CSCI 3100 Software Engineering

Project Final Report

Group 20

Go-Sports

CHAN Kong Kuen	1155063944
CHEUNG Tsun Ming	1155062384
FAN Yan Lam	1155064685
NG Kit Shing	1155062400
YIU Ka Ming	1155077602

Department of Computer Science and Engineering

The Chinese University of Hong Kong

Document version number: 2

Printing date: 4/5/2018

Contents

1 Introduction

1.1 Project Overview

"Go-Sports" is a web-based application software that facilitates sports activities in the community. It provides a user-friendly platform for sports-lovers to manage sports activities. Functions of the platform include: searching desired sports facilities, planning schedule of upcoming games, forming teams and calling for random games with other users. Our product will cover multiple functions in the single application, this can provide great convenience to sports lovers, possibly attract people currently not participating sports activities, and cooperate with sports organizations for promotion of sports life. The group member and work assignment list is as followed:

Name	SID	Work assignment
CHAN Kong Kuen	1155063944	• Front-end development (HTML layout design) • Product testing
CHEUNG Tsun Ming	1155062384	• HTML functional coding • Back-end development (appointment, chat)
FAN Yan Lam	1155064685	• Code version control and integration • Database management
NG Kit Shing	1155062400	• HTML functional coding • Back-end development (registration, search, facilities and sportsfields)
YIU Ka Ming	1155077602	• Front-end development (interface design) • Product testing

1.2 Objective

The main aim of our product is to create a centralized, up-to-date database of sports facility information with user-friendly interface and comprehensive searching functions. Personal user can search for information of sports facilities with ease, and a great range of constraints can be applied to the search to have more accurate, suitable results. In addition, user can give ratings and comments to sports facilities, which can be some extra useful information for other users in choosing facilities. With the aforementioned features, the user experience is greatly enhanced.

Our design highly emphasizes the interaction of users and the sports community. Through appointment system and in-site communication, this offers opportunities for users not knowing each other but having common sports interest to meet.

1.3 Highlights

A major component of the product is a database of sports facilities and a search engine allowing a wide range of type of constraints, such as location, rating, fare. In pursuit of yielding better search result, the product incorporates Google Map for obtaining parametrized measures of location, and includes information or feedback by other users. In order to provide information as updated as possible, the data retrieval system would get information from other websites in constant time interval.

The interaction of users is another stress of our design, multiple interaction functions are implemented to the system, including giving like, commenting and rating of facilities, messaging between users, finding sports partners.

1.4 Project Statistics

The following tables show the number of LOC (line of codes) and McCabe's number of each of the functions in our software.

Back-end files

Function	LOC (lines of code)	McCabe's Number
Registration	112	16
Log on	32	6
Profile	133	16
Facility	99	6
Sportsfield	159	12
Search	81	12
Appointment	102	12
Chat room	62	8

HTML files

File name	LOC (lines of code)	McCabe's Number
account_activation_email.html	46	1
account_activation_invalid.html	46	1
account_activation_sent.html	42	1
advanced_search.html	47	2
appointment_join.html	79	9
appointment_join_result.html	67	9
appointment_list.html	82	11
appointment_new.html	52	2
base.html	121	3
chat.html	170	5
comment.html	55	3
comment_result.html	53	3
facility_detail.html	80	3
facility_list.html	72	5
homepage.html	48	1
logoff_result.html	14	1
logon.html	43	2
logon_result.html	33	5
profile.html	59	4
profile_edit.html	49	3
profile_edit_result.html	31	5
registration.html	52	3
registration_result.html	33	5
search_result.html	70	5
sportsfield_detail.html	72	5
sportsfield_list.html	69	5

2 System Architecture Design by DFD

2.1 Architecture Design

The figure below illustrated the system architecture design of our software.

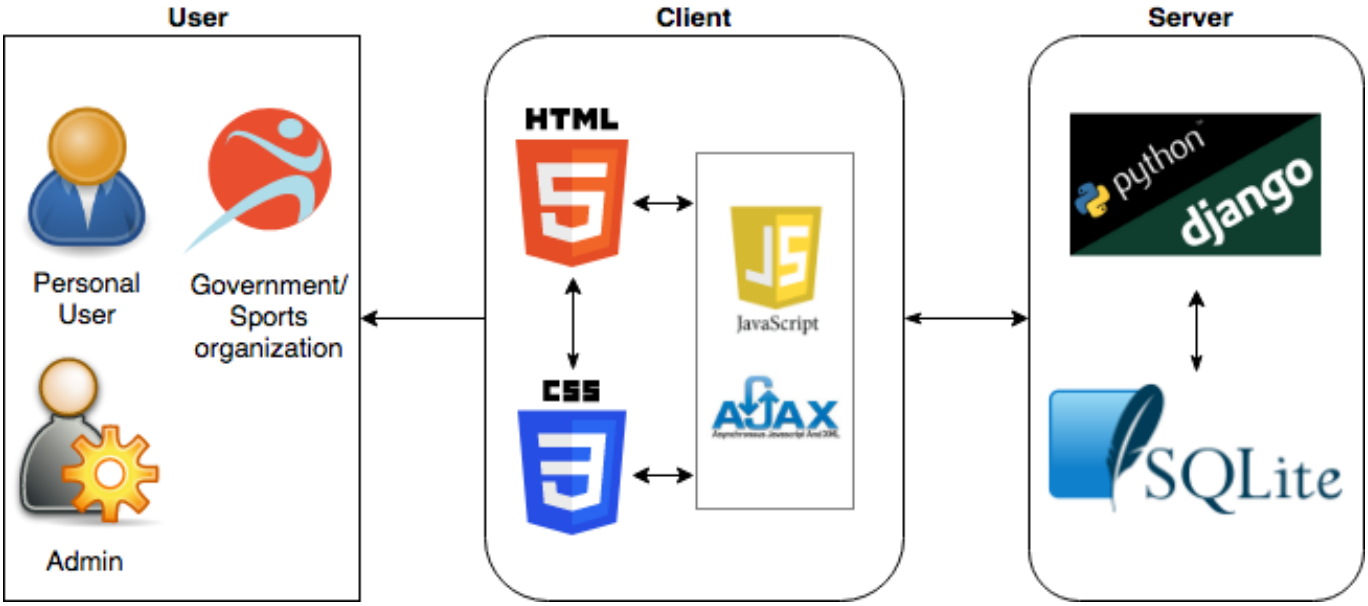


Figure 2.1.1: Architecture Diagram

2.2 System Components

Hardware	Software	Network and Internet Service
• Server	• Software solution	• Internet
• Database	• SQLite	• Google Map API
• PC	• Web server (Django)	
	• Latest web browser	

Description

Database
Our data come from different sources, some from government and some from user suggestion. The scale will keep increasing. We choose SQLite to handle various types of data.

Client software
HTML, CSS, Javascript and other framework will be applied to build a user friendly and interactive website. Any browser compatible with the above applications will be suitable for "Go-Sport", **but the application work best with Google Chrome**.

Server software
Django will be applied to implement the web server.It provides asynchronous, event driven I/O APIs to handle concurrent request.

2.3 DFDs

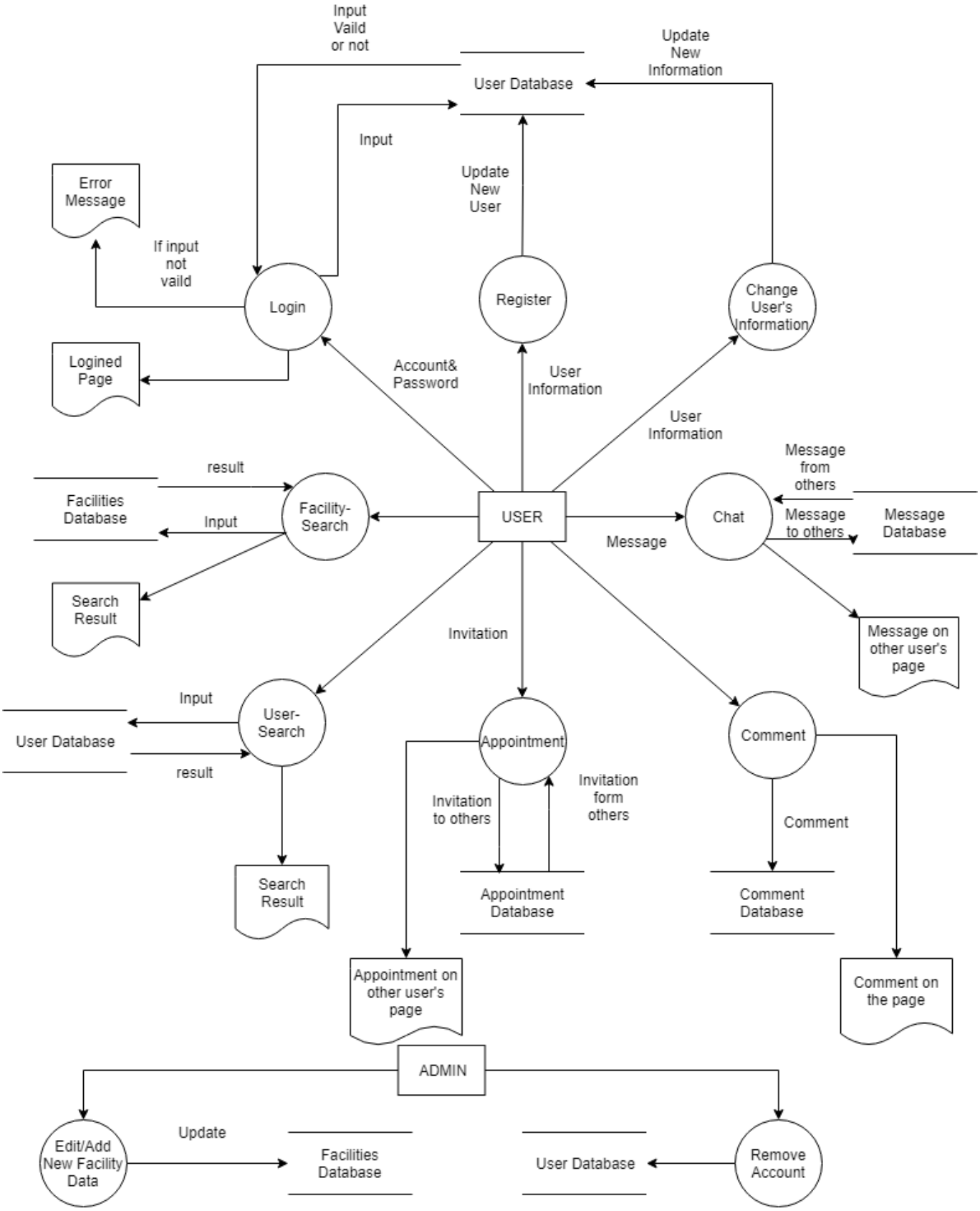


Figure 2.3.1: Data Flow Diagram

3 Detailed Description of Components by UML

3.1 Overall class diagram

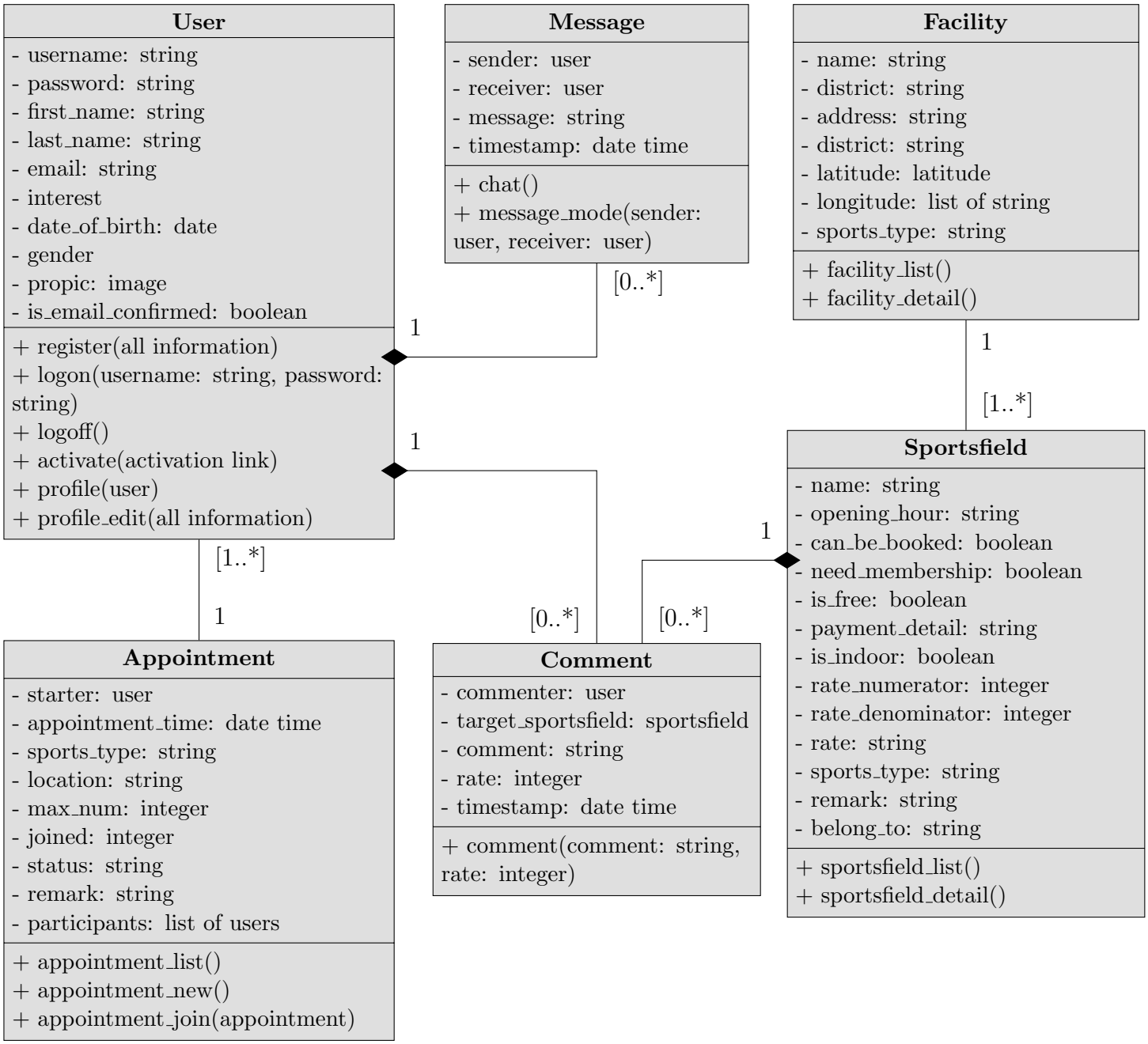


Figure 3.1.1: Overall Class Diagram

3.2 Use Case Diagram

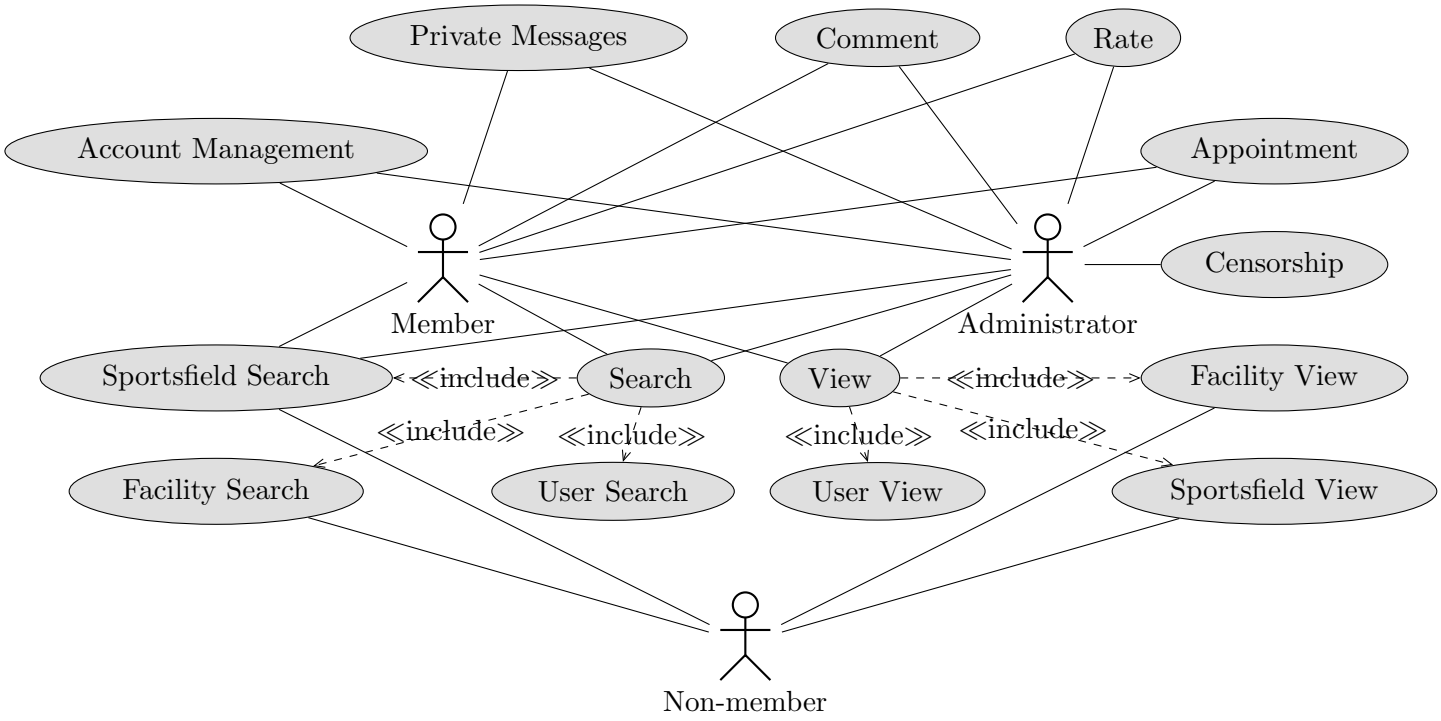


Figure 3.2.1: Use Case Diagram

3.3 Sequence Diagrams

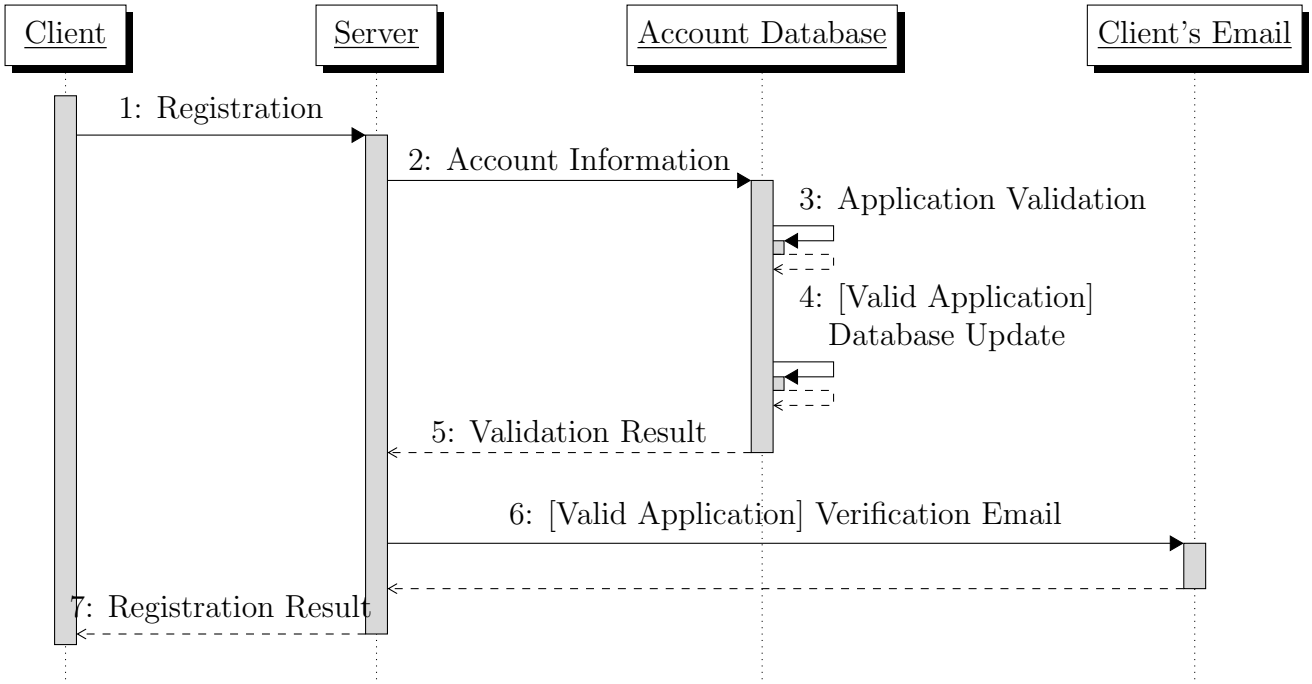


Figure 3.3.1: Sequence Diagram of Registration

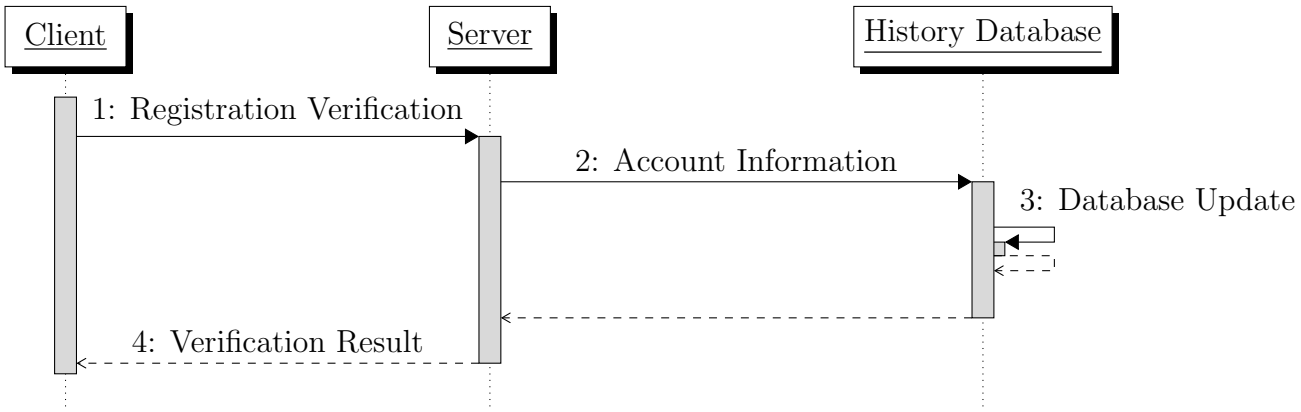


Figure 3.3.2: Sequence Diagram of Registration Verification

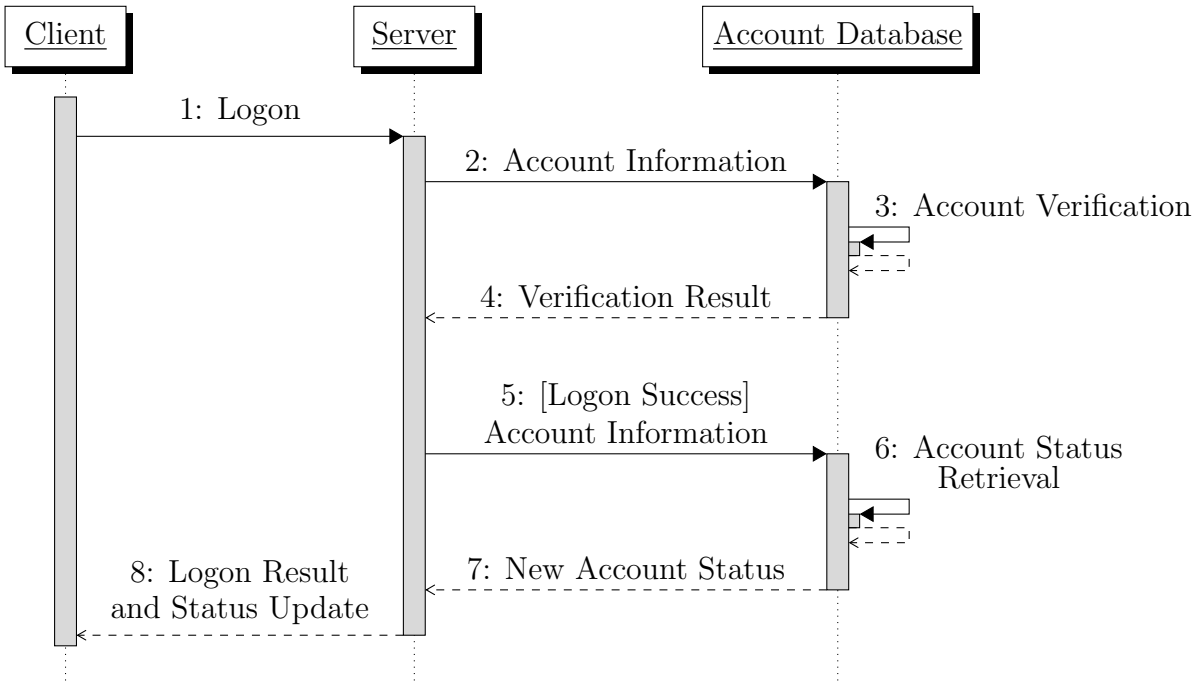


Figure 3.3.3: Sequence Diagram of Logon

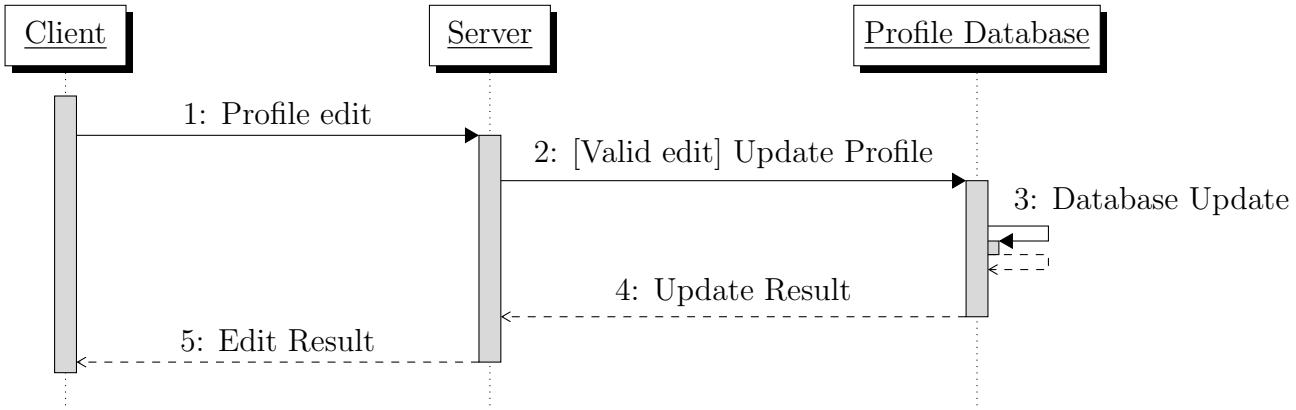


Figure 3.3.4: Sequence Diagram of Profile Edit

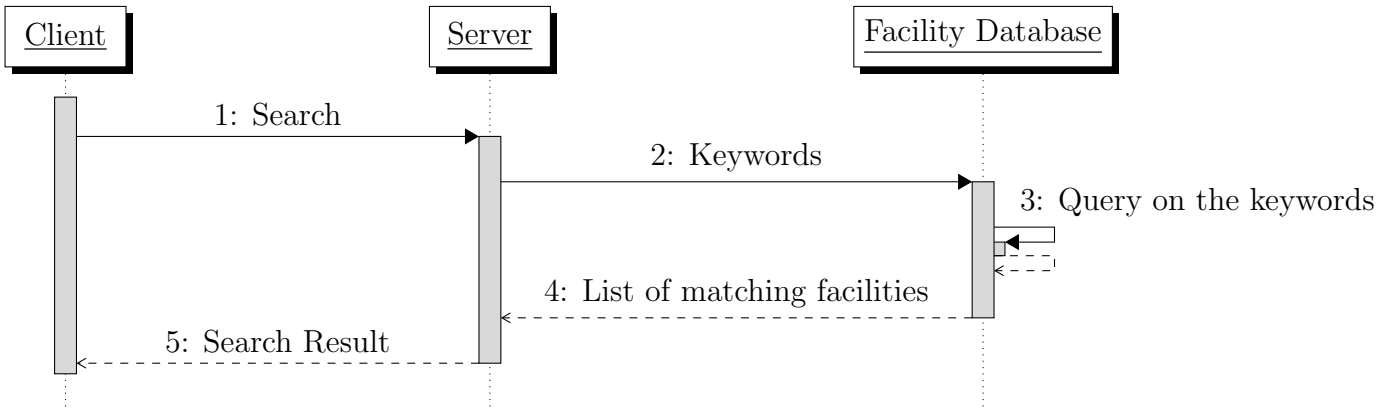


Figure 3.3.5: Sequence Diagram of Facility Search

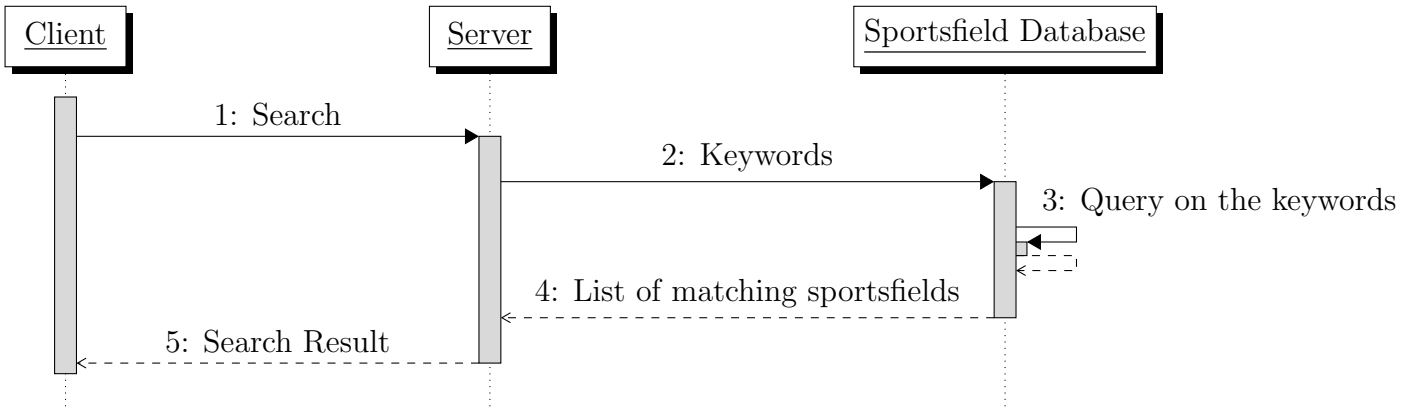


Figure 3.3.6: Sequence Diagram of Sportsfield Search

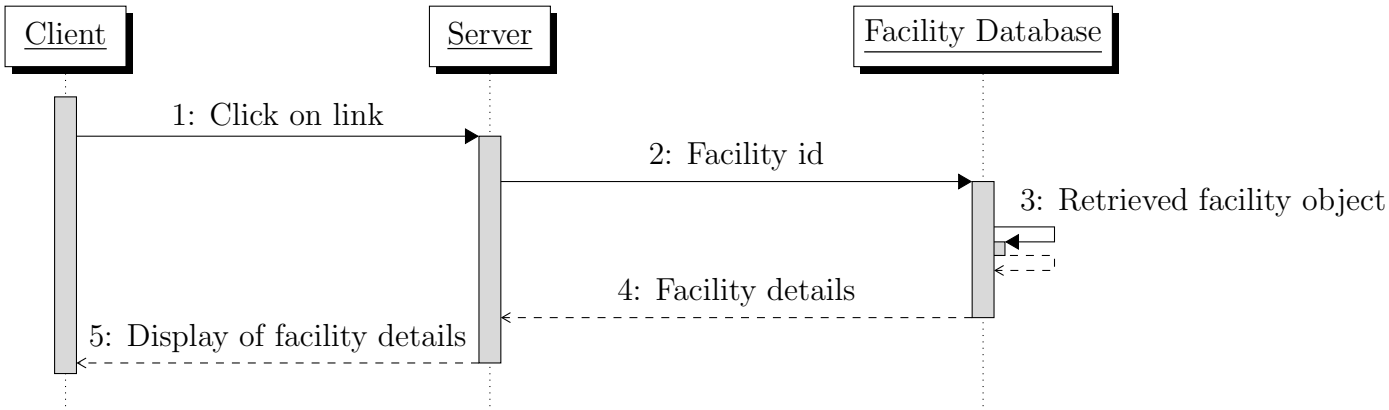


Figure 3.3.7: Sequence Diagram of Facility Detail Display

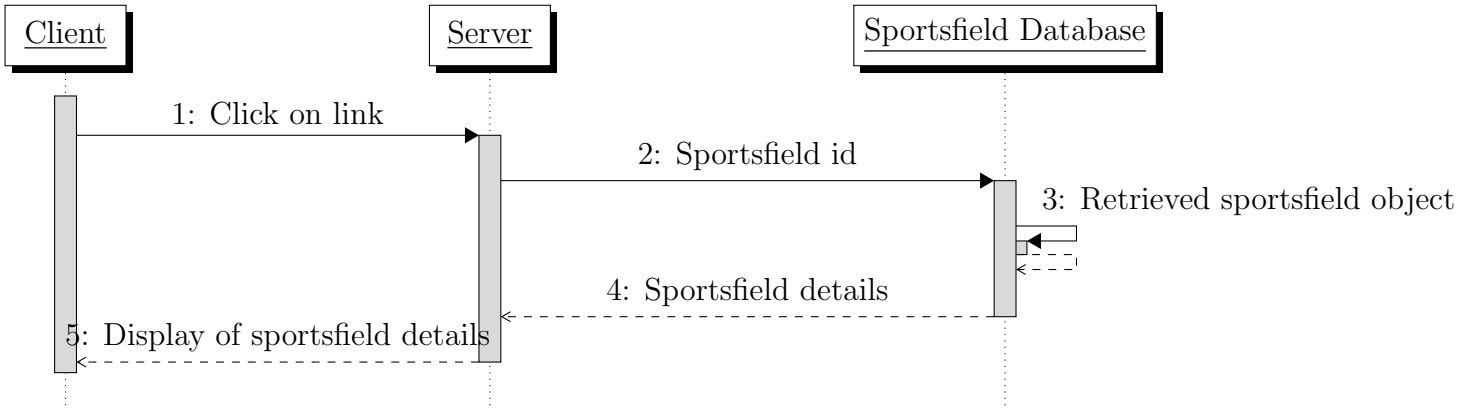


Figure 3.3.8: Sequence Diagram of Sportsfield Detail Display

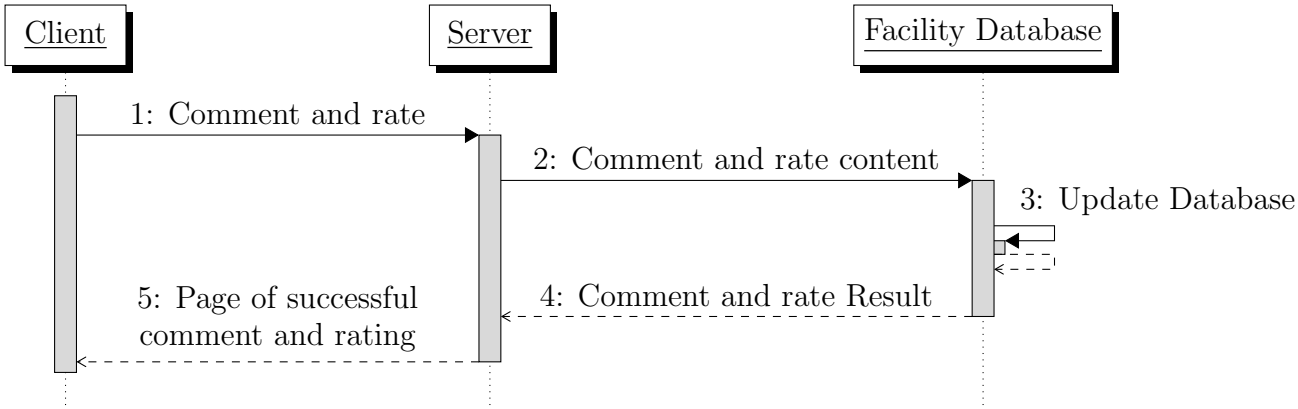


Figure 3.3.9: Sequence Diagram of Comment and Rating

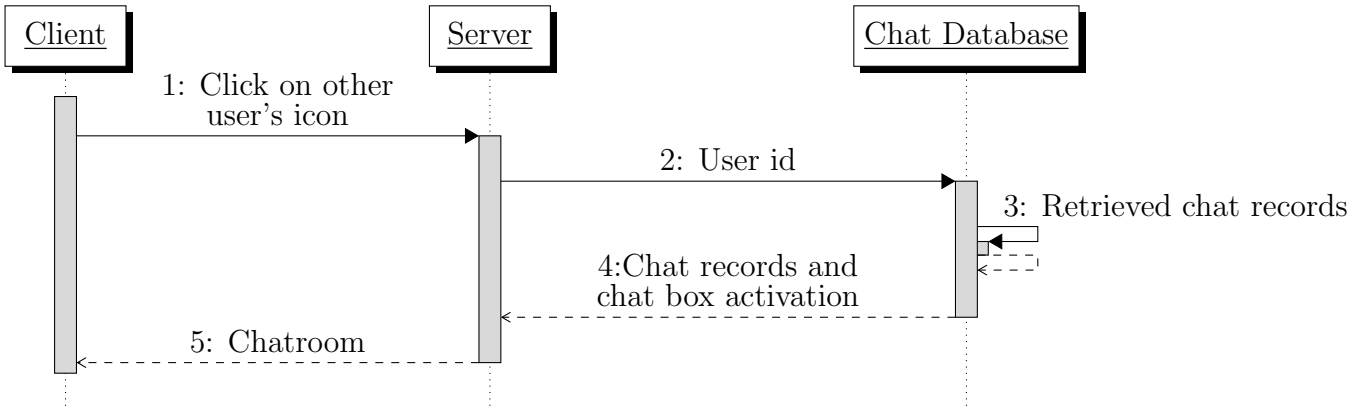


Figure 3.3.10: Sequence Diagram of Selection of Chatroom

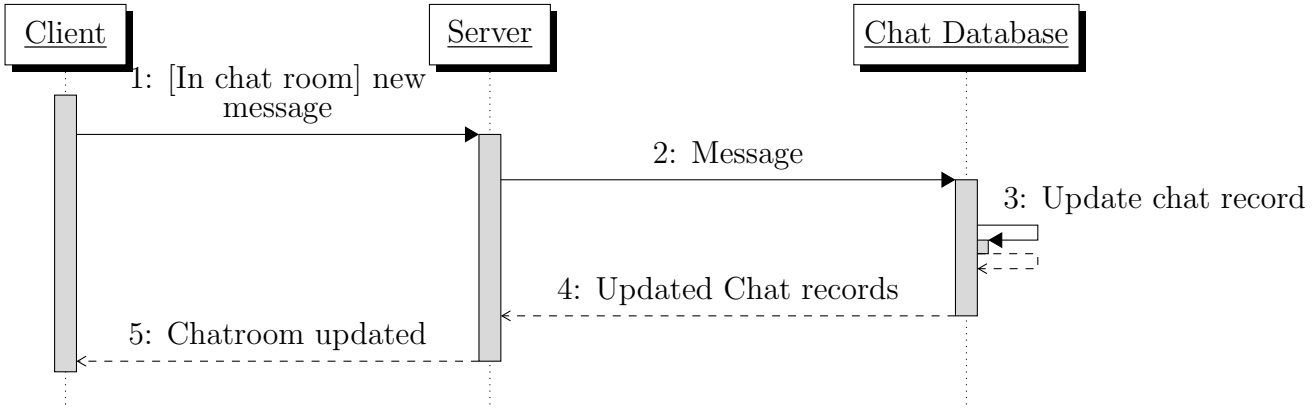


Figure 3.3.11: Sequence Diagram of New Message

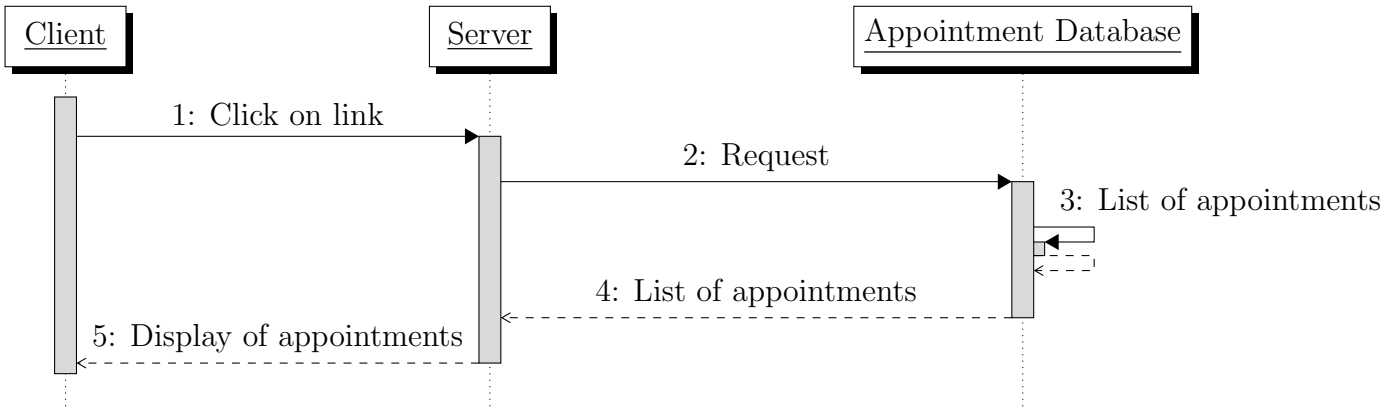


Figure 3.3.12: Sequence Diagram of Appointment Display

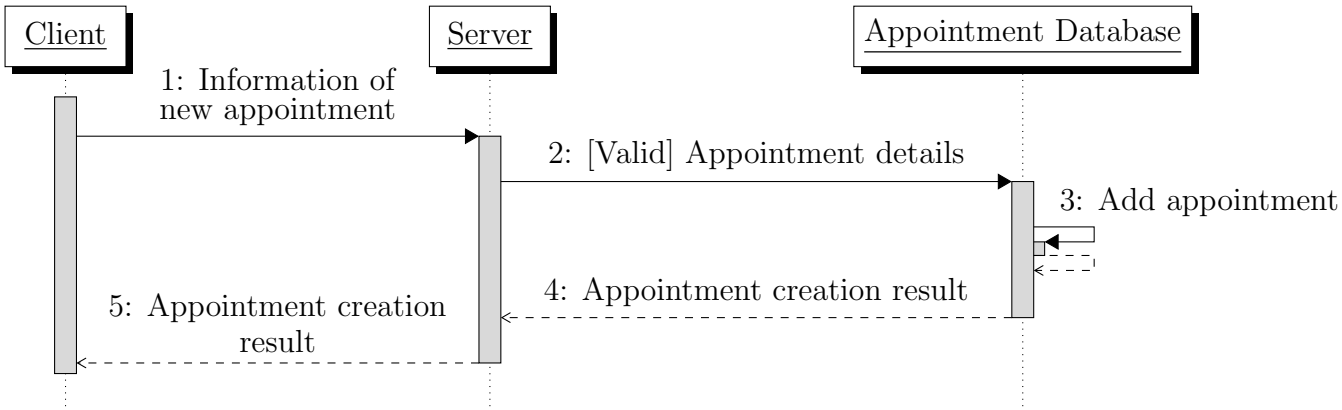


Figure 3.3.13: Sequence Diagram of Appointment Creation

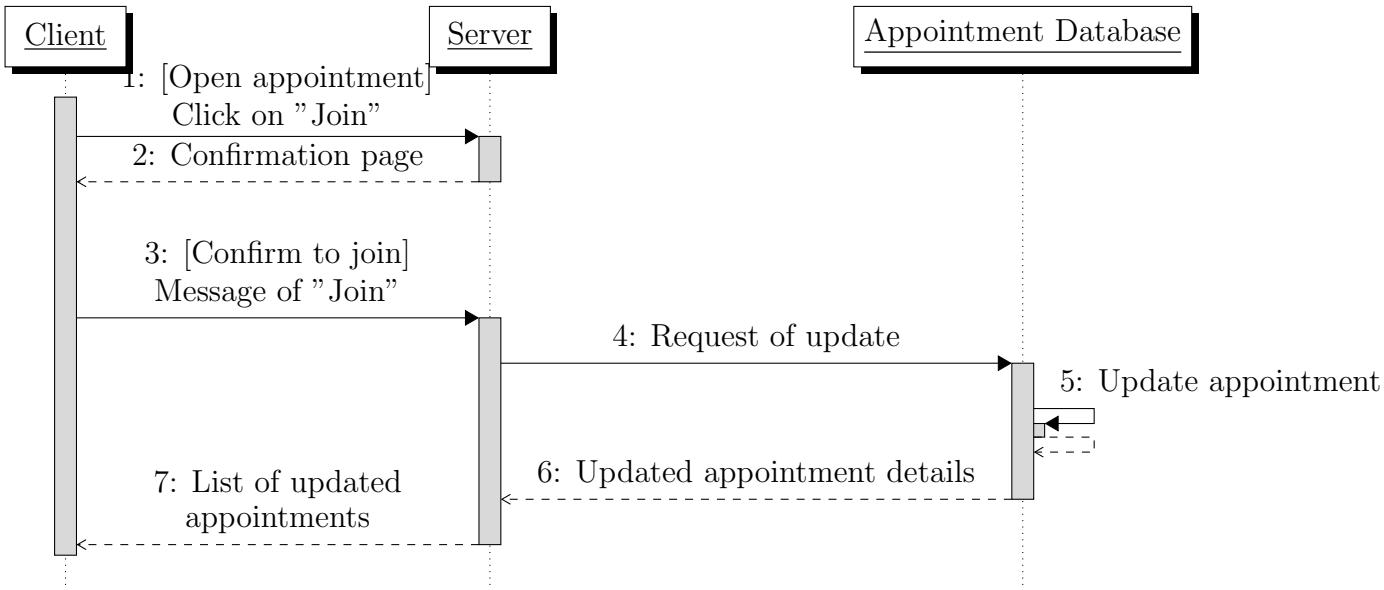


Figure 3.3.14: Sequence Diagram of Joining appointment

3.4 Functionality

The application is an interactive online platform for viewing sports facility information. Functionality of the application includes searching, browsing, appointment creation, and communication between users.

3.5 Procedures and functions

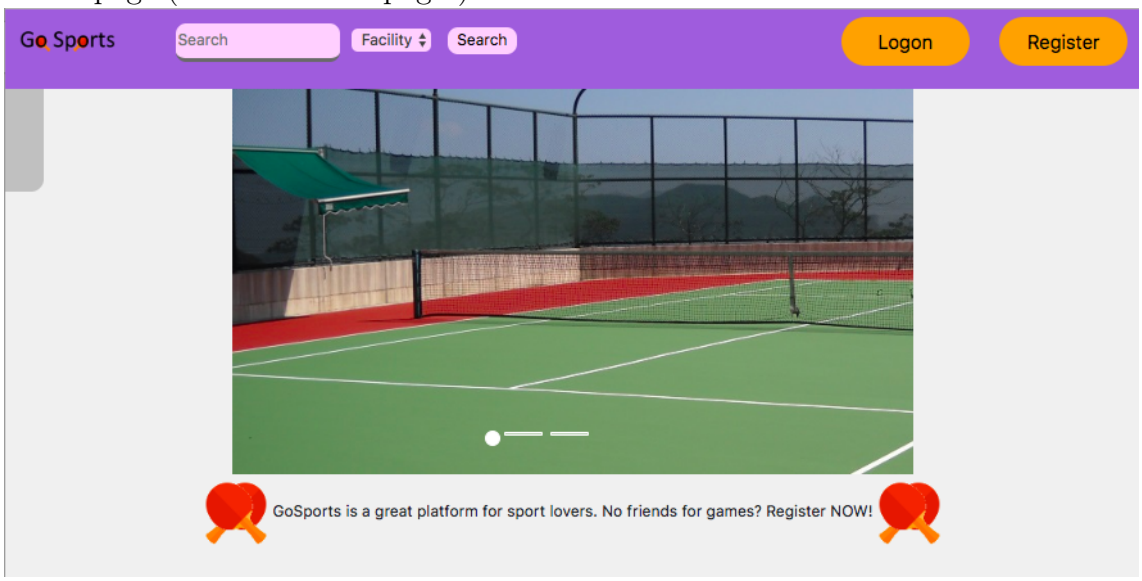
- Registration:
  To allow non-member to register account to get access to more functions provided to member exclusively.
- Log on:
  To allow registered user to log on for functions only available to members.
- Search:
  To allow searching by user: non-member can search for facilities while member can search for other users in addition.
- Browse:
  To allow browsing by user: non-member can browse for facilities and sportsfields while member can browse for other users in addition.
- Appointment:
  Users who have logged on can create, view and join appointments for sports activities.
- Chat room:
  Users who have logged on can use the personal chat room, where users can message with other members in real-time.

4 User Interface Design

4.1 Description of the User Interface

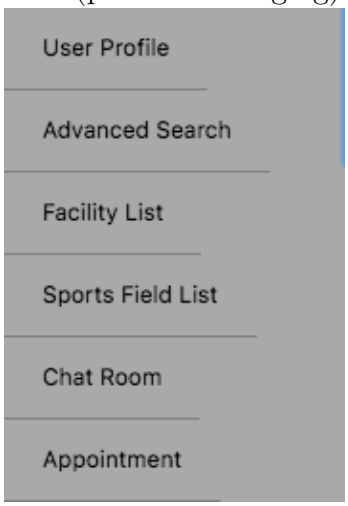
In this section, the user interface of our application is described. The major components of the user interface include:

- Homepage:
  The Homepage introduces our web application. It includes several images related to sports, represented by a slide bar. There is also a header bar at the top and a navigation bar on the left of the page (and also other pages).

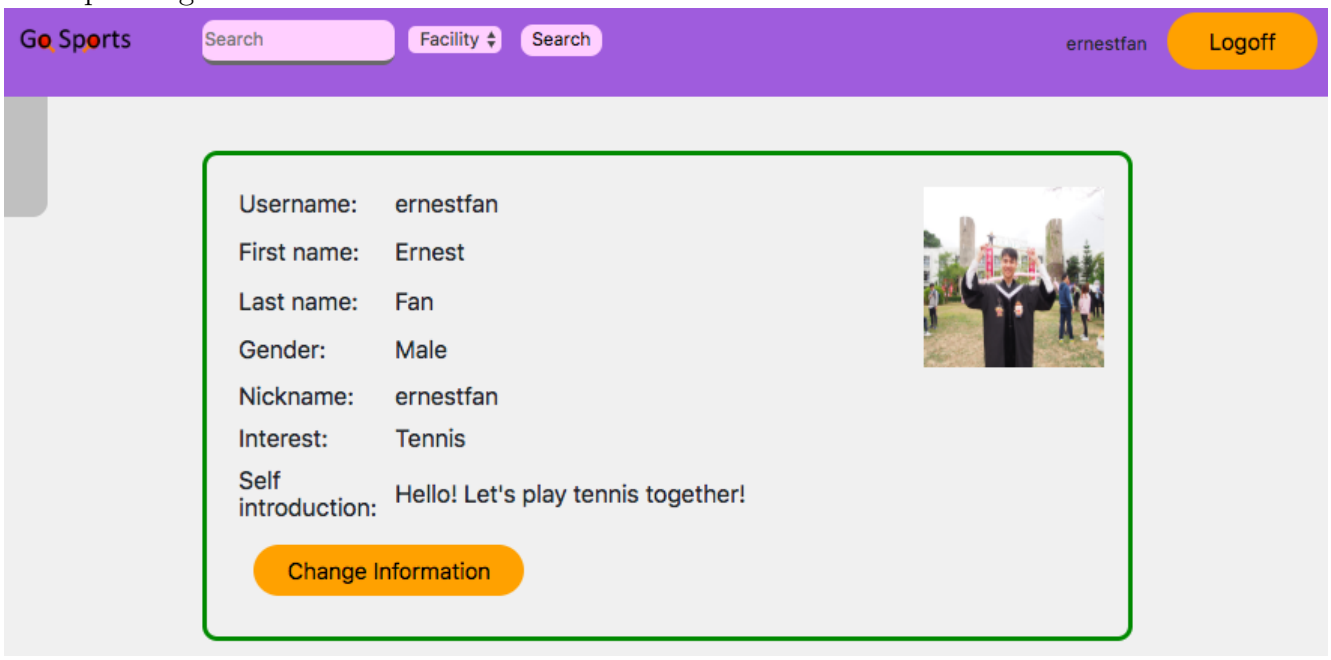


- Header bar:
  The header bar provide us the control of our log-in/log-out status, as well as a search function. Also, the icon on the left enable us to redirect to the home page immediately.

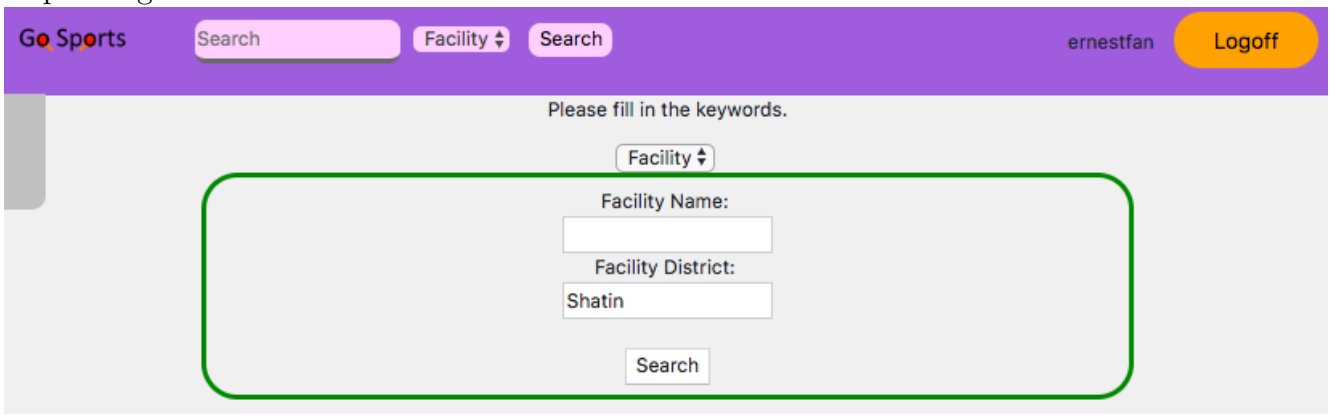
- Left navigation bar:
  The left navigation bar provide us the access of different major functions such as searching, chat room(private messaging) and viewing facilities and sportsfields.



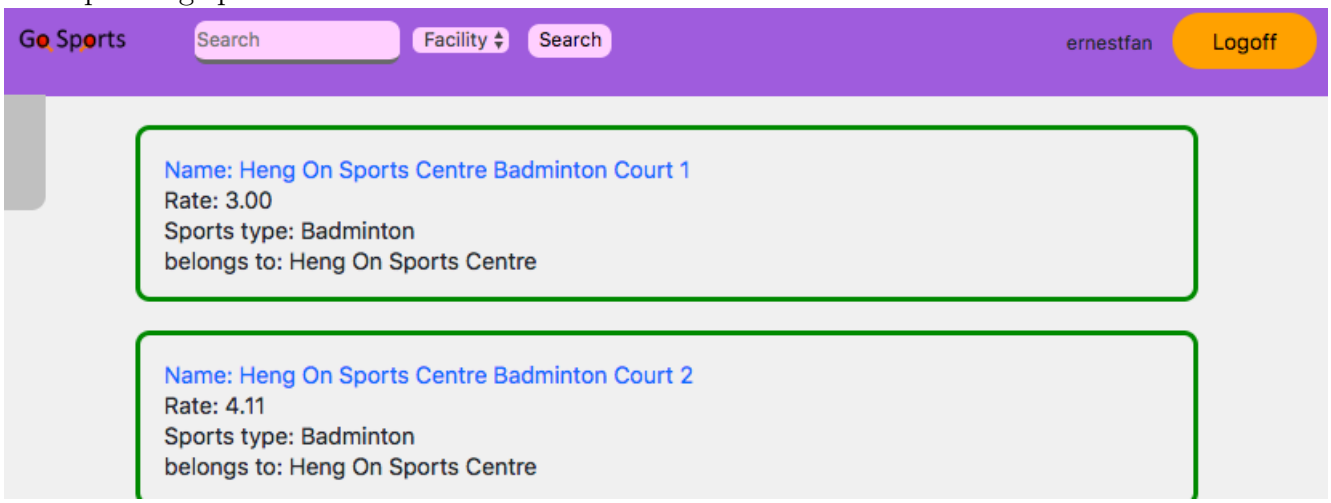
- User Profile:
  The user profile page contain the details of the user. It includes the profile picture, name and also the interested sports of the user. Users can also change personal information by clicking the corresponding button.

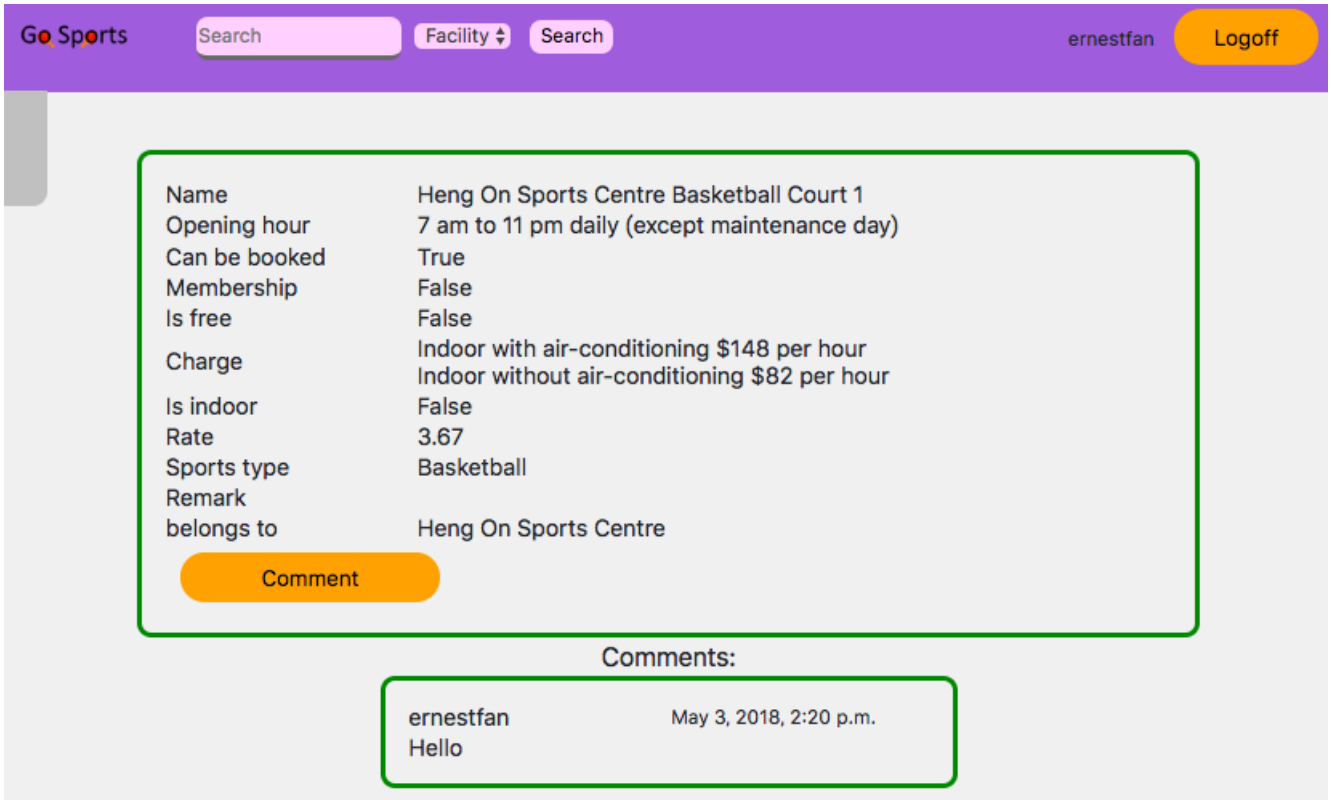


- Search page:
  The search page allow users to search for desired facilities and users. They can specify the corresponding attributes in the fields.

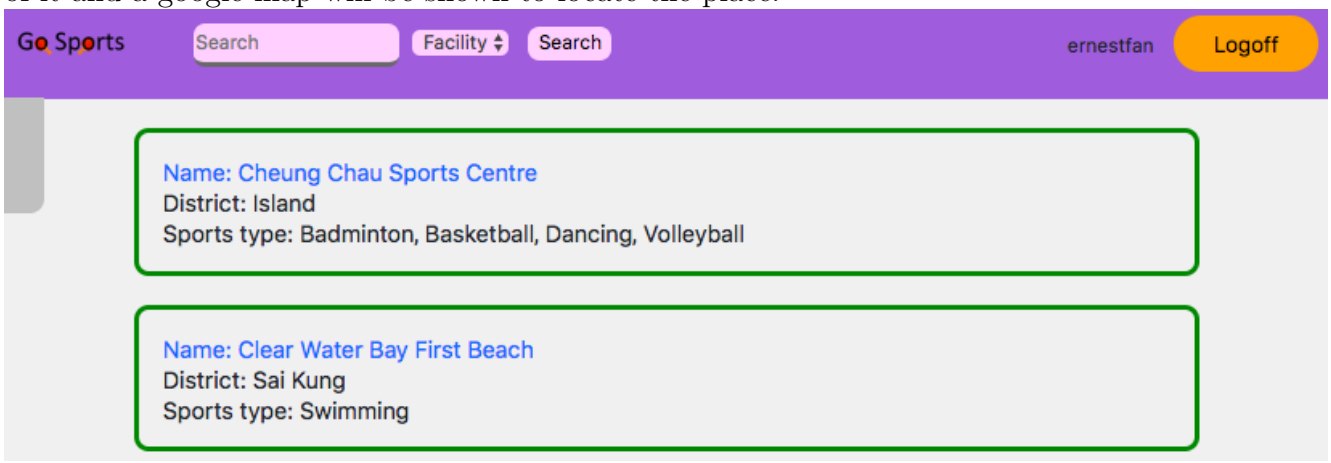


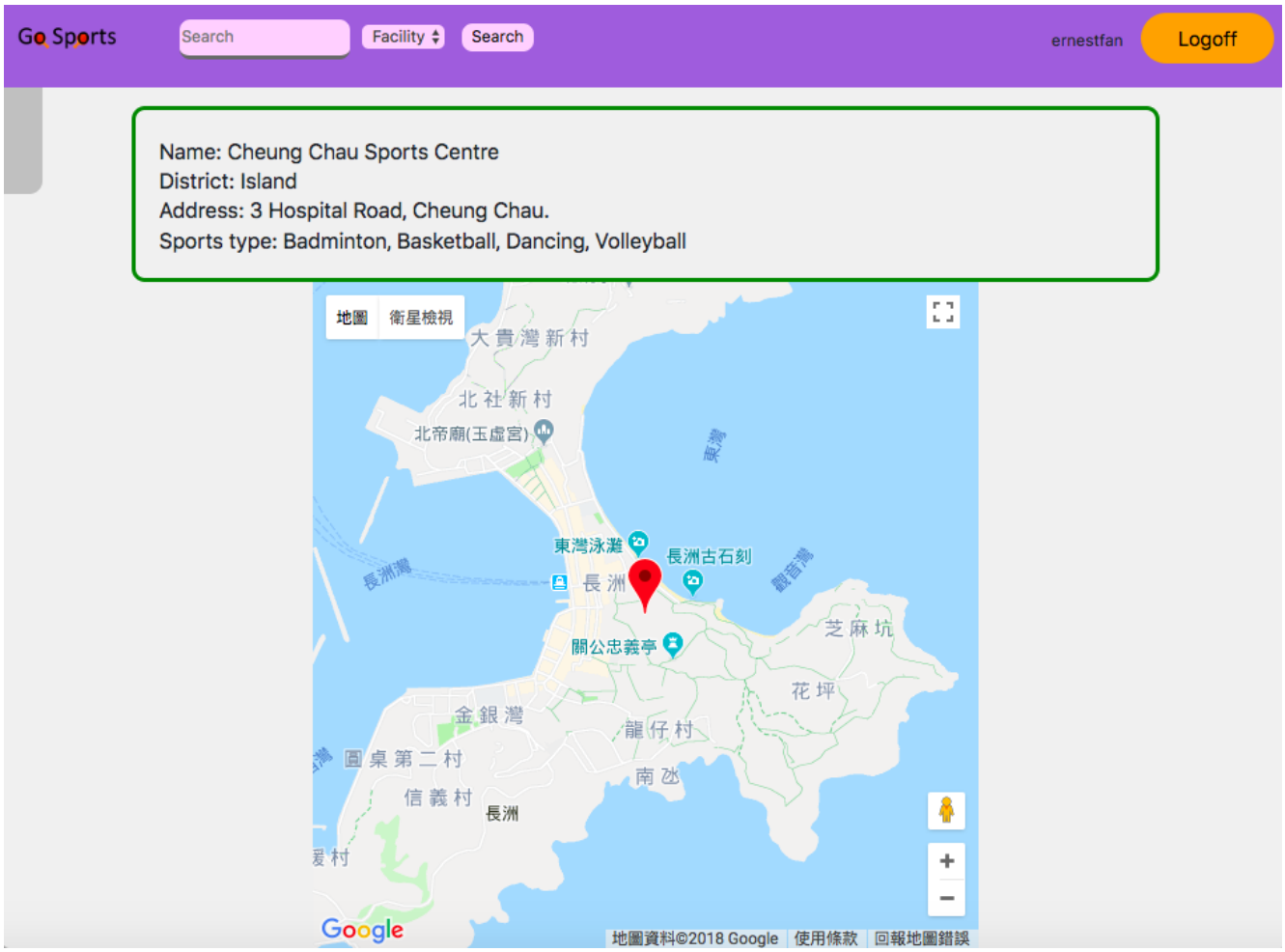
- Sportsfield:
  The sportsfield page lists all sportsfields in the database. When clicked into one of the facilities, details and comments of it will be shown. Users can also give comments and ratings to the corresponding sportsfield.



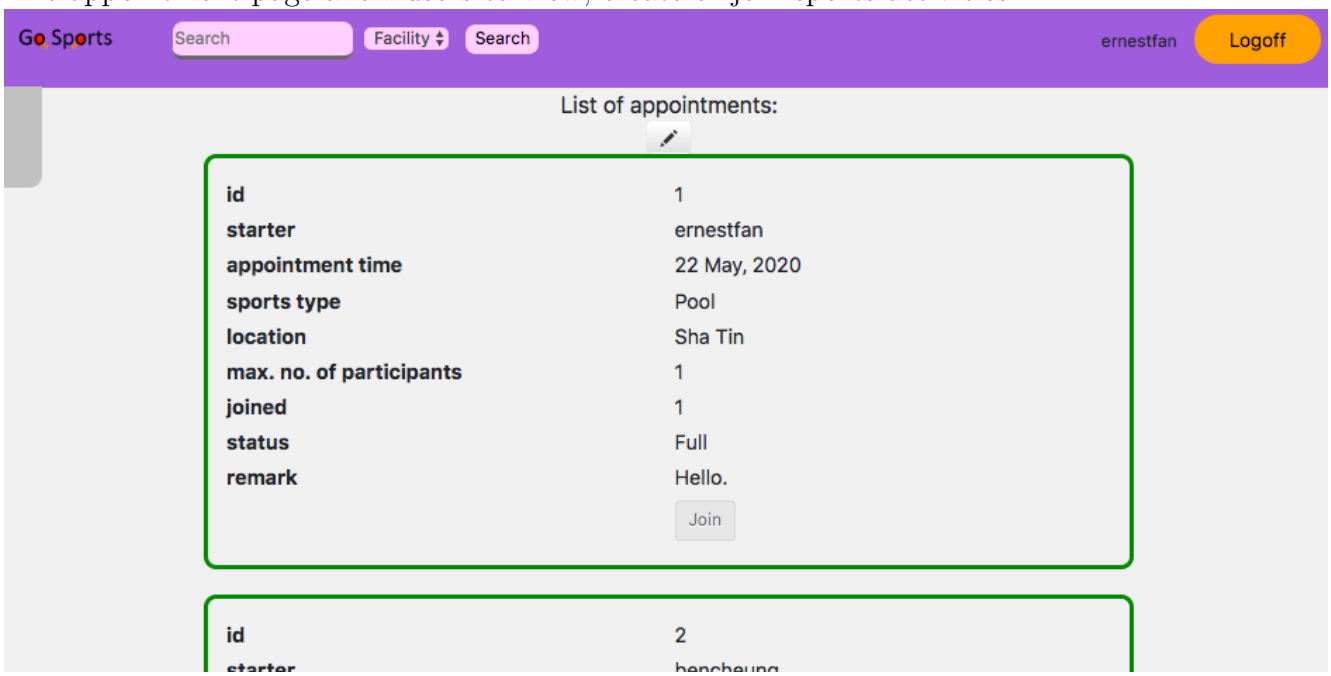


- Facility:
  The facility page lists all facilities in the database. When clicked into one of the facilities, details of it and a google map will be shown to locate the place.

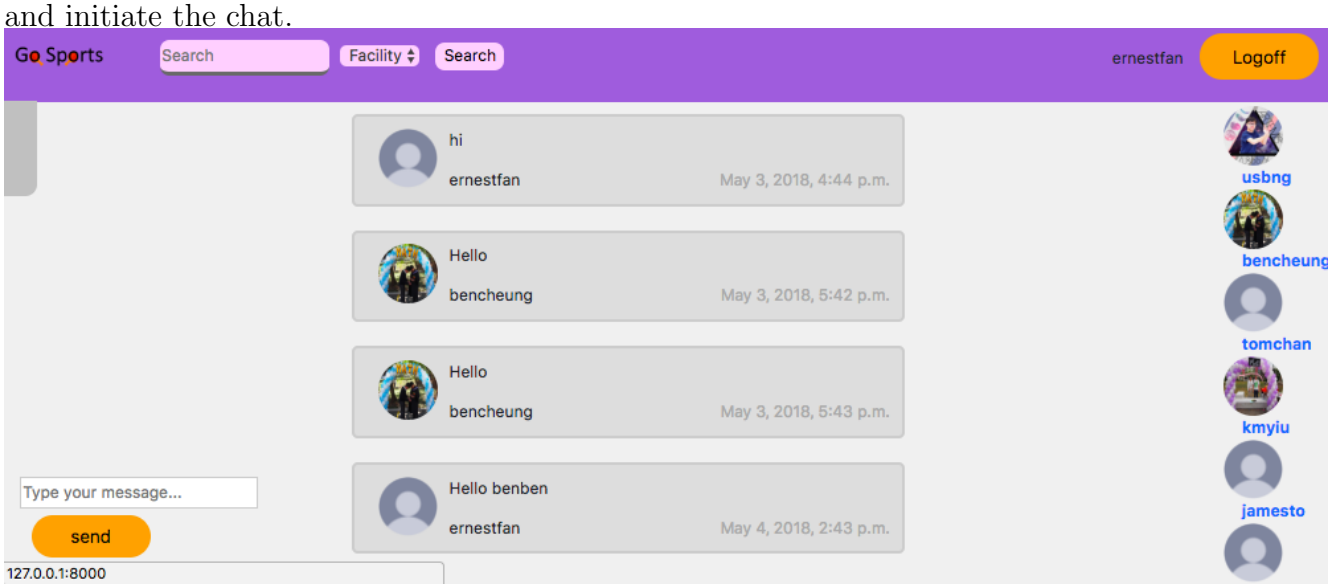




- Appointment:
  The appointment page allow users to view, create or join sports activities.



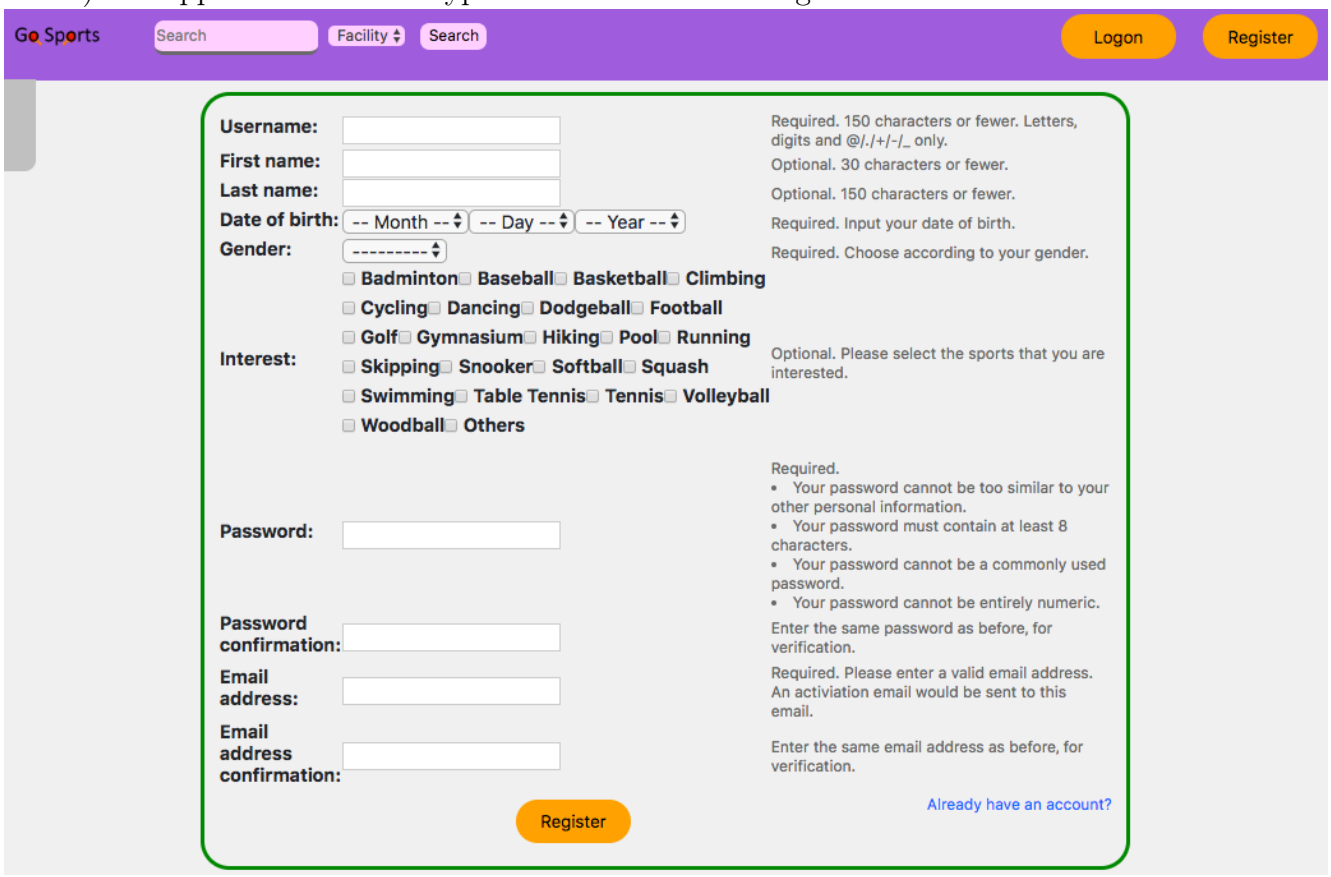
- Chat Room:
  The chat room enable users to communicate to each other. Users can choose the desired friends and initiate the chat.



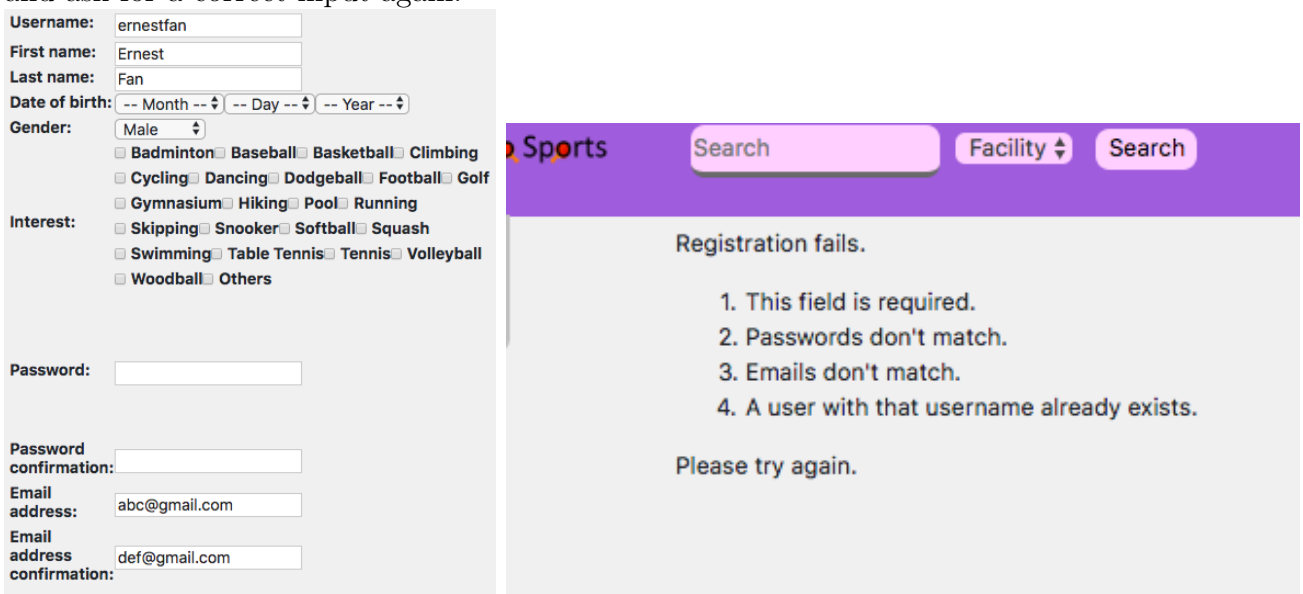
4.2 Objects and Actions with Screen Images

4.2.1 Registration

In the homepage, after the "Registration" button is clicked, the page with registration from (as shown below) will appeared. One can type in information and register for an account.



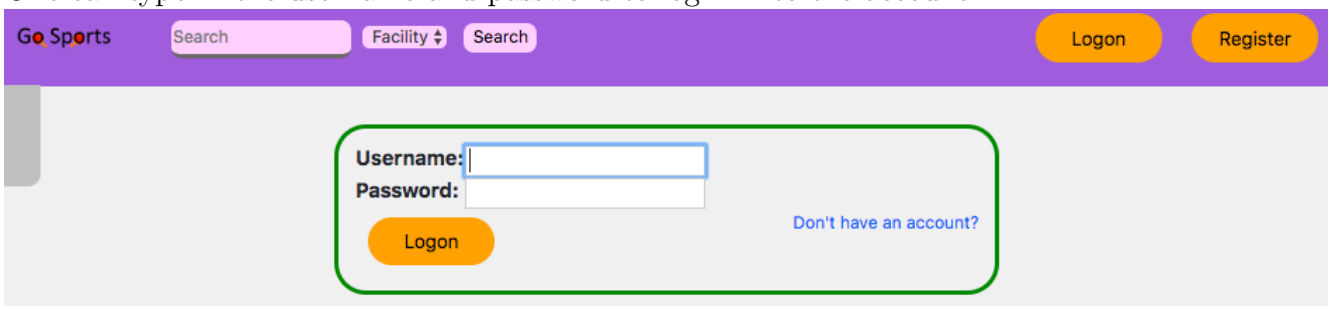
Once invalid information is submitted, our system will generate a list of error messages in a new page, and ask for a correct input again.



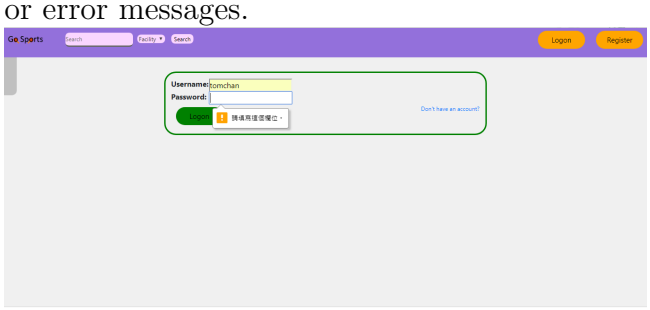
After a successful registration, a confirm email will be sent to the email address. In order to activate the account, users have to click the link given in the email.

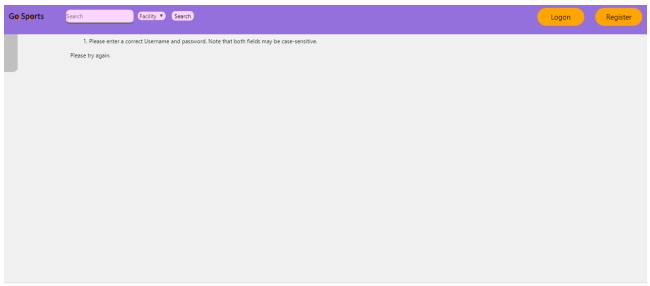
4.2.2 Log-in/Log-out

In the header bar, after the "Logon" button is clicked, the log-in page (as shown below) will appeared. One can type in the username and password to log-in into the account.

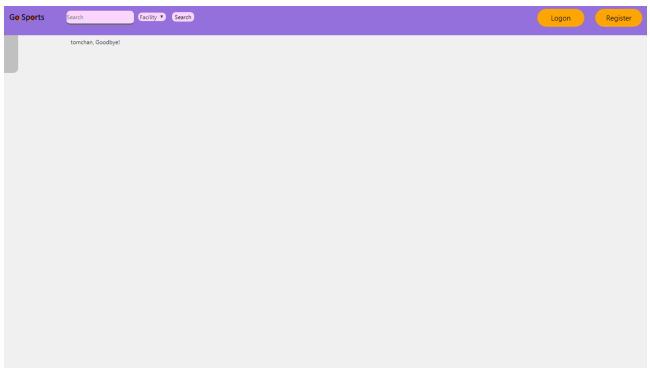


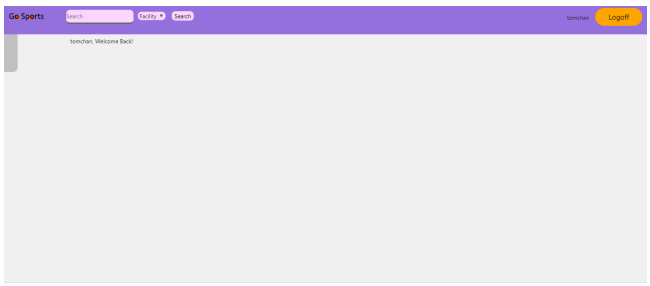
If wrong username/password is typed in, error messages will shown, either immediately or with a page or error messages.





If the input is correct, then a welcome message will be shown for a few seconds, and then it will automatically redirect to the homepage with a logged-in status. Similar good bye message will also be shown once the "Logoff" button is clicked.

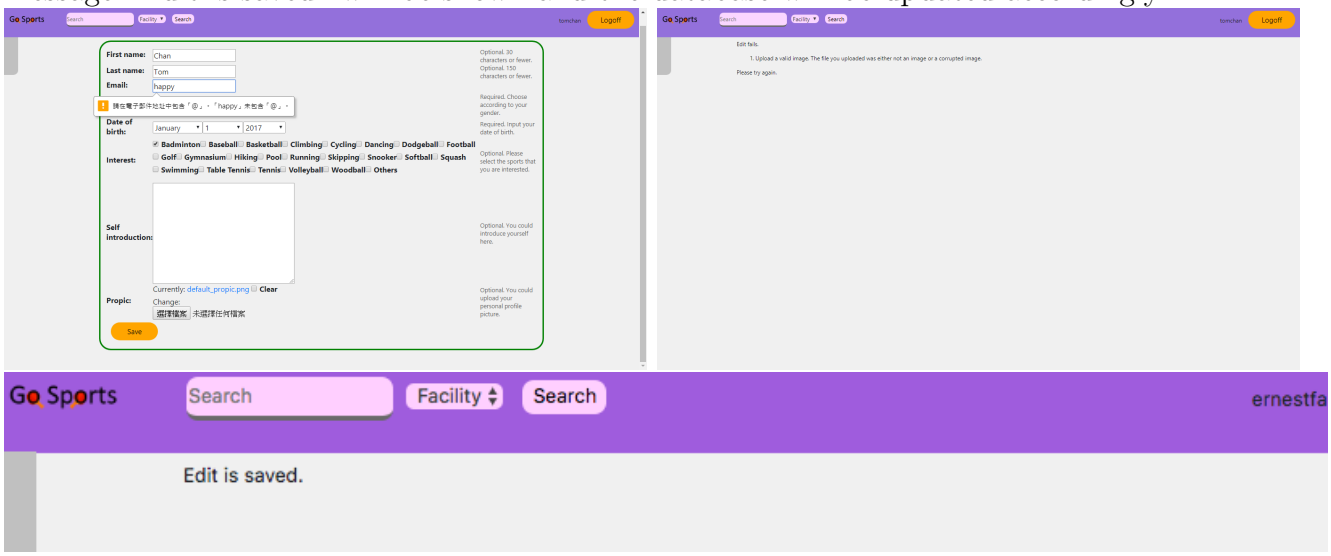




4.2.3 Editing Personal Information

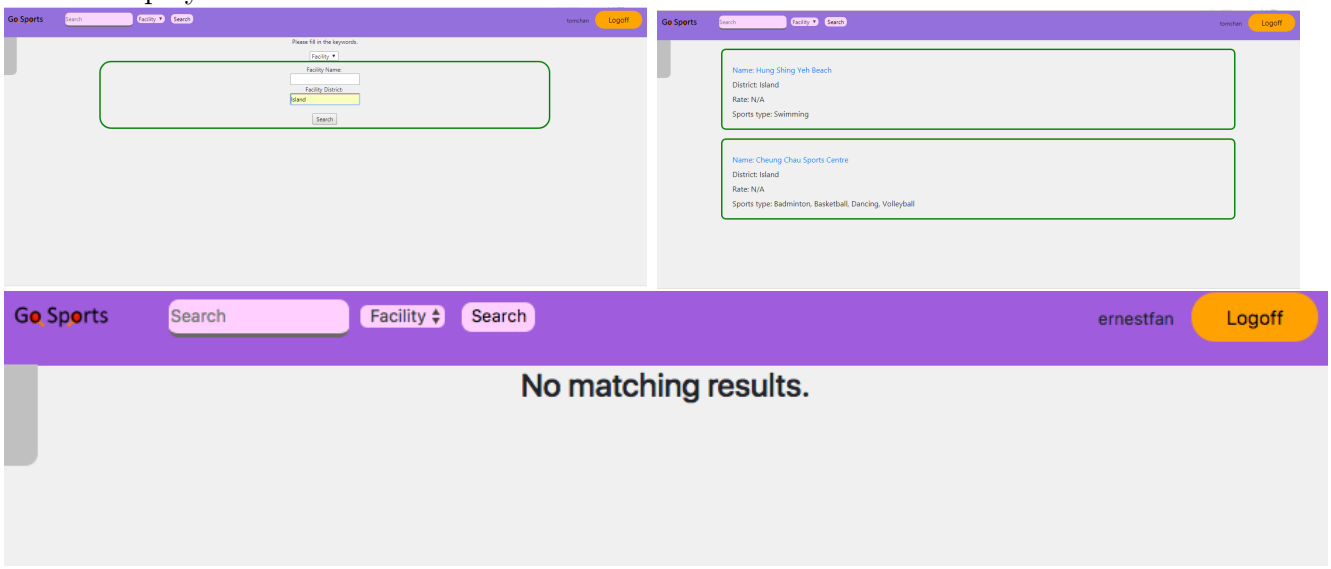
When user views his/her own personal information, they can edit their own personal information by clicking the "Change Information" button.

Once the form is submitted, error messages will be shown if there are invalid input. Otherwise, a message "Edit is saved" will be shown and the database will be updated accordingly.



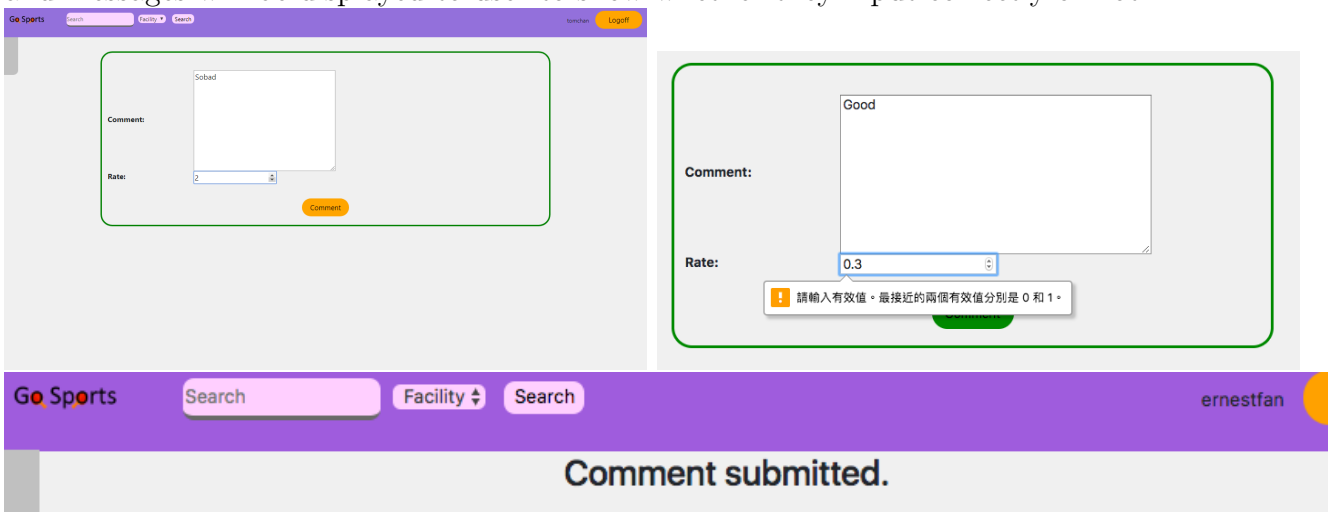
4.2.4 Searching

Users can search other users or facilities under the "Advanced Search" function. The search results will be displayed after the "search" button is clicked.



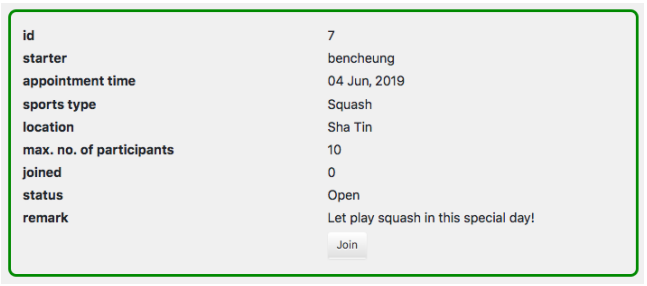
4.2.5 Commenting

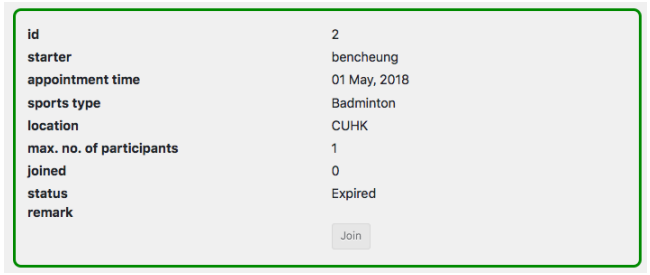
Users can comment on different sportsfields and give rating. On the page viewing all sportsfields, once the "Comment" button is clicked, user will be directed to a comment form. Error checking is also done and messages will be displayed to user to show whether they input correctly or not.



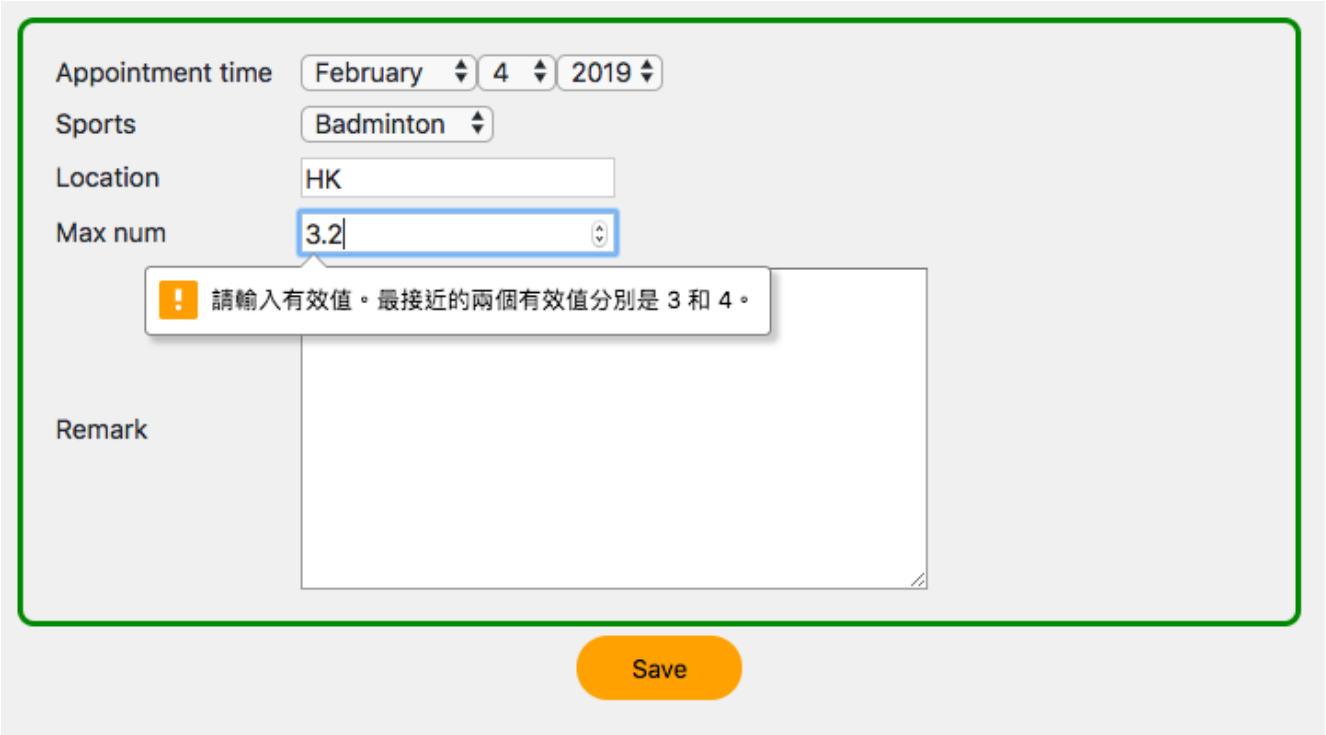
4.2.6 Appointment

Users can create appointment under the appointment function. Users can join open event that is not expired.



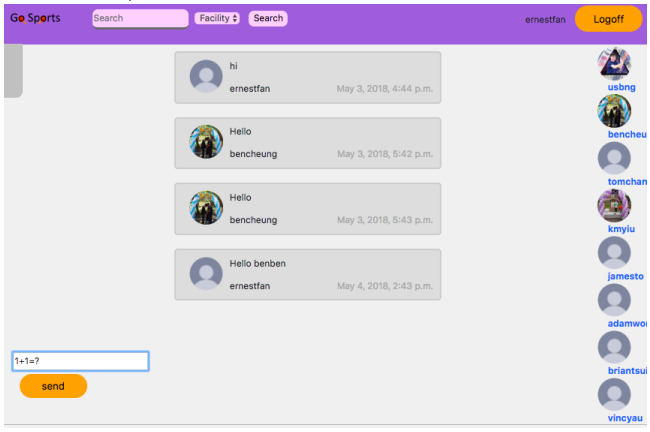


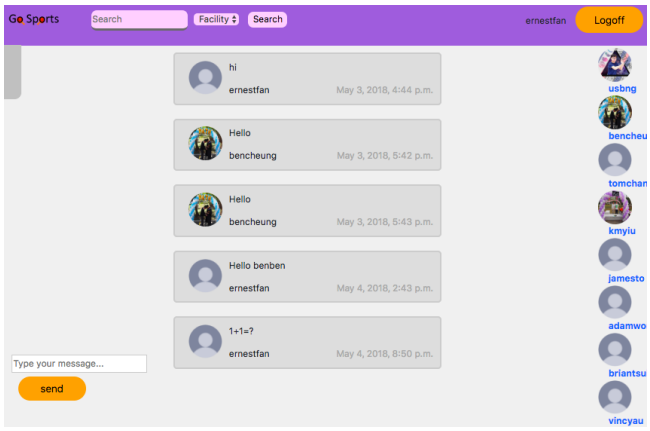
User can also create events by clicking the pencil button, specifying the type of sports, number of participant, etc. Error messages will also be captured.



4.2.7 Chat Room

Users can communicate with each other under the chat room function. One can select receiver and send him/her a message.





5 Testing

5.1 Test Overview and Test Plan

We will test the application by black-box testing. The following use case will test whether each unit(major function) meet out specification's requirement. In each use case below, the input and expected input based on the requirements will be given, and screen capture will provided to demonstrate whether the function works. We will then compare the results with the expectations.

5.2 Use case 1: Registration

5.2.1 Test set 1

5.2.1.1 Purpose

To check whether user database reacts upon post requests accordingly.

5.2.1.2 Inputs

username:cs3100
first name:sw
last name:cs
gender:male
date of birth:Jan 1,2018
password:csci3100
password confirmation:csci3100
email:a@a.com
email confirmation:a@a.com

5.2.1.3 Expected Outputs

In "CustomUser" table of SQLite database, a user will be added with unique id, 1 greater than the largest existing id, with each attribute set as stated. In the browser, the page of successful registration should be returned.

5.2.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.2.1.5 Test result

Success.

5.2.2 Test set 2

5.2.2.1 Purpose

To check whether user name does not appear in the user database before.

5.2.2.2 Inputs

A account with username:cs3100 has been created before this test case.
username:cs3100
first name:sw
last name:cs
gender:male
date of birth:Jan 1,2018
password:csci3100
password confirmation:csci3100
email:a@a.com
email confirmation:a@a.com

5.2.2.3 Expected Outputs

A warning of user name already exist should be appeared and the user is redirected to the registration page.

5.2.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.2.2.5 Test result

Success.

5.2.3 Test set 3

5.2.3.1 Purpose

To check whether the registration module can rejects special character or not.

5.2.3.2 Inputs

username:cs3100@(*1
first name:sw
last name:cs
gender:male
date of birth:Jan 1,2018
password:csci3100
password confirmation:csci3100
email:a@a.com
email confirmation:a@a.com

5.2.3.3 Expected Outputs

A warning of invalid user name exist should be appeared and the user is redirected to the registration page.

5.2.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.2.3.5 Test result

Success.

5.2.4 Test set 4

5.2.4.1 Purpose

To check whether the registration module rejects if password is empty or password and password confirmation are not matched

5.2.4.2 Inputs

username:cs3100
first name:sw
last name:cs
gender:male
date of birth:Jan 1,2018
password:'EMPTY' or csci3100
password confirmation:'EMPTY' or csci3101
email:a@a.com email confirmation:a@a.com

5.2.4.3 Expected Outputs

A warning of empty password or password does not match should be appeared and the user is redirected to the registration page.

5.2.4.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.2.4.5 Test result

Success.

5.2.5 Test set 5

5.2.5.1 Purpose

To check whether registration module rejects registration if the email is empty or format does not match.

5.2.5.2 Inputs

username:cs3100
first name:sw
last name:cs
gender:male
date of birth:Jan 1,2018
password:csci3100
password confirmation:csci3100
email:'EMPTY' or a.com
email confirmation:'EMPTY' or a.com

5.2.5.3 Expected Outputs

A warning of empty email or email does not match should be appeared and the user is redirected to the registration page.

5.2.5.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.2.5.5 Test result

Success.

5.2.6 Test set 6

5.2.6.1 Purpose

To check whether email confirmation system reacts upon post requests accordingly.

5.2.6.2 Inputs

Register an account with the following information. username:cs3100
first name:sw
last name:cs
gender:male
date of birth:Jan 1,2018
password:csci3100
password confirmation:csci3100
email:h435801@nwytg.com, a temporary email
email confirmation:h435801@nwytg.com

5.2.6.3 Expected Outputs

The user cannot be logged in if the user does not confirm his registration in the email. Once the user confirm the registration in the email, then the user is allowed to sign in.

5.2.6.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.2.6.5 Test result

Success.

5.3 Use case 2: Log on

5.3.1 Test set 1

5.3.1.1 Purpose

To check whether log on module reacts upon post requests accordingly.

5.3.1.2 Inputs

username:cs3100
password:csci3100

5.3.1.3 Expected Outputs

The user can login if the user and password match with the database.

5.3.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.3.1.5 Test result

Success.

5.3.2 Test set 2

5.3.2.1 Purpose

To check whether log on module rejects wrong password or not.

5.3.2.2 Inputs

username:cs3100
password:csci3101

5.3.2.3 Expected Outputs

A warning of wrong username or password appears and the user should not be logged in.

5.3.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.3.2.5 Test result

Success.

5.3.3 Test set 3

5.3.3.1 Purpose

To check whether log on module rejects empty password or not.

5.3.3.2 Inputs

username:cs3100
password:'EMPTY'

5.3.3.3 Expected Outputs

A warning of wrong username or password appears and the user should not be logged in.

5.3.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.3.3.5 Test result

Success.

5.3.4 Test set 4

5.3.4.1 Purpose

To check whether log on module rejects wrong username or not.

5.3.4.2 Inputs

username:cs3101
password:csci3100

5.3.4.3 Expected Outputs

A warning of wrong username or password appears and the user should not be logged in.

5.3.4.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.3.4.5 Test result

Success.

5.3.5 Test set 5

5.3.5.1 Purpose

To check whether log on module rejects empty username or not.

5.3.5.2 Inputs

username:cs3101
password:csci3100

5.3.5.3 Expected Outputs

A warning of wrong username or password appears and the user should not be logged in.

5.3.5.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.3.5.5 Test result

Success.

5.3.6 Test set 6

5.3.6.1 Purpose

To check whether log on module rejects special character or not.

5.3.6.2 Inputs

username:cs3100@(*1
password:csci3100

5.3.6.3 Expected Outputs

A warning of wrong username or password appears and the user should not be logged in.

5.3.6.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.3.6.5 Test result

Success.

5.4 Use case 3: Logoff

5.4.1 Test set 1

5.4.1.1 Purpose

To check whether logoff module reacts upon post requests accordingly.

5.4.1.2 Inputs

Sign in with the following account,
username:cs3100
password:csci3100
Then press the logoff button in navigation bar.

5.4.1.3 Expected Outputs

The user can logoff if the user requested to logoff, and the browser is redirected to the home page.

5.4.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.4.1.5 Test result

Success.

5.4.2 Test set 2

5.4.2.1 Purpose

To check whether logoff module handles if no user is signed in.

5.4.2.2 Inputs

Visit http://127.0.0.1:8000/logoff/ while no user is signed in

5.4.2.3 Expected Outputs

The browser is redirected to the home page.

5.4.2.4 Pass/Fail Criteria

The test case passes if outputs match expected outputs exactly, and the user can login successful afterwards.

5.4.2.5 Test result

Success.

5.5 Use case 4: Change User Information

5.5.1 Test set 1

5.5.1.1 Purpose

To check whether user profile in database reacts upon post requests accordingly.

5.5.1.2 Inputs

The value of input blanks are defaulted to be the current setting. No information blank are changed.

5.5.1.3 Expected Outputs

No user information is changed.

5.5.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.1.5 Test result

Success.

5.5.2 Test set 2

5.5.2.1 Purpose

To check whether user profile in database reacts upon post requests accordingly.

5.5.2.2 Inputs

first name:sw
last name:cs
gender:male
email:a@a.com
date of birth:Jan 1,2018
propic:not changed

5.5.2.3 Expected Outputs

All edit are saved and new information can be displayed in profile page instantly.

5.5.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.2.5 Test result

Success.

5.5.3 Test set 3

5.5.3.1 Purpose

To check whether user profile in database rejects special character.

5.5.3.2 Inputs

first name:sw@(*1
last name:cs
gender:male
email:a@a.com
date of birth:Jan 1,2018
propic:not changed

5.5.3.3 Expected Outputs

All edit are saved and new information can be displayed in profile page instantly.

5.5.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.3.5 Test result

Success.

5.5.4 Test set 4

5.5.4.1 Purpose

To check whether the profile editing module rejects if no gender is selected.

5.5.4.2 Inputs

first name:sw
last name:cs
gender:not selected
email:a@a.com
date of birth:Jan 1,2018
propic:not changed

5.5.4.3 Expected Outputs

A warning of no gender selected should be displayed and user profile is not changed.

5.5.4.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.4.5 Test result

Success.

5.5.5 Test set 5

5.5.5.1 Purpose

To check whether the profile editing module rejects if no date of birth is selected.

5.5.5.2 Inputs

first name:sw
last name:cs
gender:male
email:a@a.com
date of birth:not selected
propic:not changed

5.5.5.3 Expected Outputs

A warning of no date of birth selected should be displayed and user profile is not changed.

5.5.5.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.5.5 Test result

Success.

5.5.6 Test set 6

5.5.6.1 Purpose

To check whether profile picture(propic) feature the reacts upon post requests accordingly.

5.5.6.2 Inputs

first name:sw
last name:cs
gender:male
email:a@a.com
date of birth:Jan 1,2018
propic:clear propic selected

5.5.6.3 Expected Outputs

The original propic is cleared and replaced by a default propic.

5.5.6.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.6.5 Test result

Success.

5.5.7 Test set 7

5.5.7.1 Purpose

To check whether profile picture(propic) feature the reacts upon post requests accordingly.

5.5.7.2 Inputs

first name:sw
last name:cs
gender:male
email:a@a.com
date of birth:Jan 1,2018
propic:cs3100.jpg, a 256x256 image

5.5.7.3 Expected Outputs

The original propic is cleared and replaced by a default propic.

5.5.7.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.7.5 Test result

Success.

5.5.8 Test set 8

5.5.8.1 Purpose

To check whether profile picture(propic) feature rejects wrong file type.

5.5.8.2 Inputs

first name:sw
last name:cs
gender:male
email:a@a.com
date of birth:Jan 1,2018
propic:cs3100.txt

5.5.8.3 Expected Outputs

A warning of invalid image should be displayed and no information is changed.

5.5.8.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.5.8.5 Test result

Success.

5.6 Use case 5: Search User

5.6.1 Test set 1

5.6.1.1 Purpose

To check whether search use module reacts upon post requests accordingly.

5.6.1.2 Inputs

user:cs3100

5.6.1.3 Expected Outputs

A user with username cs3100 should be displayed.

5.6.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.6.1.5 Test result

Success.

5.6.2 Test set 2

5.6.2.1 Purpose

To check whether search use module reacts upon post requests accordingly.

5.6.2.2 Inputs

user:cs

5.6.2.3 Expected Outputs

Several user with username containing the string cs should be displayed.

5.6.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.6.2.5 Test result

Success.

5.6.3 Test set 3

5.6.3.1 Purpose

To check whether search use module reacts upon post requests accordingly if the there is no match in the database.

5.6.3.2 Inputs

user:ilovecsci3100

5.6.3.3 Expected Outputs

A message of no matching results should be displayed.

5.6.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.6.3.5 Test result

Fail. All user profile are displayed.

5.6.4 Test set 4

5.6.4.1 Purpose

To check whether search use module rejects empty string

5.6.4.2 Inputs

user:'EMPTY'

5.6.4.3 Expected Outputs

No user should be displayed

5.6.4.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.6.4.5 Test result

Fail. All user profile are displayed.

5.7 Use case 6: Search Facility in Advanced Search

5.7.1 Test set 1

5.7.1.1 Purpose

To check whether search facility module reacts upon post requests accordingly.

5.7.1.2 Inputs

facility name:island
facility district:'EMPTY'

5.7.1.3 Expected Outputs

A facility with name, Island East Swimming Pool should be displayed.

5.7.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.1.5 Test result

Success.

5.7.2 Test set 2

5.7.2.1 Purpose

To check whether search facility module handles several match results.

5.7.2.2 Inputs

facility name:pool
facility district:'EMPTY'

5.7.2.3 Expected Outputs

Several facilities with name containing the string pool should be displayed. No other facilities are shown.

5.7.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.2.5 Test result

Success.

5.7.3 Test set 3

5.7.3.1 Purpose

To check whether search facility module reacts if no search result are matched.

5.7.3.2 Inputs

facility name:ilovesoftware
facility district:'EMPTY'

5.7.3.3 Expected Outputs

A message of no matching results should be displayed and the browser is returned to the last page.

5.7.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.3.5 Test result

Success.

5.7.4 Test set 4

5.7.4.1 Purpose

To check whether search facility module reacts upon post requests accordingly.

5.7.4.2 Inputs

facility name:'EMPTY'
facility district:Eastern

5.7.4.3 Expected Outputs

Two facilities in Eastern should be displayed.

5.7.4.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.4.5 Test result

Success.

5.7.5 Test set 5

5.7.5.1 Purpose

To check whether search facility module reacts upon post requests accordingly.

5.7.5.2 Inputs

facility name:'EMPTY'
facility district:shenzhen

5.7.5.3 Expected Outputs

A message of no matching results should be displayed and the browser is returned to the last page.

5.7.5.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.5.5 Test result

Success.

5.7.6 Test set 6

5.7.6.1 Purpose

To check whether search facility module reacts upon post requests accordingly.

5.7.6.2 Inputs

facility name:pool
facility district:shatin

5.7.6.3 Expected Outputs

Several facilities with name containing the string pool and located in shatin should be displayed. No other facilities are shown.

5.7.6.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.6.5 Test result

Success.

5.7.7 Test set 7

5.7.7.1 Purpose

To check whether search facility module reacts if no search result are matched.

5.7.7.2 Inputs

facility name:ilovesoftware
facility district:shatin

5.7.7.3 Expected Outputs

A message of no matching results should be displayed and the browser is returned to the last page.

5.7.7.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.7.5 Test result

Success.

5.7.8 Test set 8

5.7.8.1 Purpose

To check whether search facility module rejects special character.

5.7.8.2 Inputs

facility name:@
facility district:'EMPTY'

5.7.8.3 Expected Outputs

A message of no matching results, or a warning of invalid input should be displayed. The browser is returned to the last page.

5.7.8.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.8.5 Test result

Success.

5.7.9 Test set 9

5.7.9.1 Purpose

To check whether search facility module rejects special character.

5.7.9.2 Inputs

facility name:'EMPTY'
facility district:@

5.7.9.3 Expected Outputs

A message of no matching results, or a warning of invalid input should be displayed. The browser is returned to the last page.

5.7.9.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.9.5 Test result

Success.

5.7.10 Test set 10

5.7.10.1 Purpose

To check whether search facility module reacts upon post requests if the input string are empty.

5.7.10.2 Inputs

facility name:'EMPTY'
facility district:'EMPTY'

5.7.10.3 Expected Outputs

All facilities are displayed.

5.7.10.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.7.10.5 Test result

Success.

5.8 Use case 7: Search Facility in Navigation Bar

5.8.1 Test set 1

5.8.1.1 Purpose

To check whether search facility module reacts upon post requests accordingly.

5.8.1.2 Inputs

facility:island

5.8.1.3 Expected Outputs

A facility with name, Island East Swimming Pool should be displayed.

5.8.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.8.1.5 Test result

Success.

5.8.2 Test set 2

5.8.2.1 Purpose

To check whether search facility module handles several match results.

5.8.2.2 Inputs

facility:pool

5.8.2.3 Expected Outputs

Several facilities with name containing the string pool or with district name containing the string pool should be displayed. No other facilities are shown.

5.8.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.8.2.5 Test result

Success.

5.8.3 Test set 3

5.8.3.1 Purpose

To check whether search facility module handles several match results.

5.8.3.2 Inputs

facility:s

5.8.3.3 Expected Outputs

Several facilities with name containing the string s,eg sports, or with district name containing the string s,eg shatin, should be displayed. No other facilities are shown.

5.8.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.8.3.5 Test result

Success.

5.8.4 Test set 4

5.8.4.1 Purpose

To check whether search facility module reacts if no search result are matched.

5.8.4.2 Inputs

facility:ilovesoftware

5.8.4.3 Expected Outputs

A message of no matching results should be displayed and the browser is returned to the last page.

5.8.4.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.8.4.5 Test result

Success.

5.8.5 Test set 5

5.8.5.1 Purpose

To check whether search facility module reacts upon post requests accordingly.

5.8.5.2 Inputs

facility:'EMPTY'

5.8.5.3 Expected Outputs

All facilities should be displayed.

5.8.5.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.8.5.5 Test result

Success.

5.8.6 Test set 6

5.8.6.1 Purpose

To check whether search facility module rejects special character.

5.8.6.2 Inputs

facility:@

5.8.6.3 Expected Outputs

A message of no matching results, or a warning of invalid input should be displayed. The browser is returned to the last page.

5.8.6.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.8.6.5 Test result

Success.

5.9 Use case 8: Facility Details

5.9.1 Test set 1

5.9.1.1 Purpose

To check whether facility module reacts upon post requests accordingly.

5.9.1.2 Inputs

Click the facility, Cheung Chau Sports Centre, in the facility list.

5.9.1.3 Expected Outputs

The details of Cheung Chau Sports Centre, including district, address, sports type and a google map with facility locations are shown

5.9.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.9.1.5 Test result

Success.

5.9.2 Test set 2

5.9.2.1 Purpose

To check whether facility module reacts upon post requests accordingly.

5.9.2.2 Inputs

Visit http://127.0.0.1:8000/facility/100000/
There is no facility with id 100000

5.9.2.3 Expected Outputs

A warning of no such facility should be displayed, or the browser is returned to last page.

5.9.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.9.2.5 Test result

Fail. A page not found (404) error page is displayed.

5.10 Use case 9: Sports Field Details

5.10.1 Test set 1

5.10.1.1 Purpose

To check whether sports field module reacts upon post requests accordingly.

5.10.1.2 Inputs

Click the sports field, Ma On Shan Sports Centre Basketball Court 1, in the sports field list.

5.10.1.3 Expected Outputs

The details of Ma On Shan Sports Centre Basketball Court 1, such as the opening hour, rating(in our system) and comments(in our system) are shown

5.10.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.10.1.5 Test result

Success.

5.10.2 Test set 2

5.10.2.1 Purpose

To check whether comment module reacts upon post requests accordingly.

5.10.2.2 Inputs

Add a comment in Ma On Shan Sports Centre Basketball Court 1.
comment:This is great.
rate:5 Originally there are no comment for this field.

5.10.2.3 Expected Outputs

The rating is updated and a new comment is shown in the sports field detail page.

5.10.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.10.2.5 Test result

Success.

5.10.3 Test set 3

5.10.3.1 Purpose

To check whether comment module reacts upon post requests accordingly.

5.10.3.2 Inputs

Add a comment in Ma On Shan Sports Centre Basketball Court 1.
comment:This is great too.
rate:3 Originally there are one comment for this field. The rating of the field is 5.

5.10.3.3 Expected Outputs

The rating is updated by taking average and a new comment is shown in the sports field detail page.

5.10.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.10.3.5 Test result

Success.

5.10.4 Test set 4

5.10.4.1 Purpose

To check whether comment module reacts rejects invalid rating input.

5.10.4.2 Inputs

Add a comment in Ma On Shan Sports Centre Basketball Court 1.
comment:This is great.
rate:6, tested in two situations, direct input 6 into the blank, and click the +1 button up to 6

5.10.4.3 Expected Outputs

A warning of invalid rating should be displayed and the browser should be returned to the last page.

5.10.4.4 Pass/Fail Criteria

The test case passes if the database do not make changes.

5.10.4.5 Test result

Success. There is no change in the database. However, no action is taken in the browser.

5.10.5 Test set 5

5.10.5.1 Purpose

To check whether comment module reacts rejects blank rating input.

5.10.5.2 Inputs

Add a comment in Ma On Shan Sports Centre Basketball Court 1.
comment:This is great.
rate:blank

5.10.5.3 Expected Outputs

A warning of invalid rating should be displayed and the browser should be returned to the last page.

5.10.5.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.10.5.5 Test result

Success.

5.11 Use case 10: Chat System

5.11.1 Test set 1

5.11.1.1 Purpose

To check whether chat module reacts upon post requests accordingly.

5.11.1.2 Inputs

Send a message to a selected user.
message:How are you?

5.11.1.3 Expected Outputs

The selected user should receive a notification of new message and the message is displayed in the char record.

5.11.1.4 Pass/Fail Criteria

The test case passes if the message is displayed after refreshing the website.

5.11.1.5 Test result

Success.

5.11.2 Test set 2

5.11.2.1 Purpose

To check whether chat module rejects empty message input.

5.11.2.2 Inputs

Send a message to a selected user.
message:'EMPTY'

5.11.2.3 Expected Outputs

A warning of empty input should be displayed and no message is sent.

5.11.2.4 Pass/Fail Criteria

The test case passes if no message is sent.

5.11.2.5 Test result

Success.

5.11.3 Test set 3

5.11.3.1 Purpose

To check whether chat module handle special characters.

5.11.3.2 Inputs

Send a message to a selected user.
message:@*()[];

5.11.3.3 Expected Outputs

The selected user should receive a notification of new message and the message is displayed in the char record.

5.11.3.4 Pass/Fail Criteria

The test case passes if the message is displayed after refreshing the website.

5.11.3.5 Test result

Success.

5.11.4 Test set 4

5.11.4.1 Purpose

To check whether chat module handles long string input.

5.11.4.2 Inputs

Send a message to a selected user.
message:a string with 40,000 characters.

5.11.4.3 Expected Outputs

The selected user should receive a notification of new message and the message is displayed in the char record.

5.11.4.4 Pass/Fail Criteria

The test case passes if the message is displayed after refreshing the website.

5.11.4.5 Test result

Success.

5.11.5 Test set 4

5.11.5.1 Purpose

To check whether chat module handles emoji.

5.11.5.2 Inputs

Send a message to a selected user.
message:Unicode Character 'GRINNING FACE' (U+1F600)

5.11.5.3 Expected Outputs

The selected user should receive a notification of new message and the message is displayed in the char record.

5.11.5.4 Pass/Fail Criteria

The test case passes if the message is displayed after refreshing the website.

5.11.5.5 Test result

Success.

5.11.6 Test set 5

5.11.6.1 Purpose

To check whether chat records are private to others

5.11.6.2 Inputs

Visit http://127.0.0.1:8000/chat/4/2 without signed in.

5.11.6.3 Expected Outputs

The browser should return to the last page.

5.11.6.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.11.6.5 Test result

Success.

5.11.7 Test set 5

5.11.7.1 Purpose

To check whether chat records are private to others

5.11.7.2 Inputs

Visit http://127.0.0.1:8000/chat/4/2 with a third account.

5.11.7.3 Expected Outputs

The browser should return to the last page.

5.11.7.4 Pass/Fail Criteria

The test case passes if no chat record of others are shown.

5.11.7.5 Test result

Fail. Chat records of others are displayed.

5.12 Use case 11: Appointment System

5.12.1 Test set 1

5.12.1.1 Purpose

To check whether appointment module reacts upon post requests accordingly.

5.12.1.2 Inputs

Join an activity which is not full and not expired.

5.12.1.3 Expected Outputs

The number of joined user increases by one and it is updated on appointment list.

5.12.1.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.12.1.5 Test result

Success.

5.12.2 Test set 2

5.12.2.1 Purpose

To check whether appointment module reacts upon post requests accordingly.

5.12.2.2 Inputs

Join an activity which is full.

5.12.2.3 Expected Outputs

The join button should be disabled. No actions by clicking the join button.

5.12.2.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.12.2.5 Test result

Success.

5.12.3 Test set 3

5.12.3.1 Purpose

To check whether appointment module reacts upon post requests accordingly.

5.12.3.2 Inputs

Join an activity which is expired.

5.12.3.3 Expected Outputs

The join button should be disabled. No actions by clicking the join button.

5.12.3.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.12.3.5 Test result

Success.

5.12.4 Test set 4

5.12.4.1 Purpose

To check whether appointment module reacts upon post requests accordingly.

5.12.4.2 Inputs

Create an activity with the following information.
Appointment time: Jan 1,2019
Sports: Gymnasium
Location: UC Gym
Max number of participants: 5

5.12.4.3 Expected Outputs

A new activity is added into the database.

5.12.4.4 Pass/Fail Criteria

The test case passes if and only if outputs match expected outputs exactly.

5.12.4.5 Test result

Success.

5.12.5 Test set 5

5.12.5.1 Purpose

To check whether appointment module rejects creation of expired activities.

5.12.5.2 Inputs

Create an activity with the following information.
Appointment time: Jan 1,2018
Sports: Gymnasium
Location: UC Gym
Max number of participants: 5

5.12.5.3 Expected Outputs

A warning of expired date is displayed and the database is not modified.

5.12.5.4 Pass/Fail Criteria

The test case passes if the database is not modified.

5.12.5.5 Test result

Fail. A new activity is added.

5.12.6 Test set 6

5.12.6.1 Purpose

To check whether appointment module records the participants.

5.12.6.2 Inputs

Create an activity with the following information.
Appointment time: Jan 1,2019
Sports: Gymnasium
Location: UC Gym
Max number of participants:5
Then try to join the above activity created by the same user.

5.12.6.3 Expected Outputs

The join button should be disabled. No actions by clicking the join button.

5.12.6.4 Pass/Fail Criteria

The test case passes if the database is not modified.

5.12.6.5 Test result

Success.

6 Lessons Learned

This project in software development gave us an extraordinary and unforgettable experience. No matter in the sense of complexity, scale, duration, and workload, we all learned a big lesson working throughout this project.

First of all, developing a web application requires a lot of open sources software. In our case, we implement our project using an open source Python programs developed by Django Software Foundation. Also, HTML and CSS are needed to provide the interface of the product. All such usage skills of open sources applications are not covered in class and hence the whole team have to spent quite a lot of time learning to get used to them. This trained us to be more capable in independent learning. In the future, these independent learning skills can help us to be more confident on starting up new projects and encountering non-familiar things.

Secondly, the spirit of software development is to separate modules and follow the specification and design to implement the application. It teaches us to first think carefully before really working on it. To do something big, we must first design and specify what we need and want we want. Then carefully think evaluate and analyze whether our approach is suitable or not.

Thirdly, this is a good chance for us to integrate and make good use of our knowledge. We learn many things in different courses such as database, algorithm, data structures, but we never apply them together or apply them in real life. Developing this project give us a chance to integrate those things together, and produce a meaningful product.

As for the aspect of a teamwork project, we know the significance of division of labour. Works are to be divided according to each teammates talent and interest. It will definitely speed up the whole process. Also, since this project spanned the whole semester, we may have to deal with other midterms and assignments, so better time management is needed. We learned that not to be a deadline fighter, especially for such a project in a large scale.

7 Conclusion

Go-Sports is a software engineering project that aims to let us experience a real-life software development process. It includes various aspects of software development skills such as design, implementation, coding, testing and verification.

In the first phase of the project, we come up with different idea that we can work into, and then design the requirements, features and functions that should be appeared. They are reflected in the initial design report and the first part of this final report, by the provided DFDs and UML diagrams. We must point out that this phase is actually very important, as it is the foundation of the whole project.

Then it comes with the coding phase, we implemented Go-Sports with Django, HTML and SQLite All of them are open source development tools for us to start up a web application. One has to learn the syntax and infrastructure of these tools. It is important to have a good coding style and good communications with teammates, as a team worked together and contribute to the same project. In order to produce a successful project, the codes have to be consistent and coherent. Thus, meaningful variable and command are to be given in the code.

The next stage is the testing phase. To verify that the product meet our requirement, we have to test it against with custom generated test cases. They include normal test cases that test the general behavior of our program, and also extreme test cases that test the boundary cases and exception handling of our program. In this stage, we have to generate test cases ourselves and often need to debug according to the failed test case.

Finally, it is the documentation phase. Every detail of the software development process are to be recorded for the sake of maintenance and evolution. This report is one of the product. Discussions and improvement suggestions can be made so that the unnecessary problems can be avoided if we have to develop similar software.

To conclude, this project is a training on software development. Different from coding and algorithm practices, it aim on not only the skills on coding, but also design and documentation. It equipped us the skills that maybe useful when we have to develop software in the future.