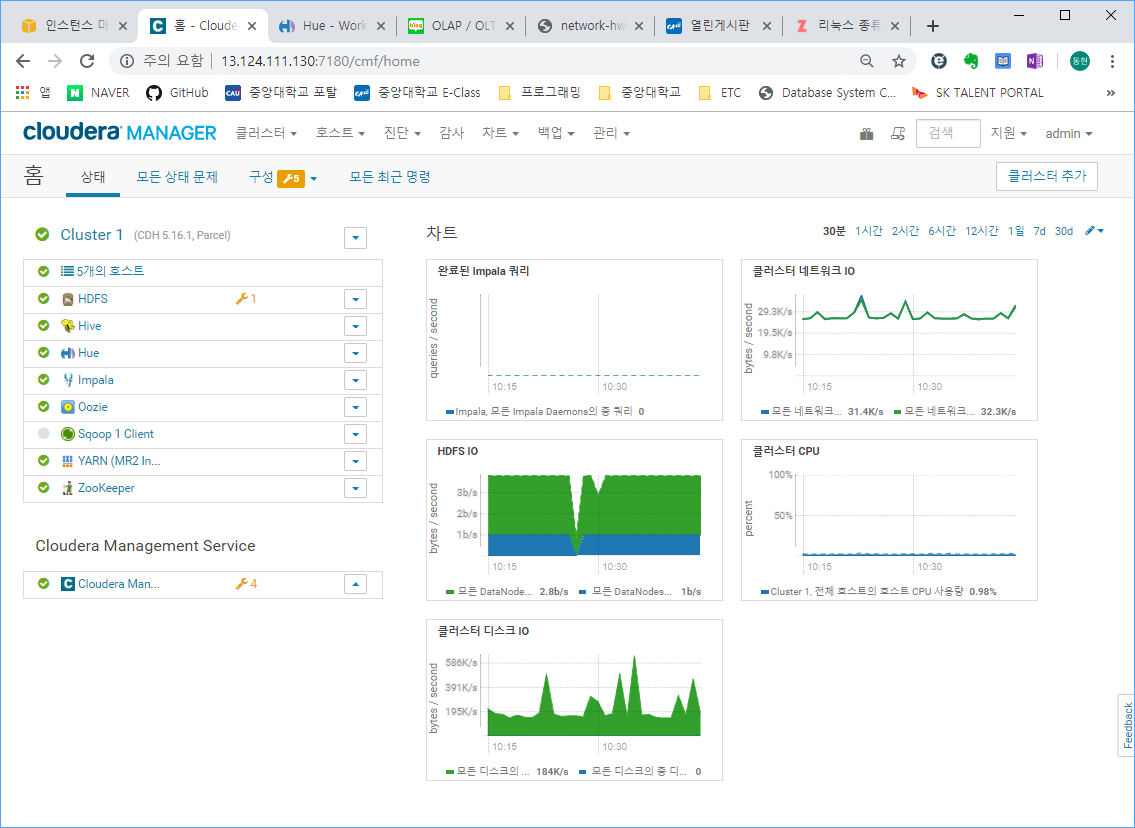
## **4.3 Data Analysis**

Based on the previously generated data, Hadoop imports the data stored in the RDBMS and analyzes the data. Then, the Scheduler that exports the analyzed data to the RDBMS is constructed to implement the data analysis and recommendation system. We will describe what Hadoop is used for this purpose, and specifically what services in Hadoop are used for the analysis.

1. **What is Hadoop?**

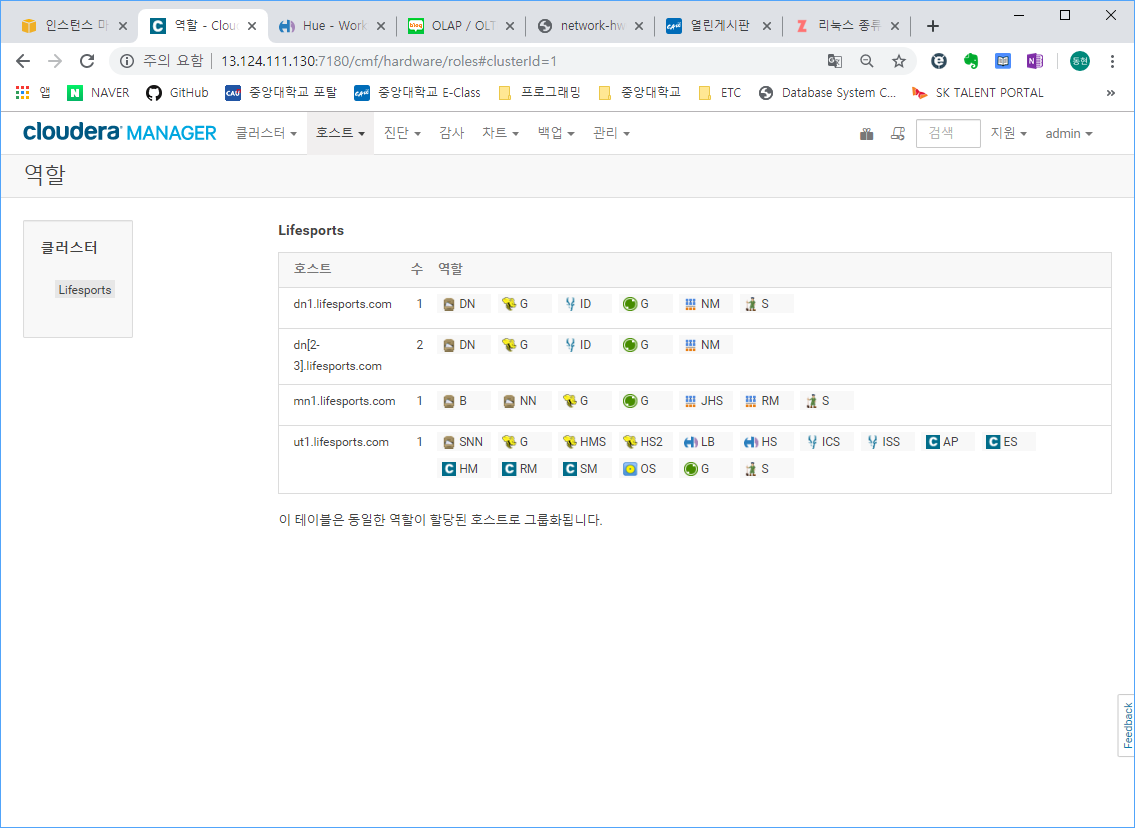
Hadoop provides data storage and processing services in a distributed cluster environment. Compared with the existing RDB, it can construct a cluster with several low cost servers to distribute a large amount of data at a lower cost, and to process the distributed stored data in parallel. Furthermore, a variety of services that support collection, storage, processing, analysis and visualization of data through distributed clusters have been added to the Hadoop Eco-system, making OLAP a cost-effective tool for OLAP domains.

Hadoop, which is being developed as an open source, is currently available in version 3.0. However, there are various distribution versions such as MapR, Hortonworks HDP, Cloudera CDH, etc., because it is difficult to install various Hadoop-Ecosystem and manage nodes composed of several servers separately from Hadoop core service YARN and HDFS. For the project, data analysis was conducted using CDH 5.16 version.



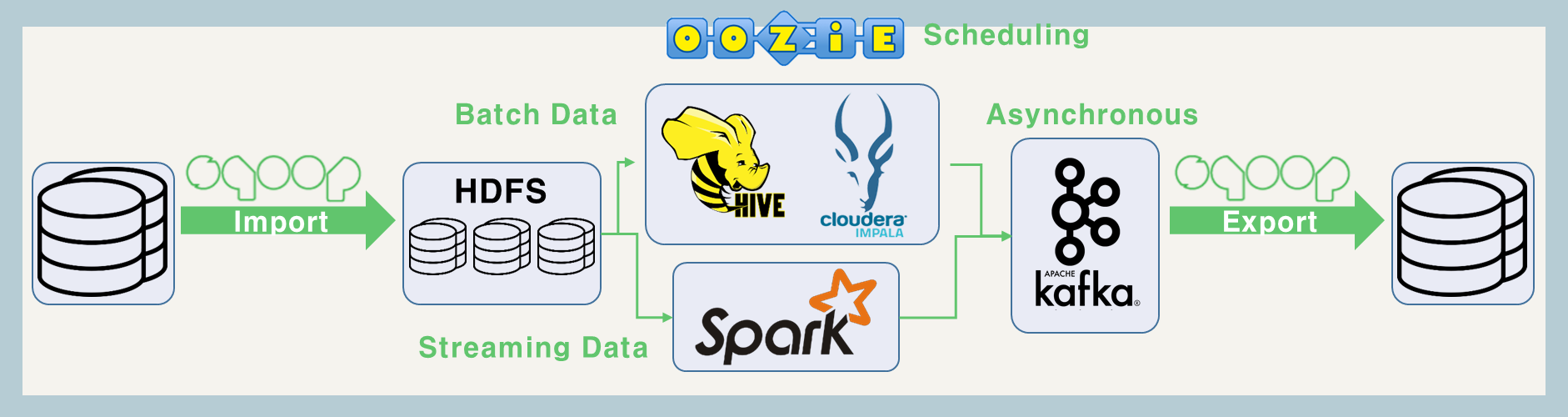
1. **Cluster building environment**

Clusters were configured using five AWS EC2 m5.xlarge instances. Cluster service configuration that is confirmed through Cloudera manager which is a tool used for cluster management is as follows.



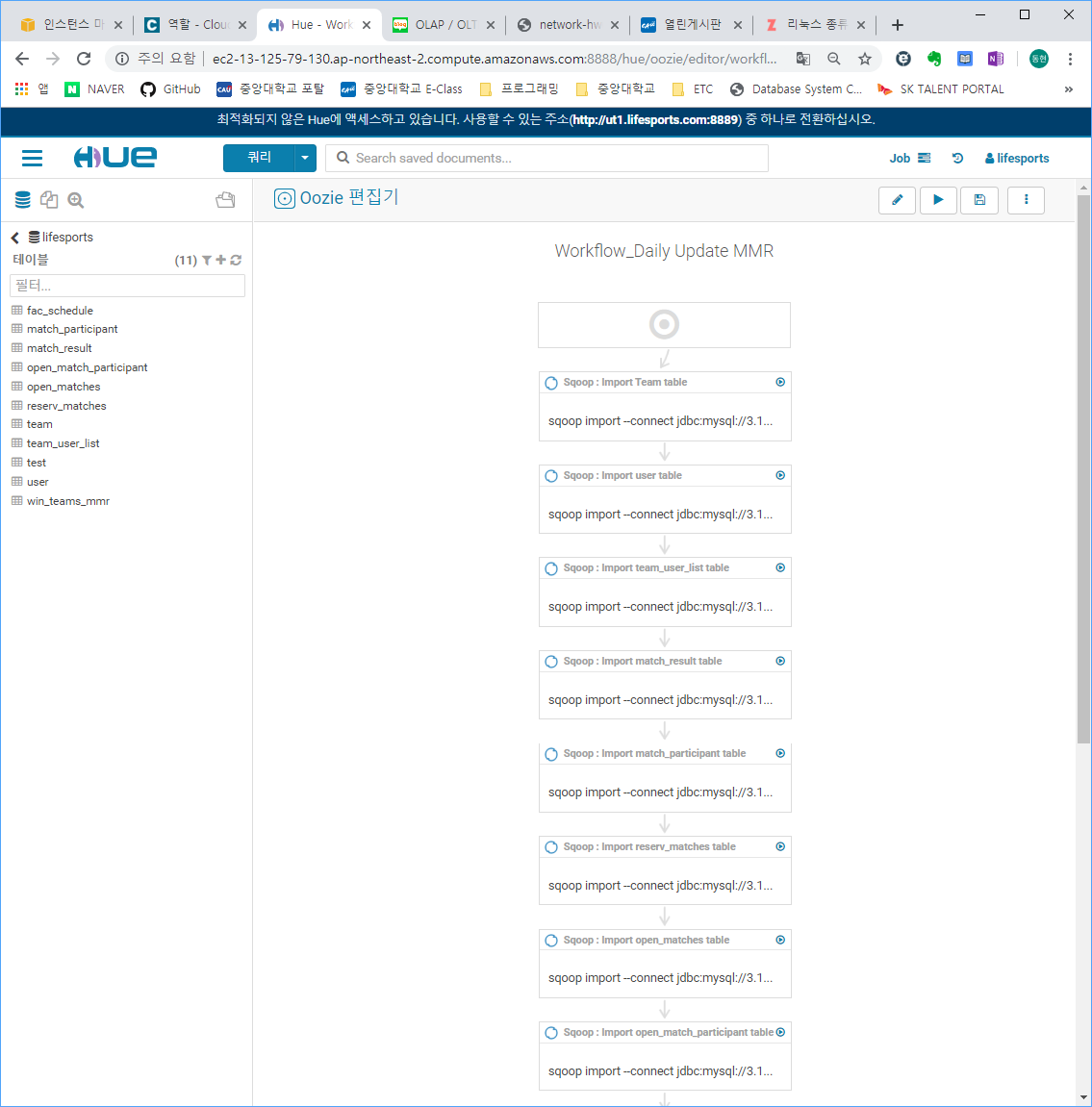
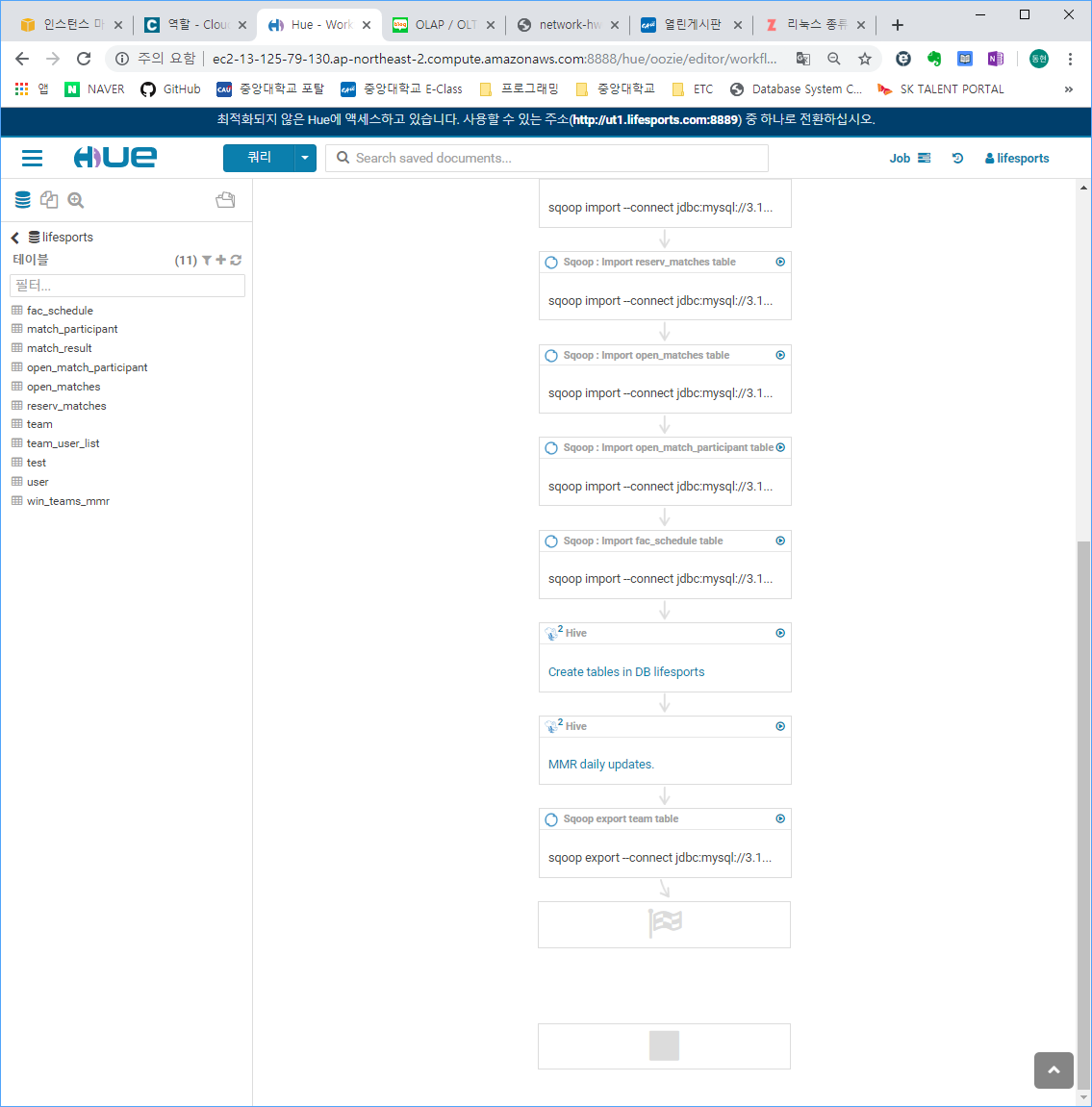
|  |  |  |  |
| --- | --- | --- | --- |
| **Service** | **Roles** | | **Description** |
| **HDFS** | NN | Namenode | Metadata for blocks in Datanode has been stored here. |
| SNN | Secondary Namenode | Garauntee fault tolerance by updating fsimage and editslog file regularly. |
| DN | Datanode | Place that the blocks saved. Metadata for the blocks are saved in namenode. |
| B | HDFS Balancer | Analyzes block placement and balances data across the DataNodes. |
| **YARN** | JHS | Job History Server | Save the logs about the job executed like how many containers assigned and so on. |
| RM | Resource Manager | Determine how much resource would be assigned for the job |
| NM | Node Manager | Manage the resource and schedule for jobs in each node |
| **Sqoop** | G | Gateway | Sqoop job can be submitted through the nodes that the sqoop gateway installed. |
| **Hive** | G | Gateway | Hive job can be submitted through the nodes that the hive gateway installed. |
| HS2 | Hive Server2 | Compile the hive query to MR job so that the hive job run |
| HMS | Hive Metastore Server | Save the meta data about hive databases |
| **CM** | AP | CMS Alert Publisher | generates and delivers alerts for certain types of events |
| ES | CMS Event Server | aggregates relevant Hadoop events and makes them available for alerting and searching |
| HM | CMS Host Monitor | collects health and metric information about hosts |
| RM | CMS Report Manager | generates reports that provide an historical view into disk utilization by user, user group, and directory, processing activities by user and YARN pool, and HBase tables and namespaces. |
| SM | CMS Service Monitor | collects health and metric information about services and activity information from the YARN and Impala services |
| **Oozie** | OS | Oozie Server | Manage the oozie job by deploying oozie launcher for each job |
| **Zookeeper** | S | Zookeeper Server | Used for fault tolerance and load balancing. At least three nodes contain the server |
| **Hue** | HS | Hue Server | Provide the GUI for almost jobs in Hadoop eco-service and provide file browser and job browser, and so on. |
| LB | Hue Load Balancer | Load balancing for the jobs hue submitted. |

1. **Analyze data using Hadoop**

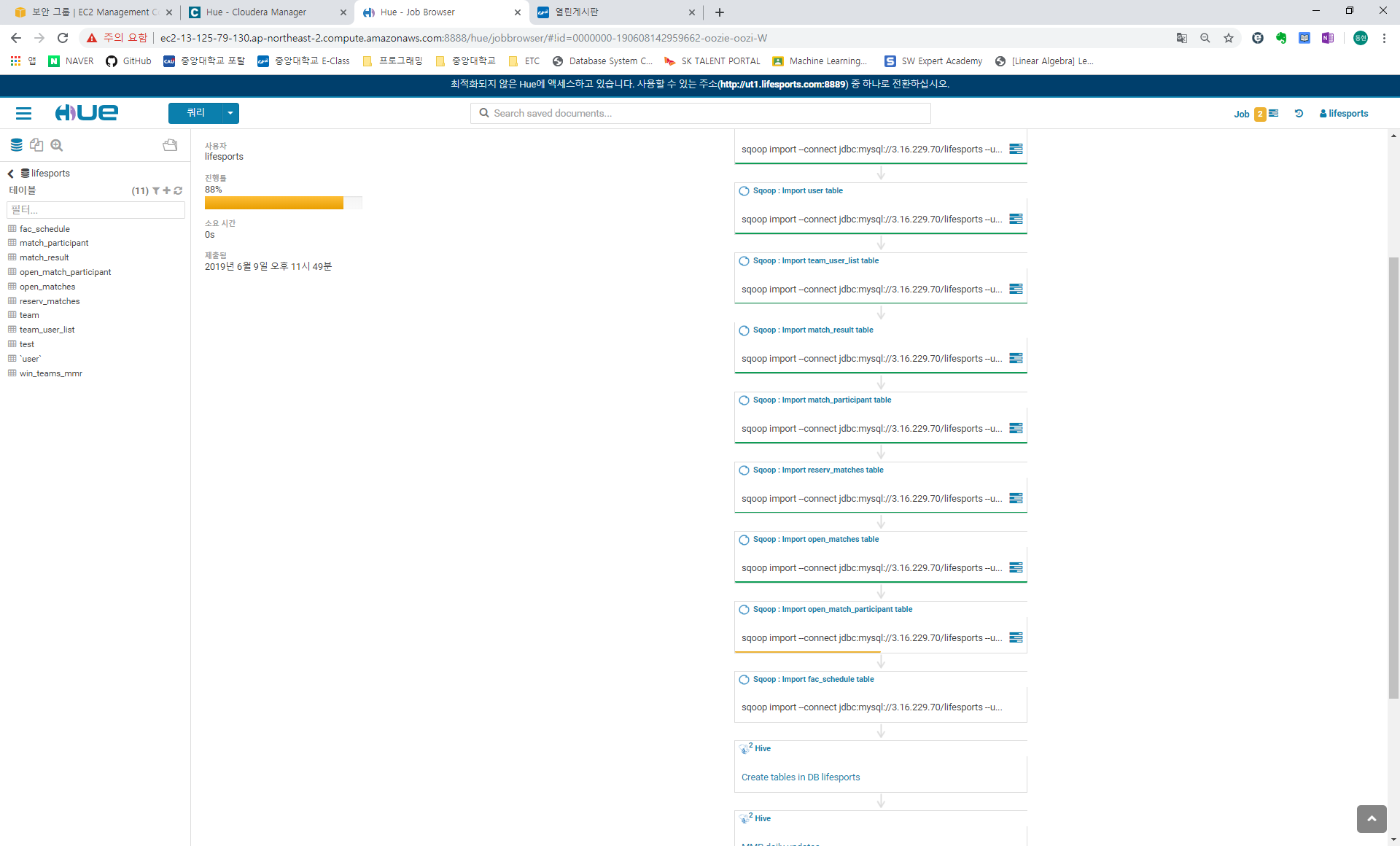


Hadoop application plan that project is like above. For large batch data, analysis was performed using Hive and impala. For data requiring real-time decision, Spark was trying to analyze, but only batch data analysis using Hive / Impala was performed due to cluster performance problem. Import the necessary tables from the RDB into HDFS using Sqoop, and assign the directory to the location of the external table for Hive or Impala analysis. Impala and Hive are both analytic tools that use almost the same syntax as SQL. In Impala, Hive's metastore is used, daemon is used to enable faster queries, and Hive supports a variety of SQL Syntaxes compared to Impala. So that they can be utilized properly. Store the analyzed data in Kafka and export the stored data asynchronously to the existing RDB table. This entire process allows Oozie to register the scheduler so that it can be performed automatically in the early morning hours with a small number of users.

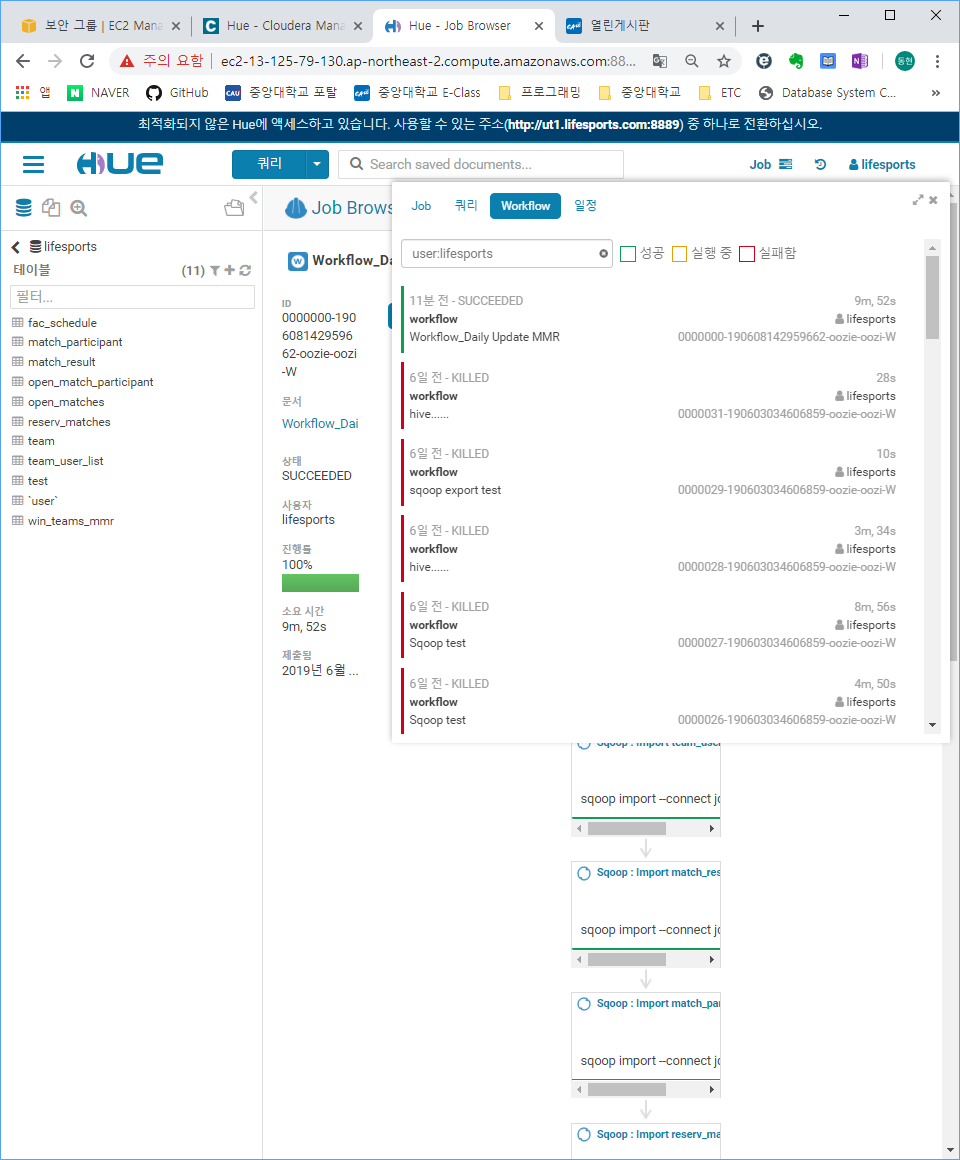
 The screens shown below are oozie workflows and schedules that receive information about the games performed during the day and update the MMR information of users and teams. MMR update and other information to be updated on a daily basis, data is imported from RDB using Sqoop, data is analyzed using Hive, updated table is configured, and then the table is exported to an existing RDB, Allows tables in the RDB to be updated daily.



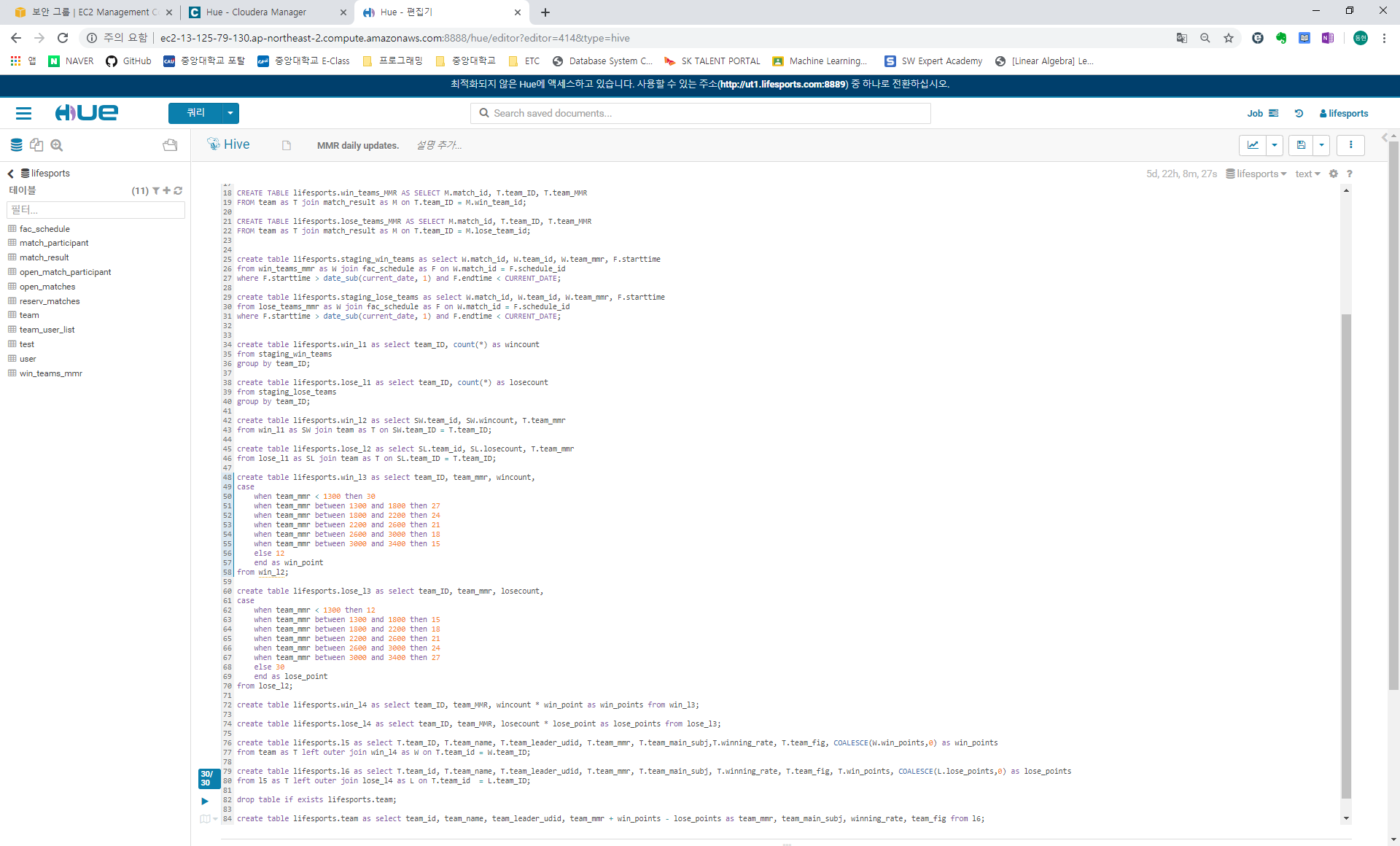
As you execute the workflow, you can see the screen like below which the actions for Sqoop and Hive are executed properly.



Completion of the workflow can be shown through Hue job browser.



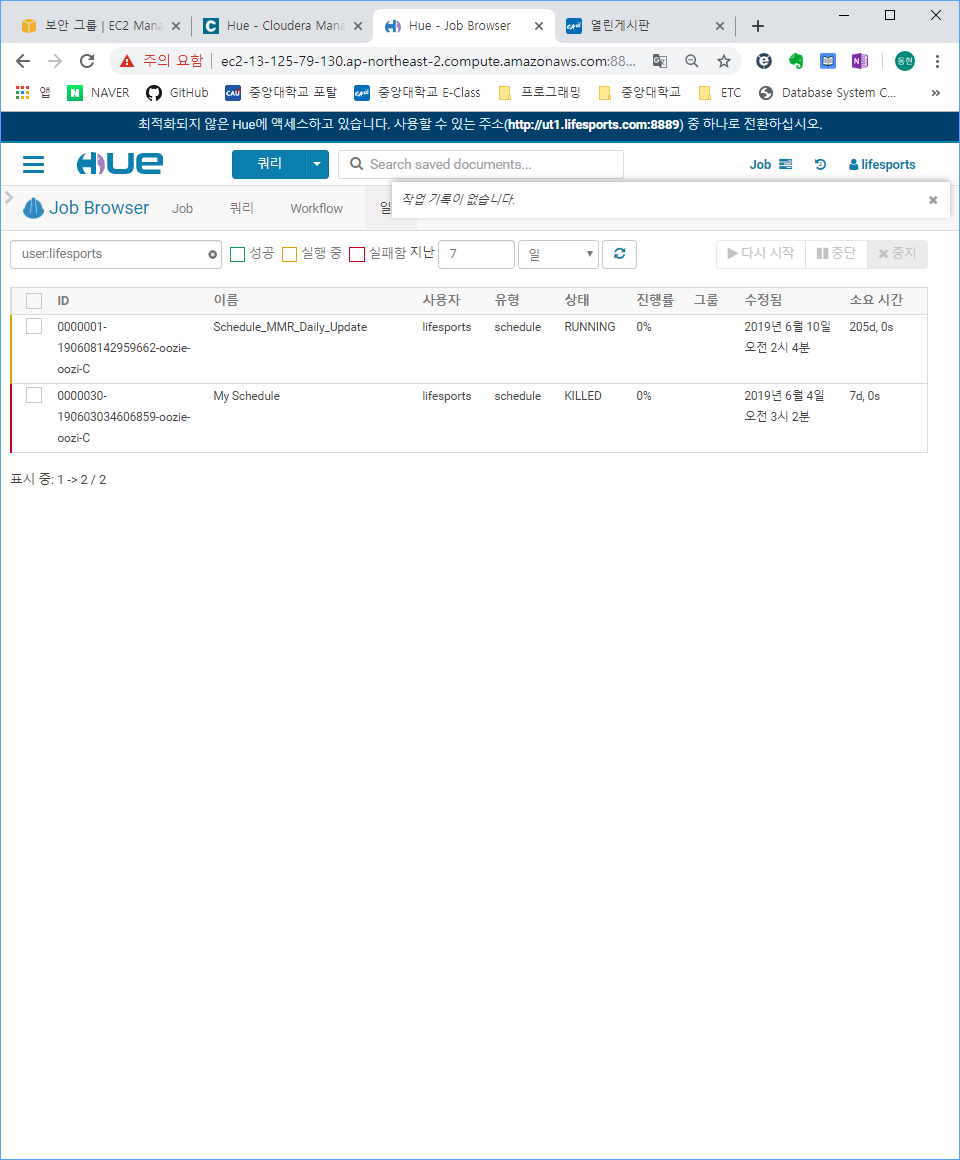
These are the stored queries for daily updating MMR. Calculate the MMR based on users’ tier, match results for open matches and reserved matches and user’s winning rate.



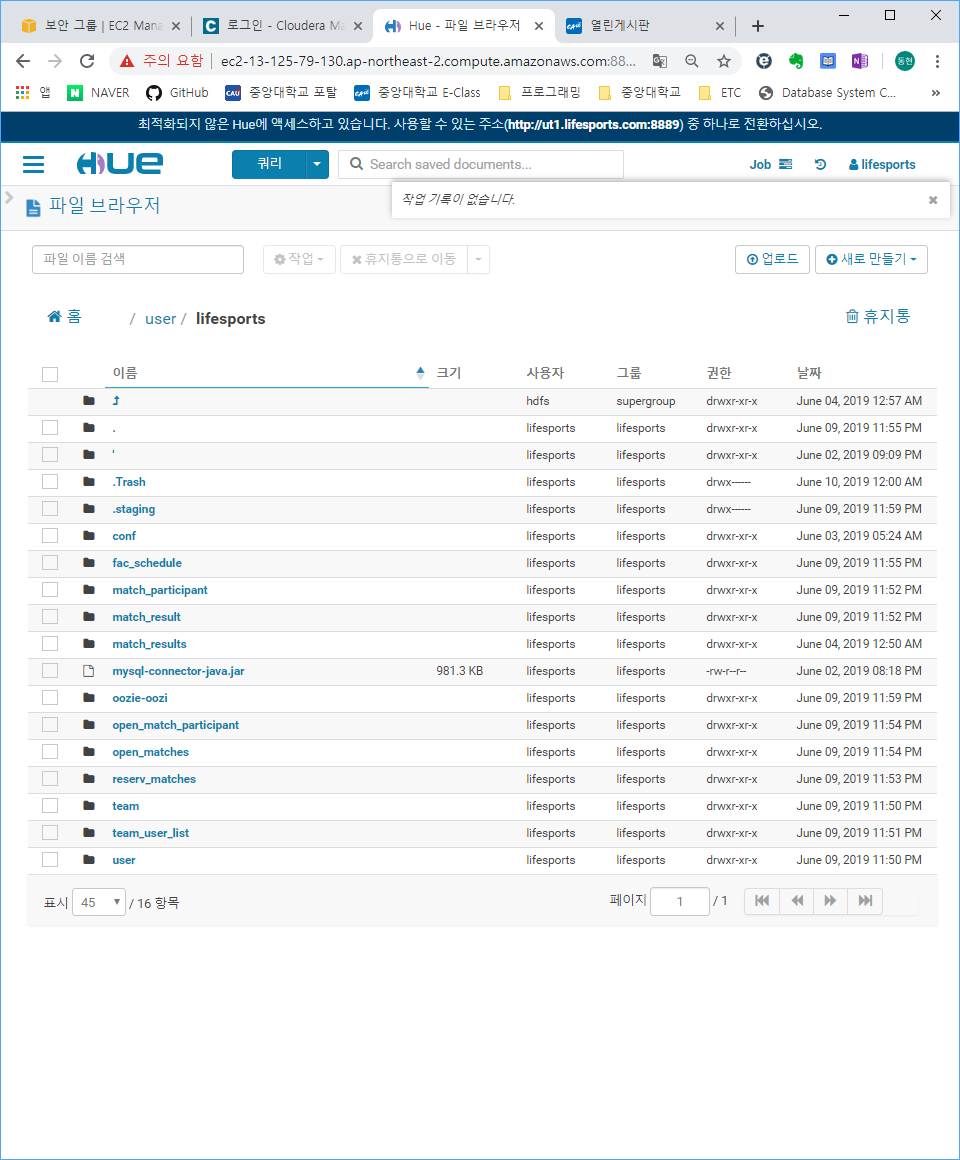
Using the workflow made before, create the Oozie scheduler so that the MMR and winning rate could be updated during 2 to 4 am, every day.



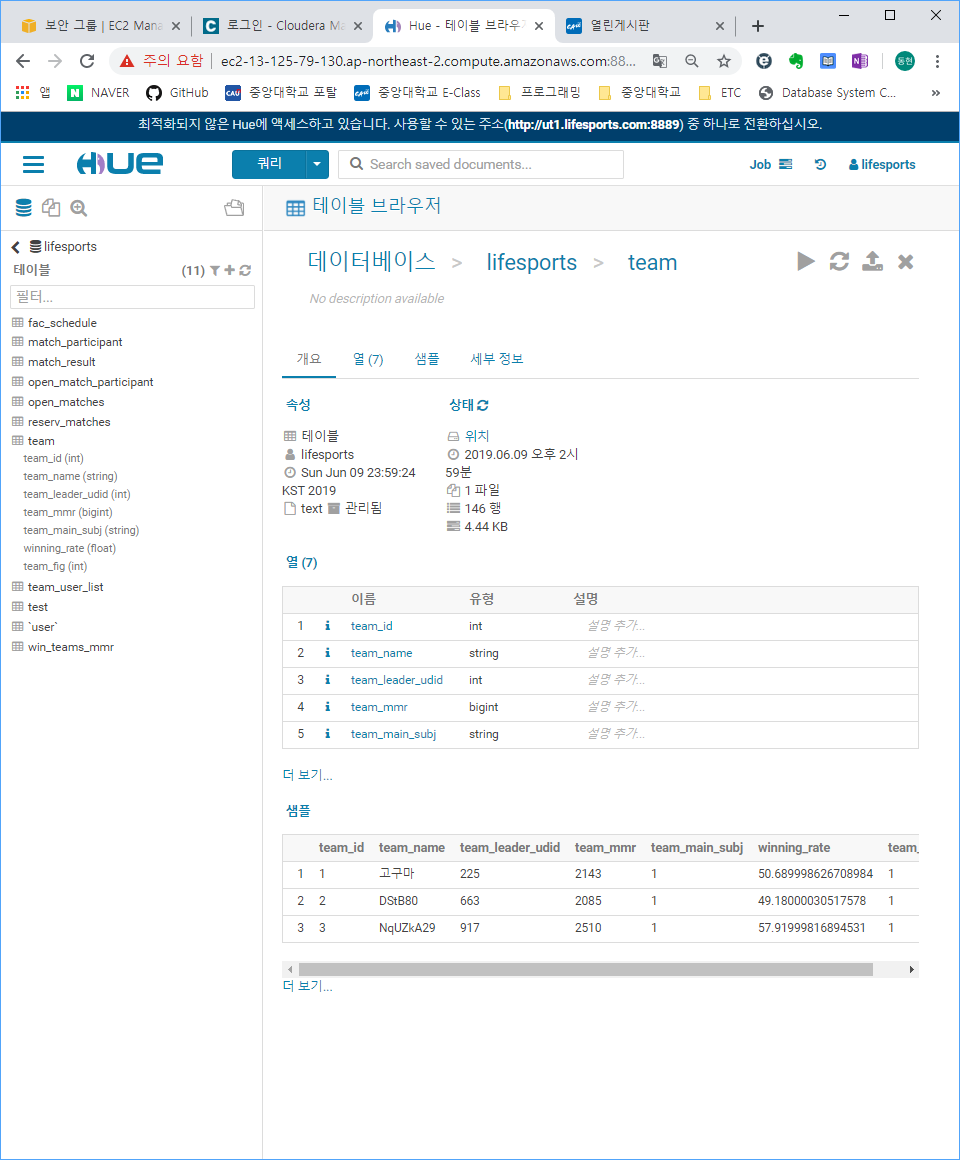
You can see the schedule is running.



You may see the directories for each tables in RDB are generated.



You may see the tables are generated properly through Hue table browser.



You may see the Oozie job has been terminated successfully through Oozie server Web UI.

