# Hong Kong Institute of Vocational Education (Tsing Yi)
## Department of Information Technology

## Higher Diploma in Software Engineering (IT114105)
## Assignment (ITP4511)
## Report
## (2021/2022)

### Tang Chun Hei (200022972)
### Group: B14

We declare that this is a group project and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

| Student | Contribution to the project (%) (Total 100%) | Signature |
|---|---|---|
| Tang Chun Hei | 100% | Hei |
|  |  |  |

a) Assumption and the user and system requirements

**Customer** will need to request booking with specific date, time, trainer and location. The unavailable trainer and location will not be shown on the list. After request a booking, the user can check the previous request status or cancel the booking request.

**Staff/ Personal Trainer** will need to list all the personal trainer and gym center details. Also, they can handle the add, edit and delete personal trainer and gym center for administration. They can change the booking request status to Confirm or Decline.

**Senior Manager** will need to list all of the existing users, add and delete users. The manager will allow to edit the user's role and all the details. Also, generate a detailed report for selected trainer or gym center with income and booking rate in monthly and yearly.
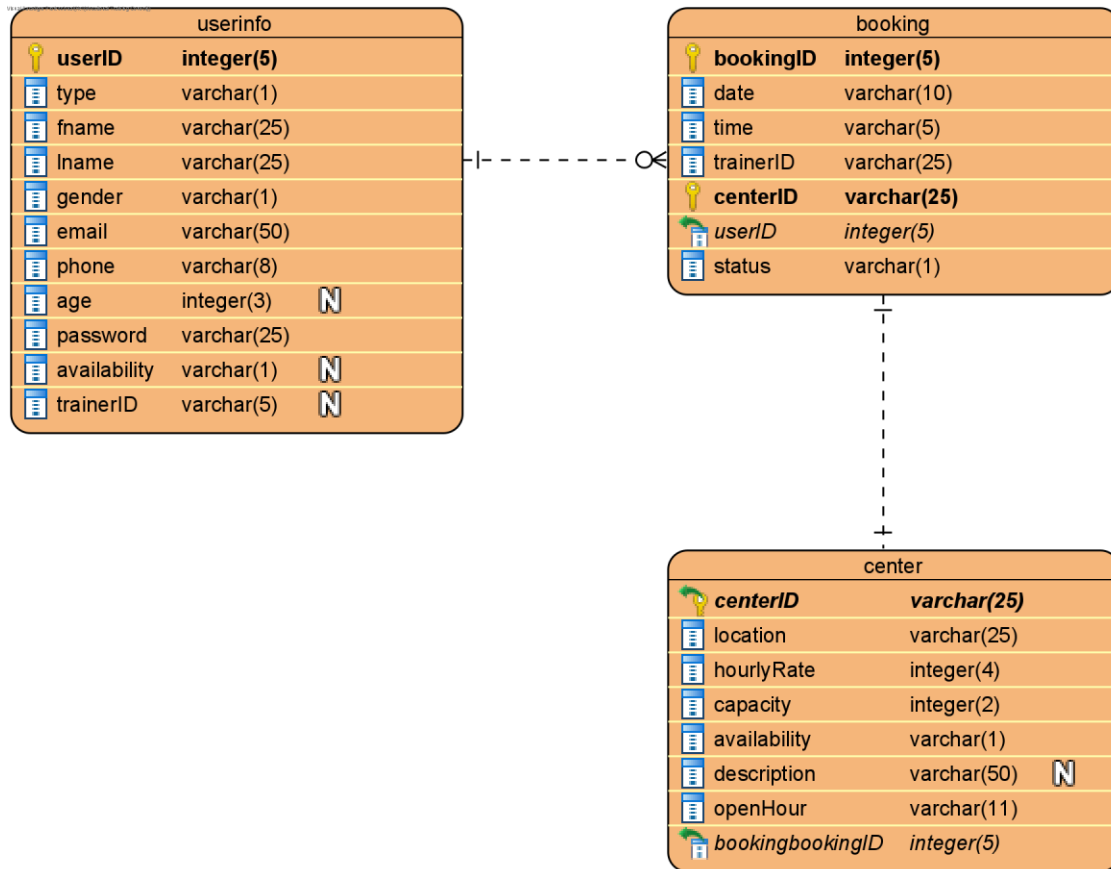
b) Site map
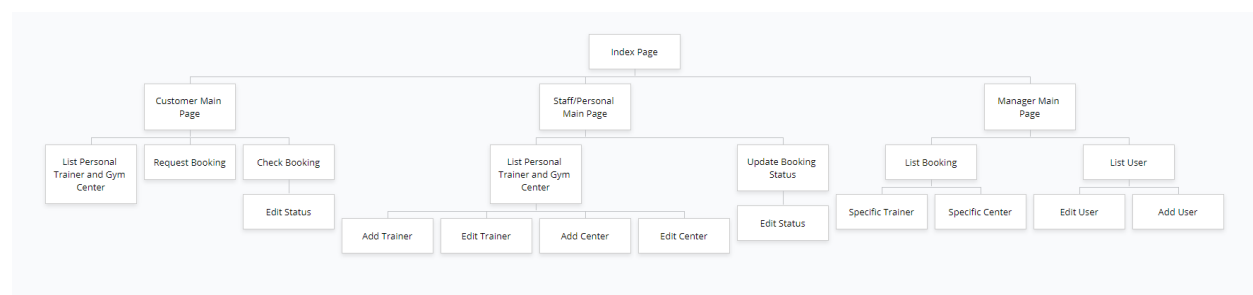


c) System structure on how MVC Model is applied

The MVC model aims to let the users browse the site and transfer data through the controller. Therefore, each time the user requests some data from the database, the controller will get some variable from the previous page and request the data which will use on the next page. For example, the Customer Main Page will store some user's information. When the user clicks "Check booking", the controller will find the information base on the user's information, so that the screen will only show that user's booking not all the user booking. Same step, when the user wants to cancel the request, the system will get the selected booking record and display on the next page. After the user submit the change, the system will return the status to the database and save it and so on.

## d) Database structure

### userinfo

| | userID | integer(5) |
|---|---|---|
| | type | varchar(1) |
| | fname | varchar(25) |
| | lname | varchar(25) |
| | gender | varchar(1) |
| | email | varchar(50) |
| | phone | varchar(8) |
| | age | integer(3) |
| | password | varchar(25) |
| | availability | varchar(1) |
| | trainerID | varchar(5) |

### booking

| | bookingID | integer(5) |
|---|---|---|
| | date | varchar(10) |
| | time | varchar(5) |
| | trainerID | varchar(25) |
| | centerID | varchar(25) |
| | userID | integer(5) |
| | status | varchar(1) |

### center

| | centerID | varchar(25) |
|---|---|---|
| | location | varchar(25) |
| | hourlyRate | integer(4) |
| | capacity | integer(2) |
| | availability | varchar(1) |
| | description | varchar(50) |
| | openHour | varchar(11) |
| | bookingbookingID | integer(5) |

e) Brief description (1 or 2 pages only) on the major characteristics and design of your application



For the menu, the color patten must let the user see everything clearly and smoothly. For the table, the heading is easy to identify in each column. I use two color to separate the row to let the user can read the data easier.
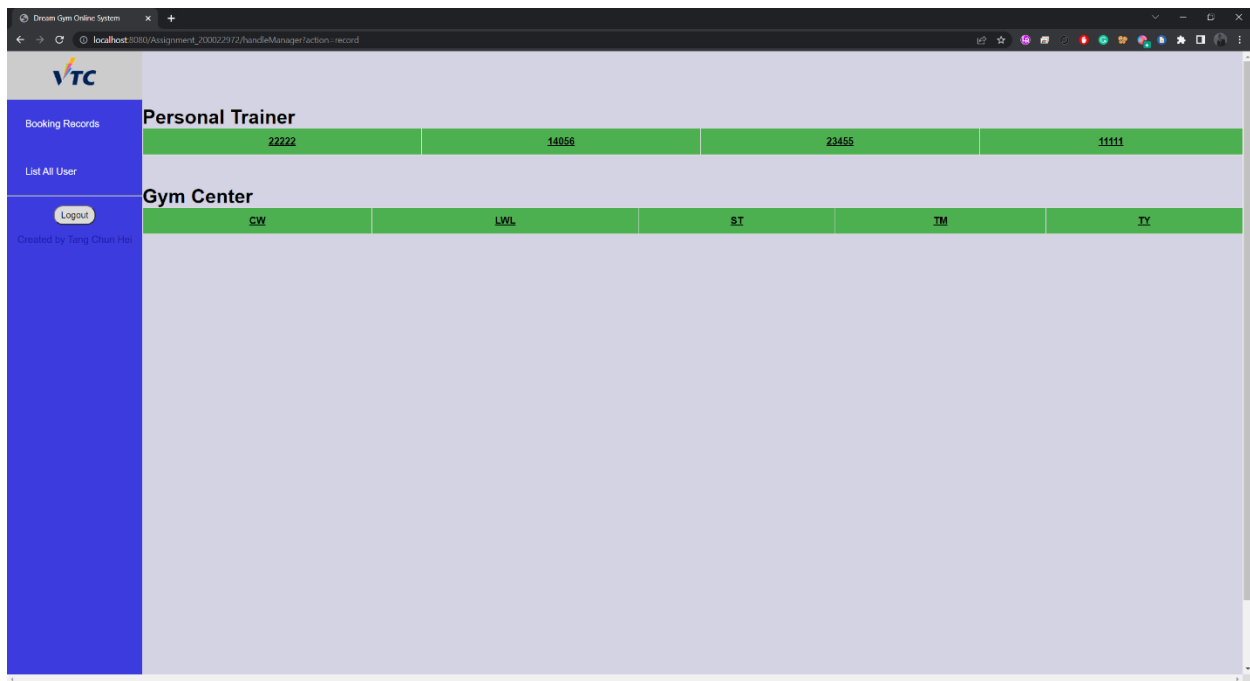
For the clickable link on main container, I choose black and under line to make it distinguishable.

f) Conclusions

To summarize, the whole project uses the MVC model to locate and operate the data from page to page and database. The things that I need to concern are the system structure since if I design the whole structure well, the coding part will become easier and clearer. Besides, the data type of each database is also important as it aims to store the data which users will use. These two things must design and complete before the coding work.

g) Skill checklist which lists your used skills (or technologies) in a single page and highlights the skills and technologies applied in your project
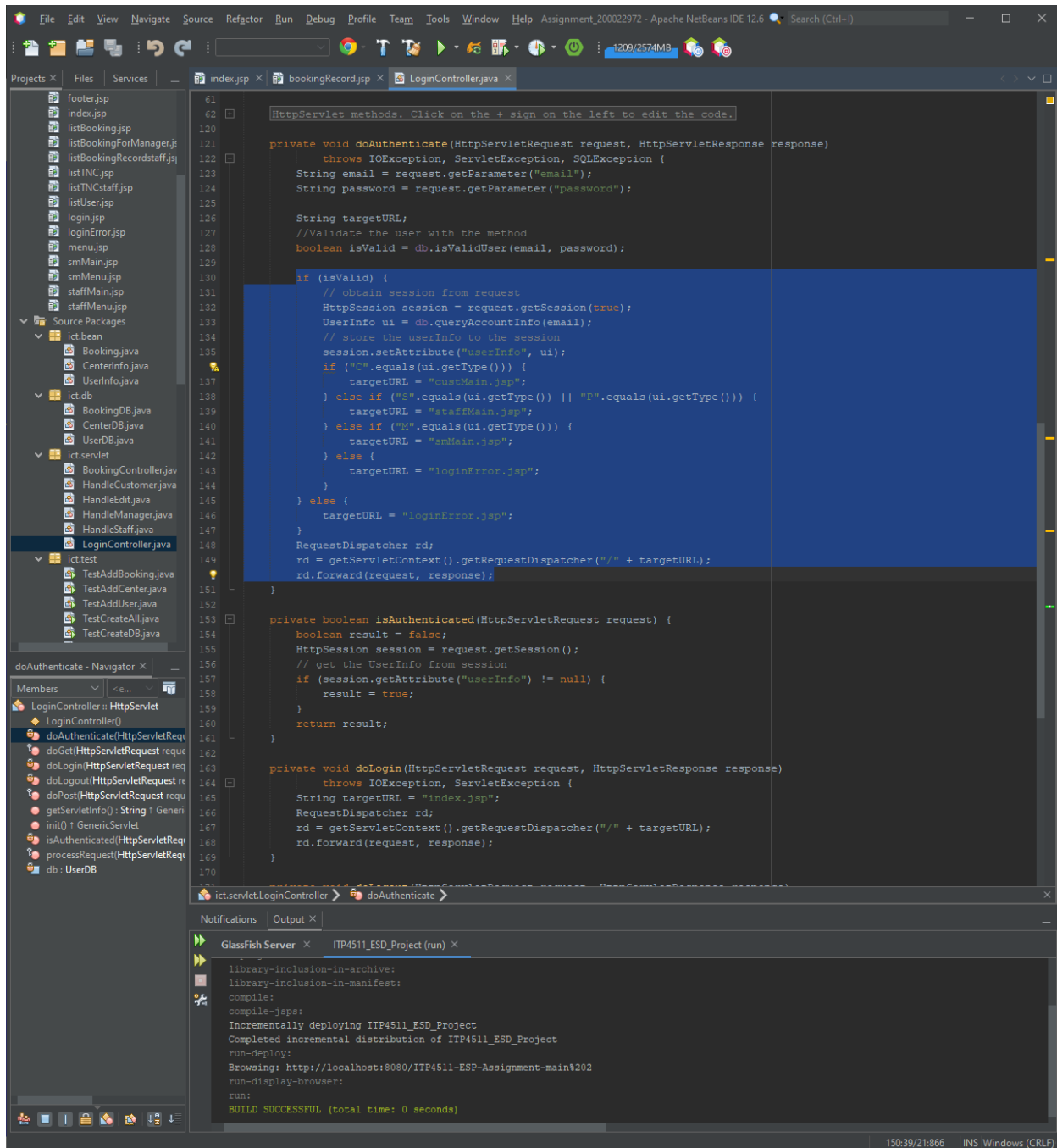


Use JSP/servlets to dynamically generate HTML pages

Use JSP/servlets to accept user inputs from browser

Use JSP Action

Use JavaBean

Use JDBC for database connection

Use session checking

Use login control

Apply the MVC model