Higher Diploma in Software Engineering (IT114105)

Coursework (Semester 5 - 2021/2022)

Enterprise Systems Development (ITP4511)

Group C7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Work break down | Student 1 | Work break down | Signature |  |
| Student 1 | Ng Chun Yin 200090262 | * Analytic / Report * Account Management * Login Control / Session Checking |  | 50% |
| Student 2 | Ngai Pan 20099268 | * Personal Trainer and Gym Center Management * Personal trainer and Gym Venue Booking |  | 50% |

Contents

[1 Assumption and the user and system requirements 4](#_Toc102256741)

[1.1 Assumptions 4](#_Toc102256742)

[1.2 User and system requirements 4](#_Toc102256743)

[1.2.1 Personal Trainer/Gym Center Management (for Staff/ Personal Trainer) 4](#_Toc102256744)

[1.2.2 Personal trainer and Gym Venue Booking (for Customer) 5](#_Toc102256745)

[1.2.3 Analytic / Report (for Senior Management) 5](#_Toc102256746)

[1.2.4 Account Management (for Senior Management) 5](#_Toc102256747)

[1.2.5 Login Control/Session checking 6](#_Toc102256748)

[2 Site map 7](#_Toc102256749)

[2.1 Entire Site Map 7](#_Toc102256750)

[2.2 Personal Trainer Part 7](#_Toc102256751)

[2.3 Staff Part 8](#_Toc102256752)

[2.4 Customer Part 9](#_Toc102256753)

[2.5 Senior Management Part 10](#_Toc102256754)

[3 System structure on how MVC Model is applied 11](#_Toc102256755)

[3.1 Personal Trainer and Gym Center Management (for Personal Trainer) 11](#_Toc102256756)

[3.2 Personal trainer and Gym Venue Booking (for Customer) 17](#_Toc102256757)

[3.3 Analytic / Report (for Senior Management) 20](#_Toc102256758)

[3.4 Account Management (for Senior Management) 22](#_Toc102256759)

[4 Database structure 25](#_Toc102256760)

[5 Brief major characteristics and design of the application 28](#_Toc102256761)

[6 Conclusions 30](#_Toc102256762)

[7 Skill checklist 31](#_Toc102256763)

[7.1 Use JSP/servlets to dynamically generate HTML pages 32](#_Toc102256764)

[7.2 Use JSP/servlets to accept user inputs from browser 33](#_Toc102256765)

[7.3 Use JSP Action 35](#_Toc102256766)

[7.4 Use JavaBean 36](#_Toc102256767)

[7.5 Use JDBC for database connection 37](#_Toc102256768)

[7.6 Use session checking 39](#_Toc102256769)

[7.7 Use login control 40](#_Toc102256770)

[7.8 Apply the MVC model 42](#_Toc102256771)

[7.9 Other skills applied 44](#_Toc102256772)

[7.9.1 Google chart 44](#_Toc102256773)

[7.9.2 JavaScript- Table to CSV (found on internet) 44](#_Toc102256774)

# Assumption and the user and system requirements

## Assumptions

* User role of trainers and customers cannot be changed.
* All the management role cannot be disabled at one time
* All roles belong to 4 role types: customer, trainer, staff, management.
* Booking rate mean booking rate in this year.
* The year's booking rate is calculated by (number of confirmed booking of the trainer/center) / (number of days in this year \* 12).
* The month’s booking rate is calculated by (number of confirmed booking of the trainer/center) / (number of days in this month \* 12).

## User requirements

### Personal Trainer/Gym Center Management (for Staff/ Personal Trainer)

● Show a list of all personal trainer and gym center

● Add / edit / delete personal trainer and gym center\*

● Enable/disable listing of personal trainer and gym center on booking system

● Confirm/decline booking request

### Personal trainer and Gym Venue Booking (for Customer)

● Show a list of personal trainer and gym center and related features (e.g. Image, name, description, availability status)

● Send booking request / check booking request status

● Booking reminder

● Check/ Update personal booking record

### Analytic / Report (for Senior Management)

● Show a list of booking records of the selected personal trainer (s) and/or Gym Center (s)

● Show the booking rate of selected personal trainer and gym center (calculated by month/year)

● Show the monthly/ yearly income generated by each personal trainer (personal training fee) and gym center (venue booking fee).

### Account Management (for Senior Management)

● Show a list of existing users

● Create and delete users

● Edit users with detail and roles

● Manage the user role

### Login Control/Session checking

* User have to login to use the web system
* After user login they have to login again to use the system

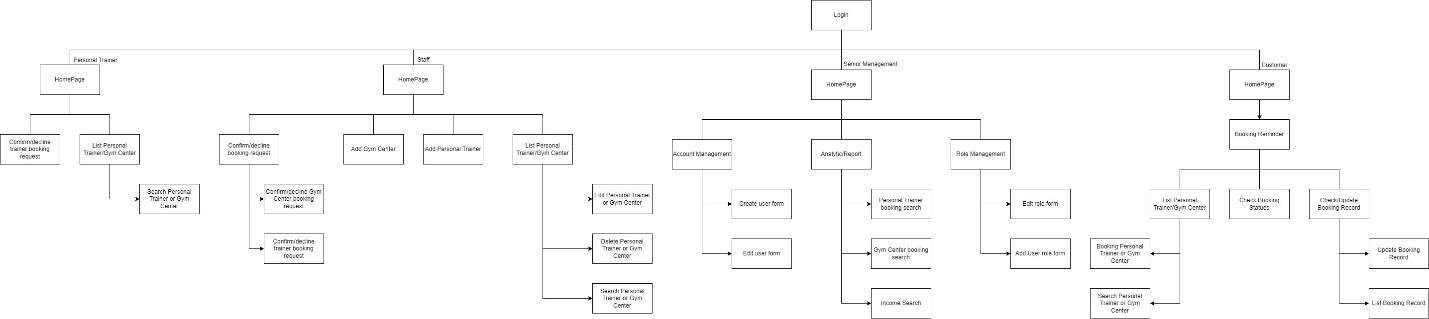
## System Requirements

Diagram

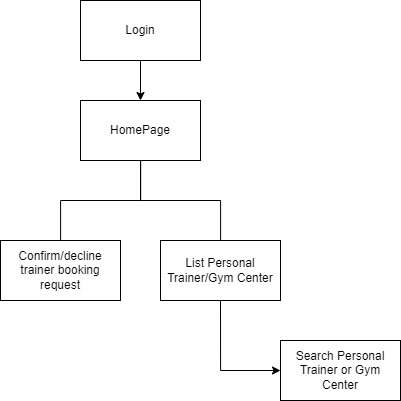
Description automatically generated  
Four-Tiered Client Server Architecture

# Site map

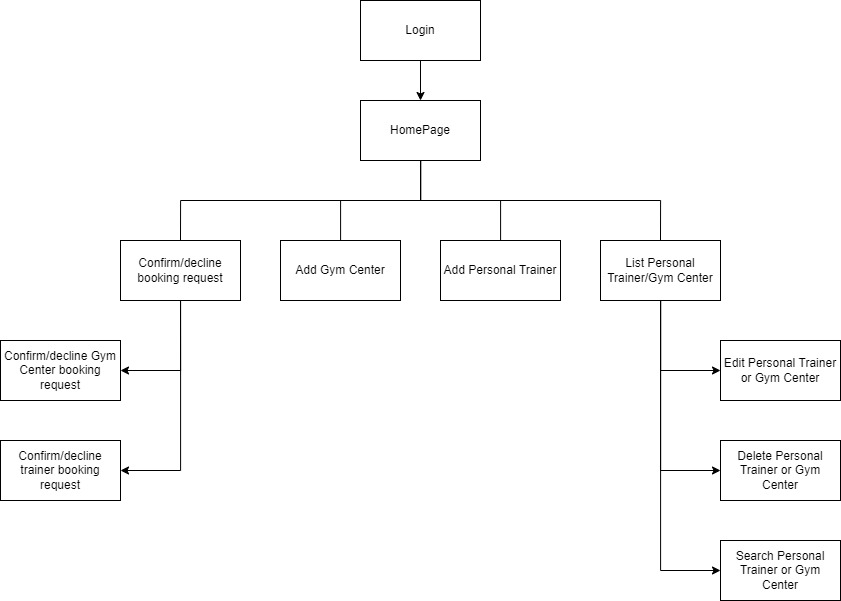
## Entire Site Map



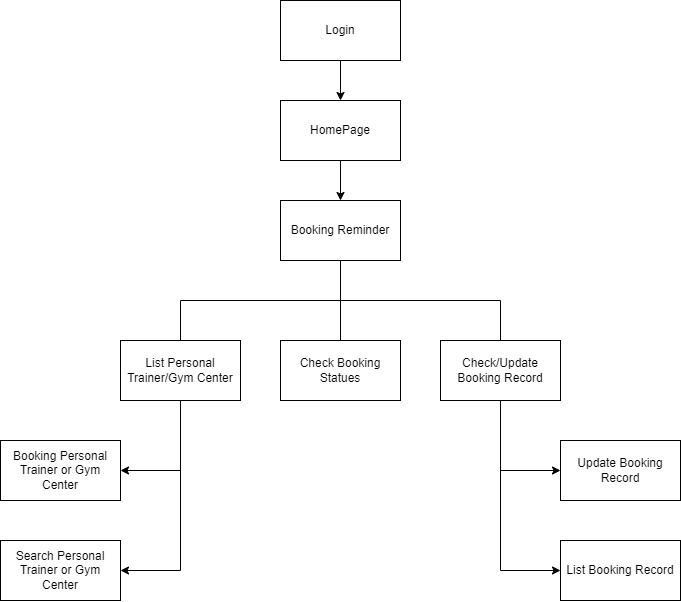
## Personal Trainer Part



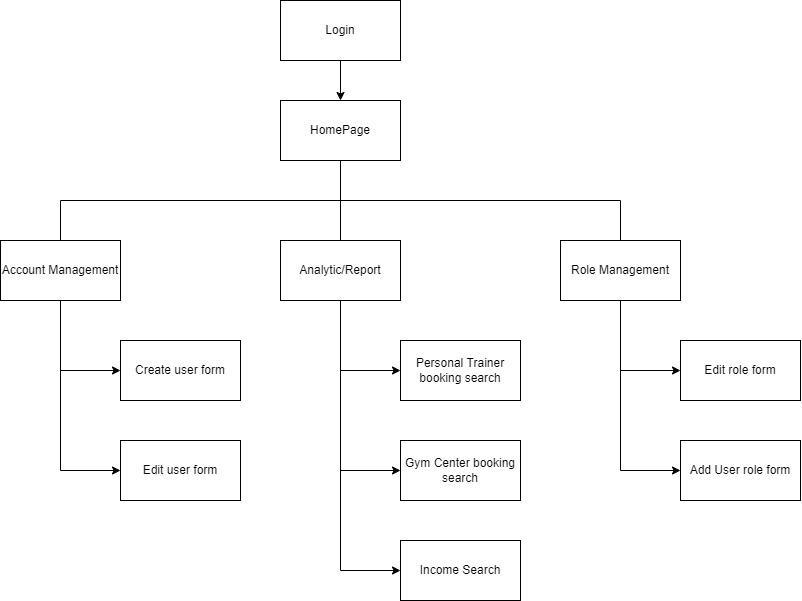
## Staff Part



## Customer Part

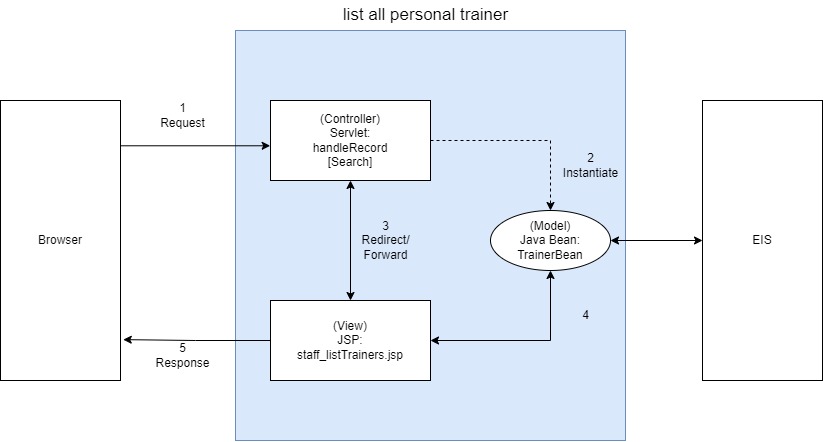


## Senior Management Part

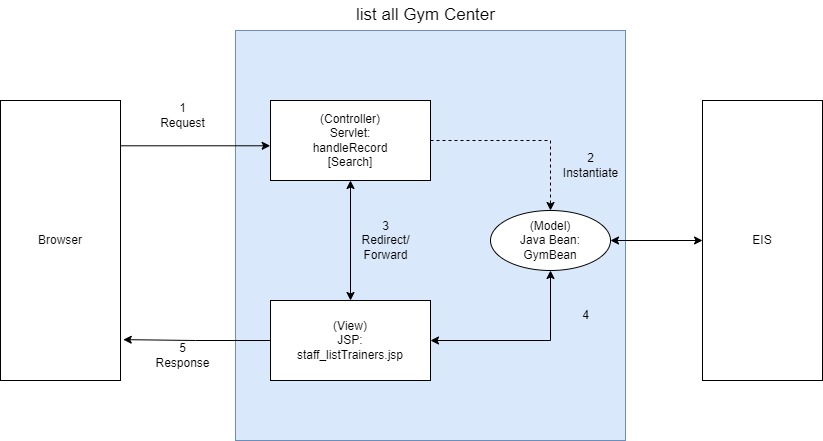


# System structure on how MVC Model is applied

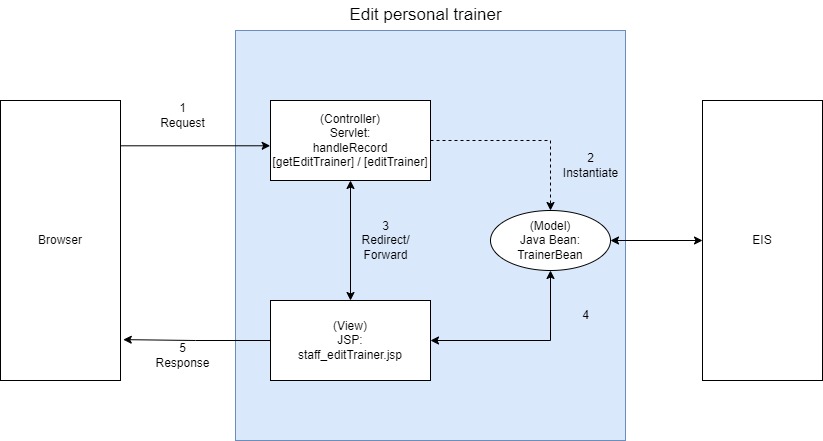
## Personal Trainer and Gym Center Management (for Personal Trainer)



Servlet passes the gym center list to the page to let the personal trainer or staff view all the personal trainer records.

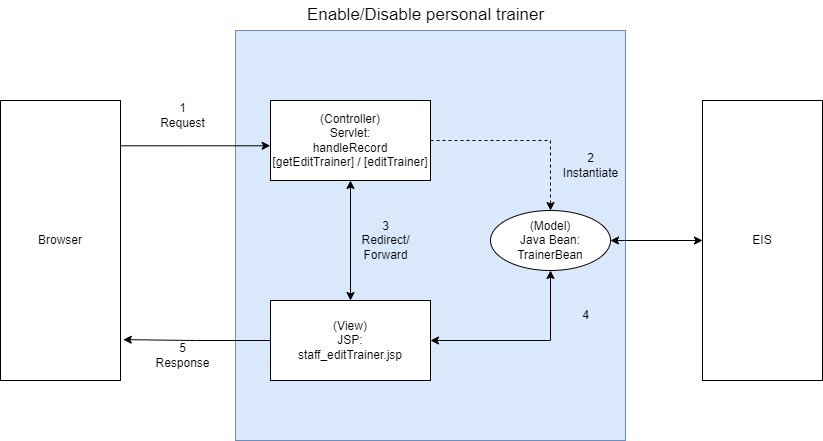


Servlet passes the gym center list to the page to let the personal trainer or staff view all the Gym Center records.



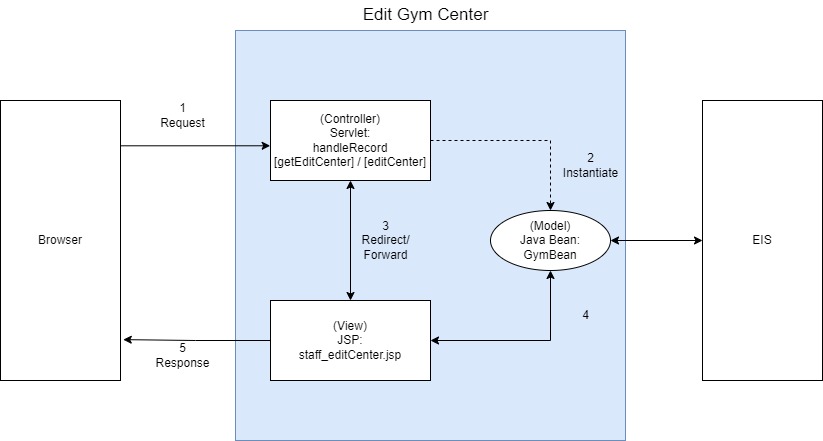
When servlet gets the id, it will search from the database to create a new trainer bean and pass the bean to the view to display the data.

When servlet receives a trainer modification, it will create a new trainer Bean and use it to modify the database record.



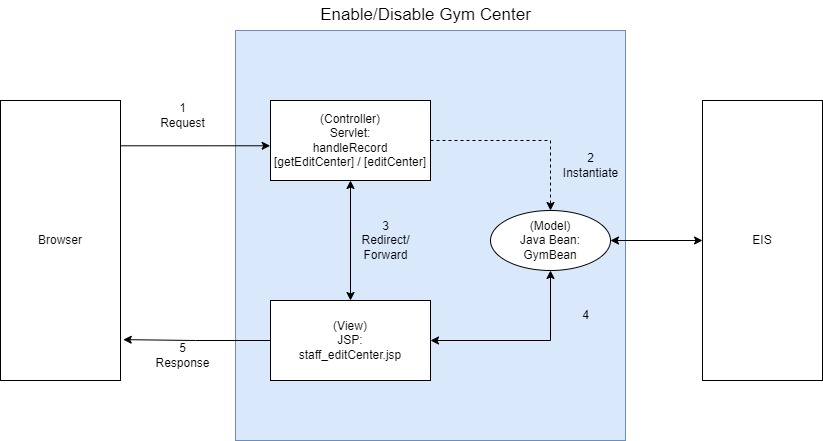
When servlet gets the id, it will search from the database to create a new trainer bean and pass the bean to the view to display the enable status.

When the servlet receives a trainer modification, it will create a new trainer Bean and use it to modify the trainer status.



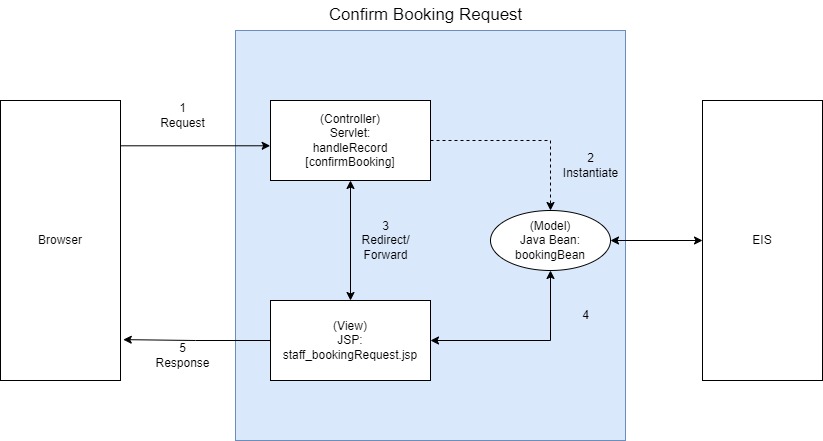
When servlet gets the id, it will search from the database to create a new Gym center bean and pass the bean to the view to display the data.

When servlet receives a trainer modification, it will create a new Gym center Bean and use it to modify the database record.

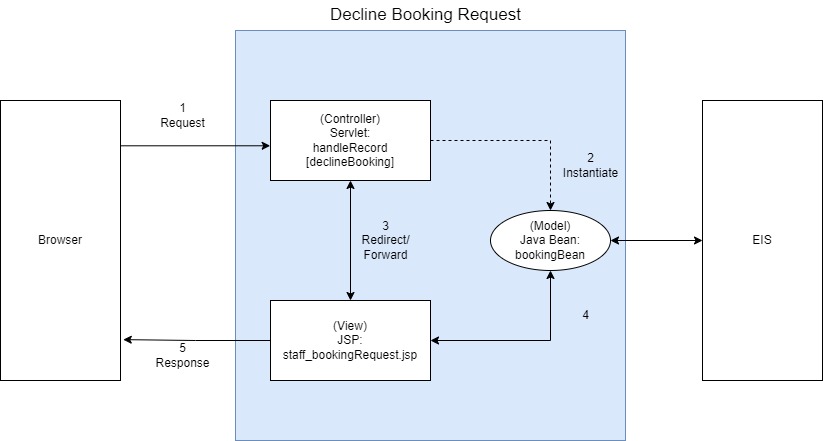


When servlet gets the id, it will search from the database to create a new Gym center bean and pass the bean to the view to display the enable status.

When the servlet receives a trainer modification, it will create a new Gym center Bean and use it to modify the trainer status.



When the servlet receives a booking confirmed operation, it will create a new booking Bean and use it to modify the booking status.



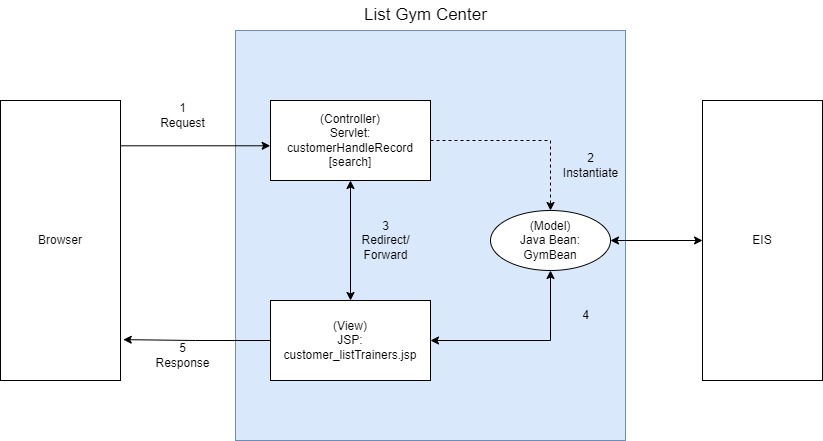
When the servlet receives a booking declined operation, it will create a new booking Bean and use it to modify the booking status.

## Personal trainer and Gym Venue Booking (for Customer)

Diagram

Description automatically generated

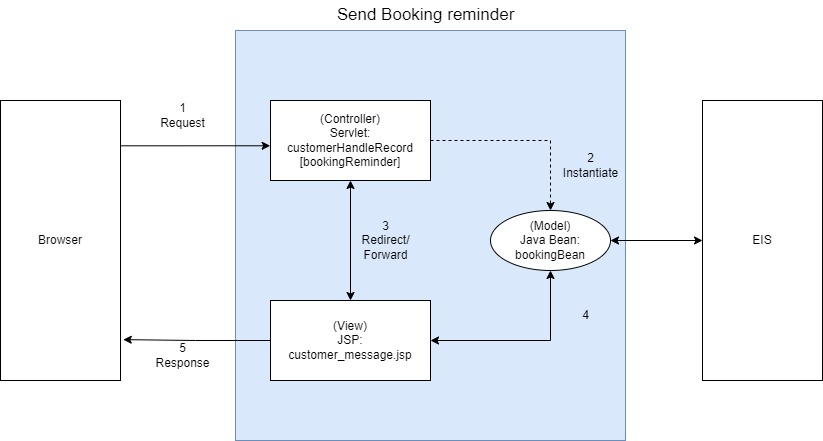
Servlet passes the gym center list to the page to let the customer view all the personal trainer records.



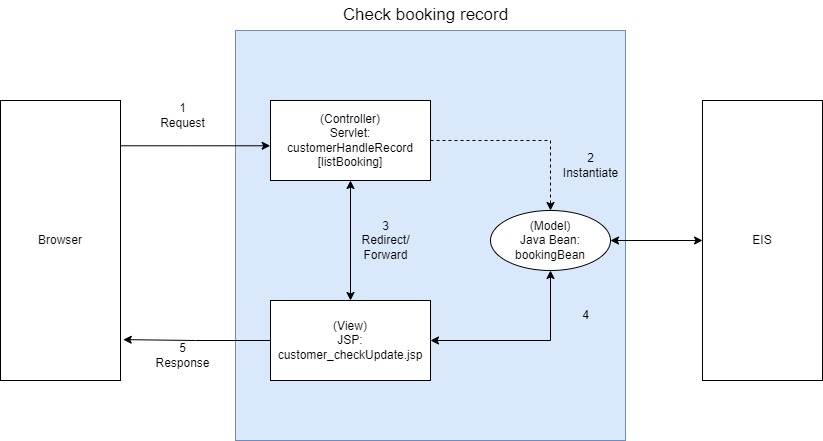
Servlet passes the gym center list to the page to let the customer view all the gym center records.

Diagram

Description automatically generated

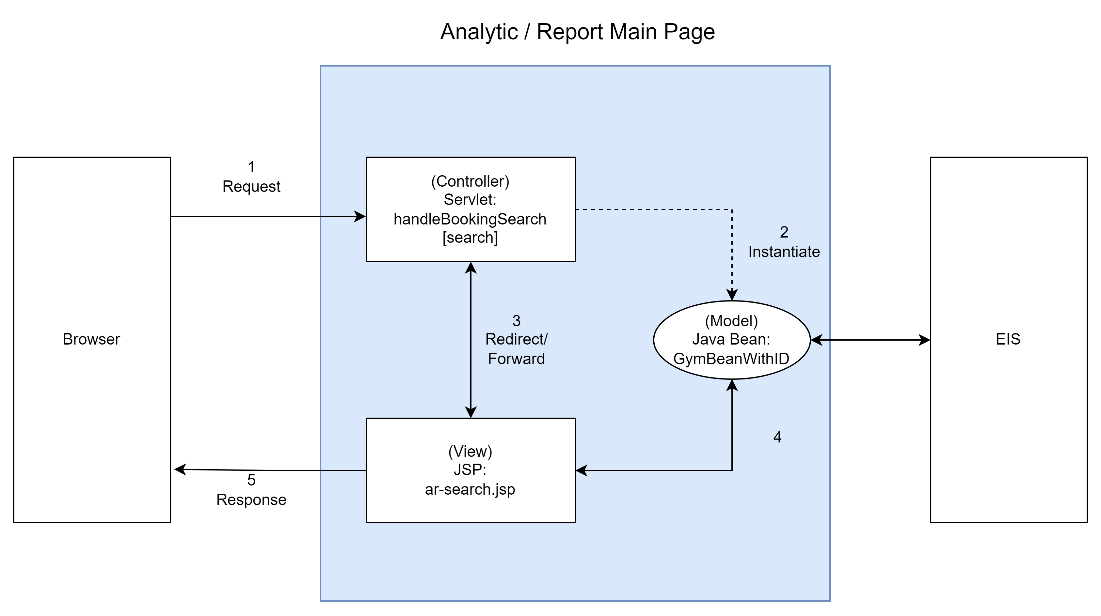
Servlet passes the booking list to the page to let the customer view all the booking records.

Servlet passes the booking reminder data to the page to let the customer view the booking reminder.

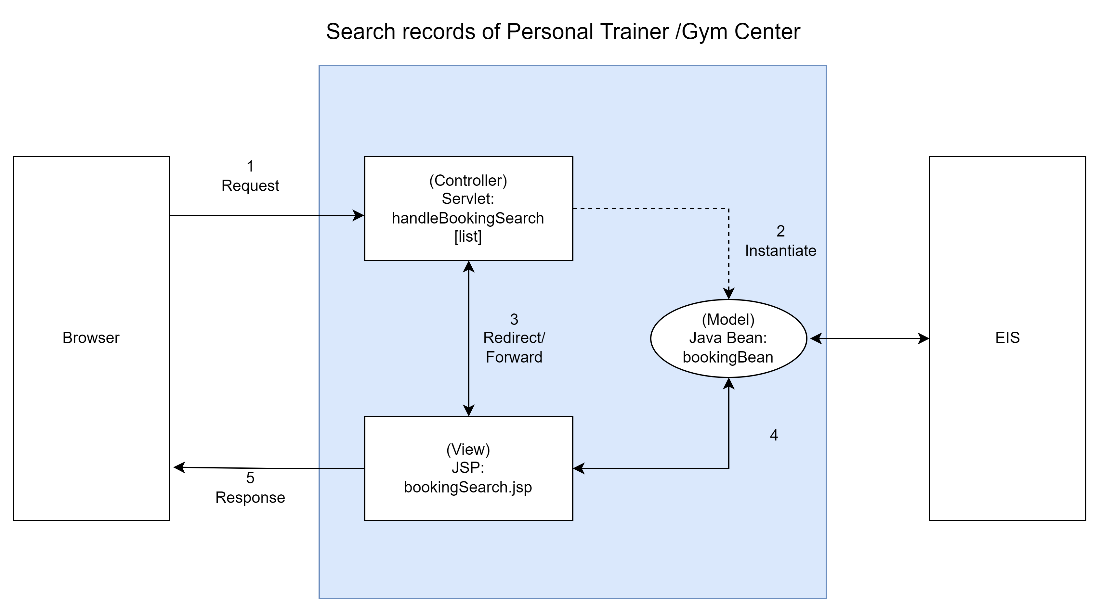


Servlet passes the booking record data to the page to let the customer view the booking record.

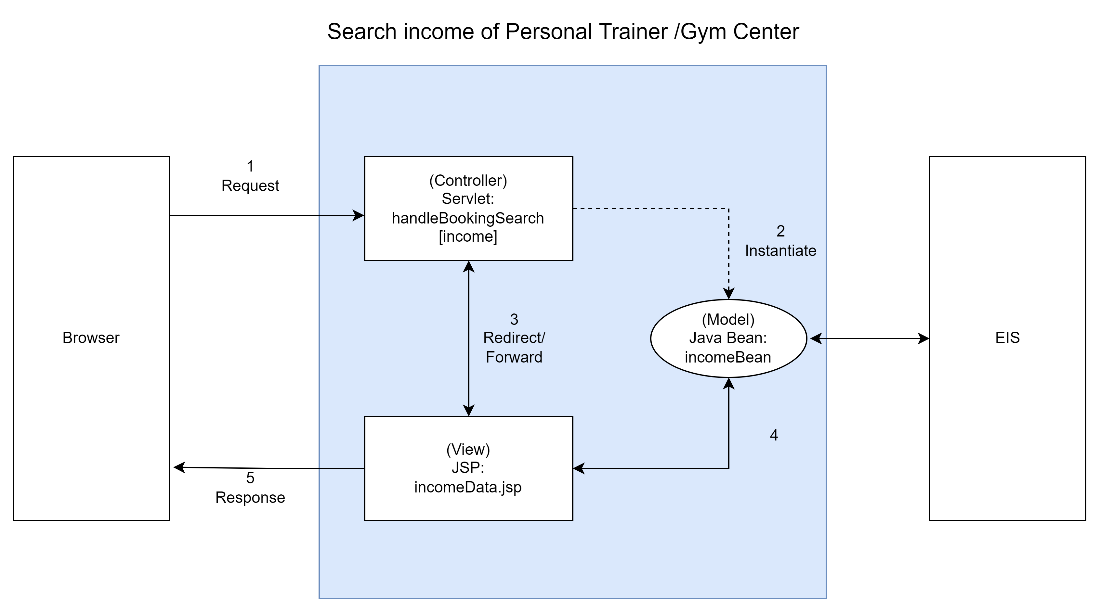
## Analytic / Report (for Senior Management)



Servlet pass the gym center list to the page to let user to choose gym center.

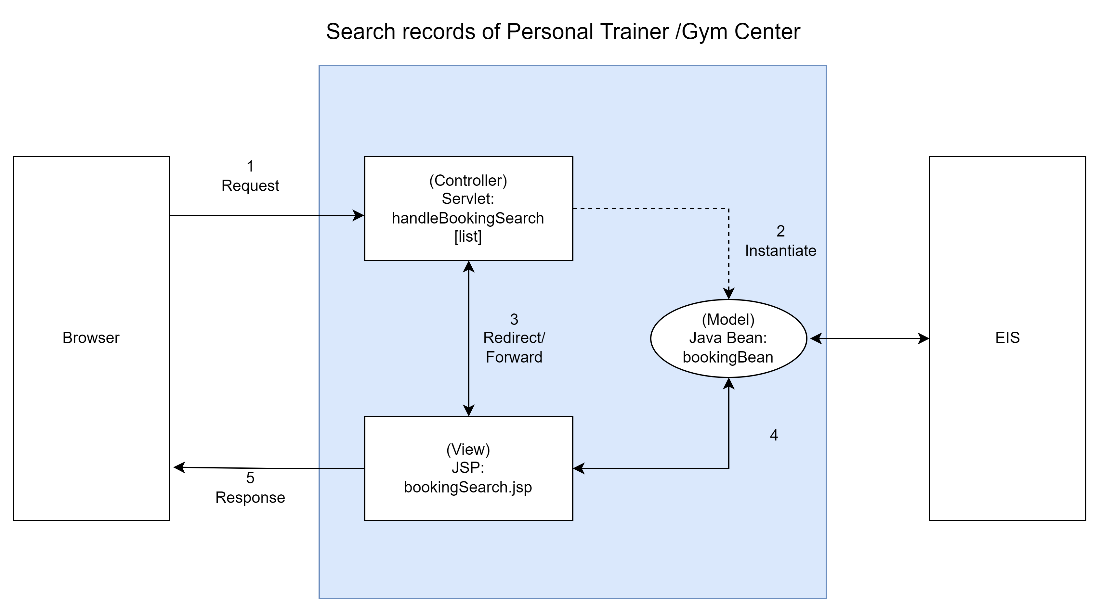


The servlet passes the related booking record to the page to list record selected center/trainer as a table.

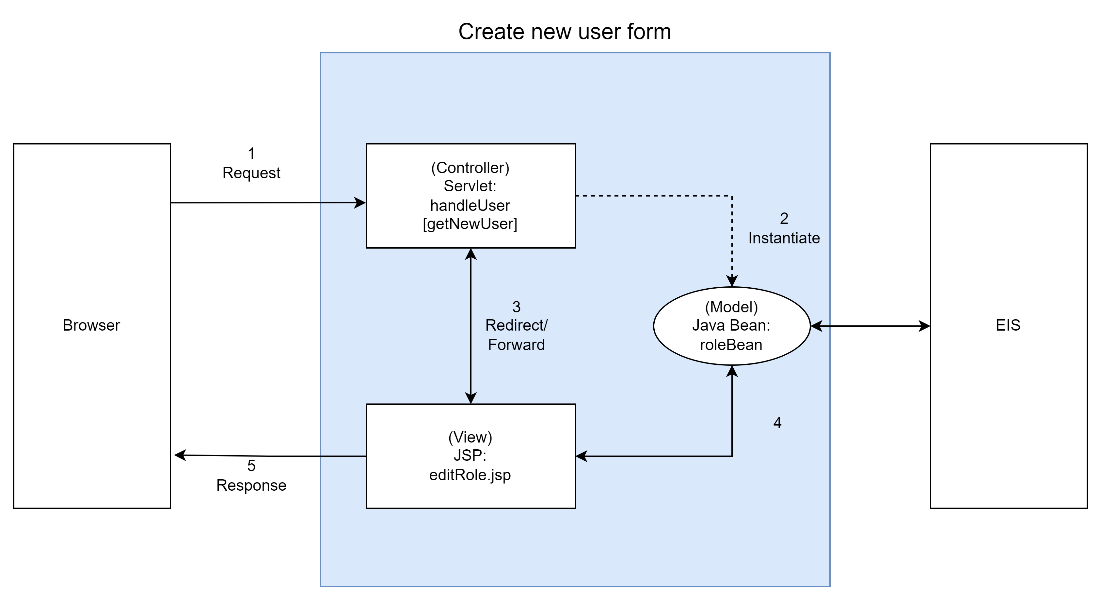


The servlet passes the income data record to the page to show income of each center/trainer at a specific time.

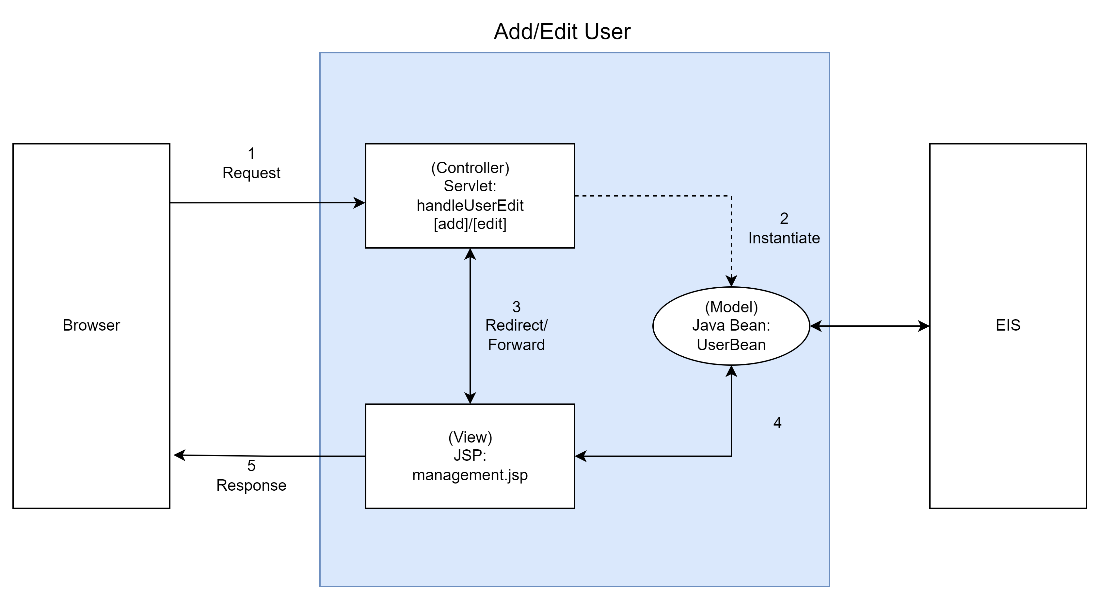
## Account Management (for Senior Management)



The servlet passes the user data record to the page to show all user data.

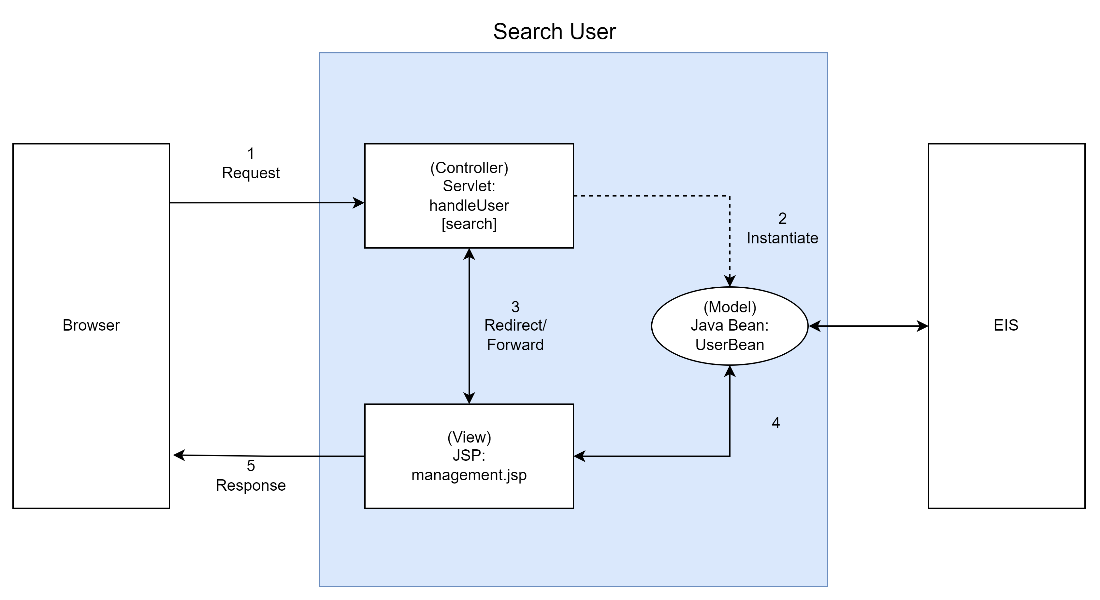


The servlet passes all roles to the page to let user choose what role the new account has.

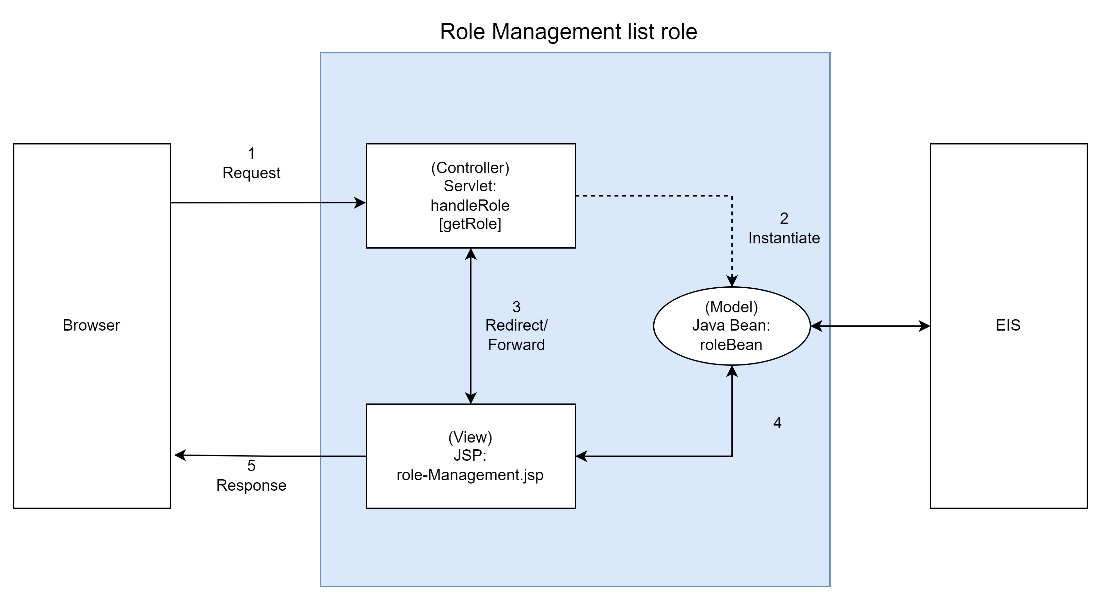


For add user, the servlet passes the new user data record to the page to the user list page.

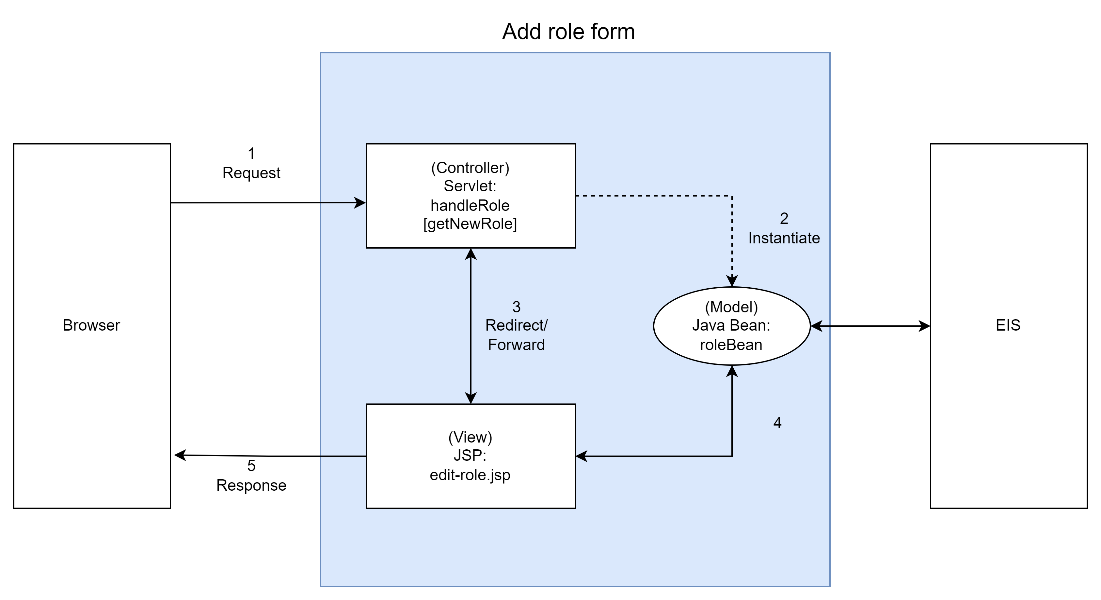
For edit user, the servlet passes the edited user record to the page to the user list page.



The servlet passes all the related user record to the page to let user view/edit/delete.



The servlet passes all roles to the page to let user view and do management.



The servlet passes all roles to the page form to let user choose what role the new account has.

# Database structure

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **booking Table** | | | | | | |
| **Element or value display name** | **Description** | **Data type** | **Character length** | **Acceptable Values** | **Required? (Function required)** | **Accepts null value?** |
| **id** | The unique Id of each booking | int | 20 | Must integer | Y | N |
| **date** | The booking date of each booking | date | X | Year-month-day | Y | N |
| **time** | The booking time of each booking | time | X | Hour: minute: second | Y | N |
| **type** | The Type of each booking | varchar | 20 | “trainer” or “center” | Y | N |
| **typeid** | The trainer or center id of each booking | varchar | 20 | No constraint | Y | N |
| **custid** | The customer id of each booking | int | 20 | No constraint | Y | N |
| **status** | The status of each booking | varchar | 20 | “confirm”, “decline” or “waiting” | Y | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Gym centre Table** | | | | | | |
| **Element or value display name** | **Description** | **Data type** | **Character length** | **Acceptable Values** | **Required? (Function required)** | **Accepts null value?** |
| **id** | The unique Id of each gym center | int | 20 | Must integer | Y | N |
| **centerName** | The center Name of each center | varchar | 20 | No constraint | Y | N |
| **description** | The description of each center | varchar | 20 | No constraint | Y | N |
| **states** | The states of each center | varchar | 20 | “enable” or “disable” | Y | N |
| **hrRate** | The hourly rate of each center | int | 20 | No constraint | Y | N |
| **imageID** | The image ID of each center | int | 20 | No constraint | Y | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Role Table** | | | | | | |
| **Element or value display name** | **Description** | **Data type** | **Character length** | **Acceptable Values** | **Required? (Function required)** | **Accepts null value?** |
| **id** | The role Id of each role | int | 20 | Must integer | Y | N |
| **name** | The name of each role | varchar | 20 | No constraint | Y | N |
| **Description** | The description of each role | varchar | 20 | No constraint | Y | N |
| **Type** | The type of each role | varchar | 20 | “trainer”, “staff”, “management” and “customer” | Y | N |
| **enable** | The enable of each role | varchar | 20 | No constraint | Y | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Role Table** | | | | | | |
| **Element or value display name** | **Description** | **Data type** | **Character length** | **Acceptable Values** | **Required? (Function required)** | **Accepts null value?** |
| **id** | The role Id of each role | int | 11 | Must integer | Y | N |
| **name** | The name of each role | varchar | 11 | No constraint | Y | N |
| **Description** | The description of each role | text | X | No constraint | Y | N |
| **Type** | The type of each role | varchar | 20 | “trainer”, “staff”, “management” and “customer” | Y | N |
| **enable** | The enable of each role | varchar | 20 | No constraint | Y | N |

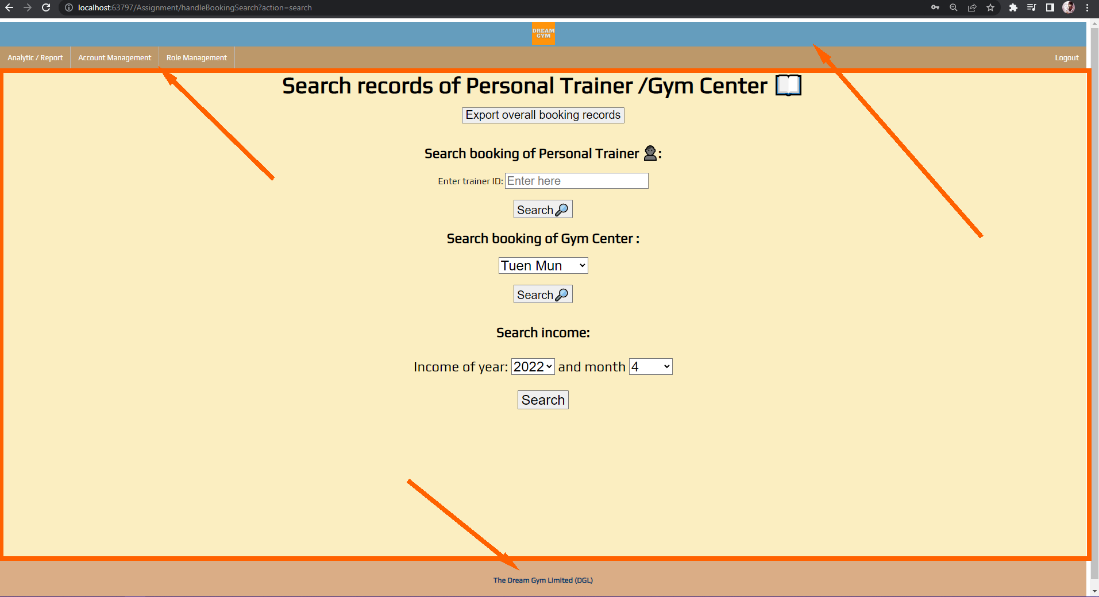
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Role Type Table** | | | | | | |
| **Element or value display name** | **Description** | **Data type** | **Character length** | **Acceptable Values** | **Required? (Function required)** | **Accepts null value?** |
| **id** | The role type Id of each role | int | 11 | Must integer | Y | N |
| **name** | The role type name of each role | varchar | 20 | No constraint | Y | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Trainer Table** | | | | | | |
| **Element or value display name** | **Description** | **Data type** | **Character length** | **Acceptable Values** | **Required? (Function required)** | **Accepts null value?** |
| **Id** | The Id of each Trainer | int | 11 | Must integer | Y | N |
| **name** | The name of each Trainer | varchar | 20 | No constraint | Y | N |
| **states** | The states of each Trainer | varchar | 20 | No constraint | Y | N |
| **hrRate** | The hourly rate of each Trainer | int | 20 | Must integer | Y | N |
| **Description** | The description of each Trainer | varchar | 100 | No constraint | Y | N |
| **sportType** | The sport Type of each Trainer | varchar | 100 | No constraint | Y | N |
| **imageID** | The image ID of each Trainer | varchar | 20 | No constraint | Y | N |

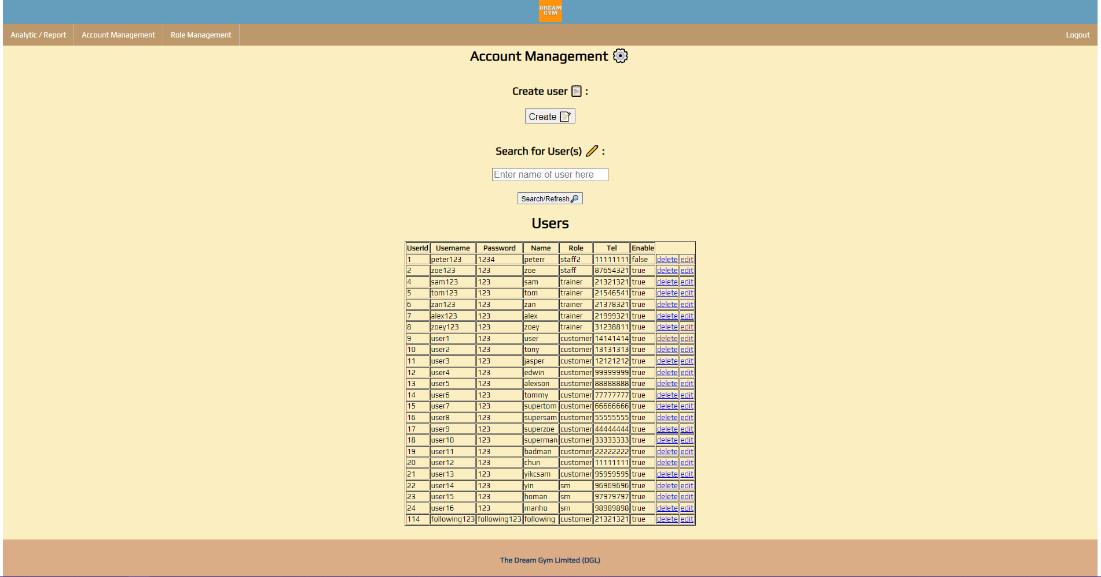
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Table** | | | | | | |
| **Element or value display name** | **Description** | **Data type** | **Character length** | **Acceptable Values** | **Required? (Function required)** | **Accepts null value?** |
| **userid** | The Id of each user | int | 11 | Must integer | Y | N |
| **username** | The username of each user | varchar | 20 | No constraint | Y | N |
| **Password** | The password of each user | varchar | 20 | No constraint | Y | N |
| **Name** | The name of each user | varchar | 20 | Must integer | Y | N |
| **Role** | The role of each user | varchar | 20 | No constraint | Y | N |
| **Tel** | The telephone number of each user | int | 8 | Must integer | Y | N |
| **enable** | The status of each user | varchar | 20 | No constraint | Y | N |

# Brief major characteristics and design of the application

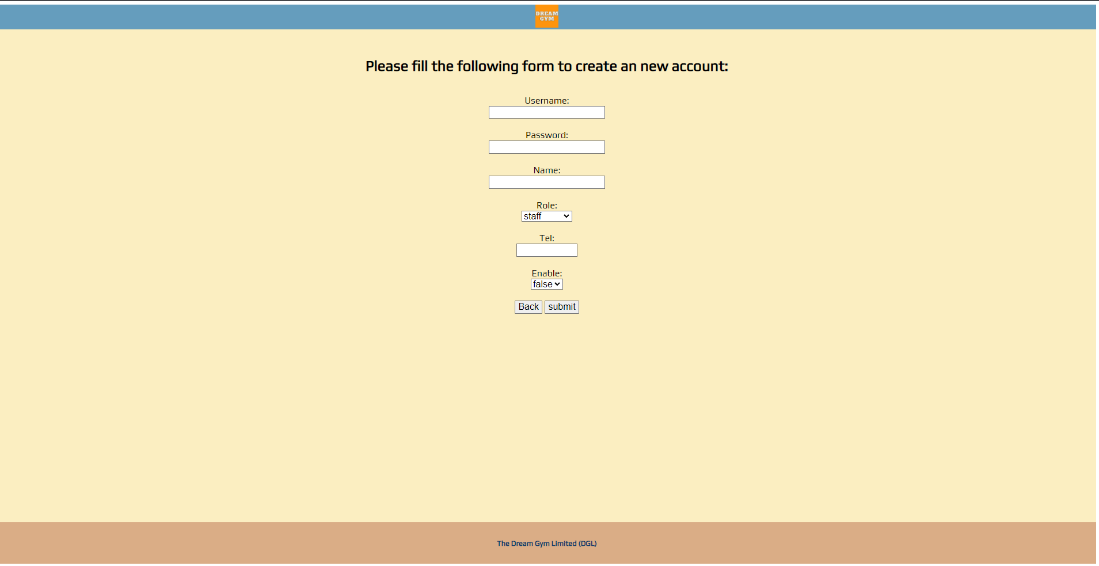
The website design consists of 3-4 parts. Which includes the header, menu bar, page content and the footer.



Our website will use some tables to show different records.



Our website will have some forms to let user create/edit records.



For showing income, we will use income to show.



# Conclusions

There are three objectives for this proposed project: To prevent error-prone handwritten records. To let management, do some analysis or report more easily; To improve gym center's and personal trainer's booking management, for staff, trainers and customers.

We achieved those objectives by writing a web system using JSP and the MVC technology. After The Dream Gym Limited (DGL) used our system, all the management/ searching/ booking work became quicker and easier. Users just need to login to the website to do what they want, and the paperwork and manpower are less to be used.

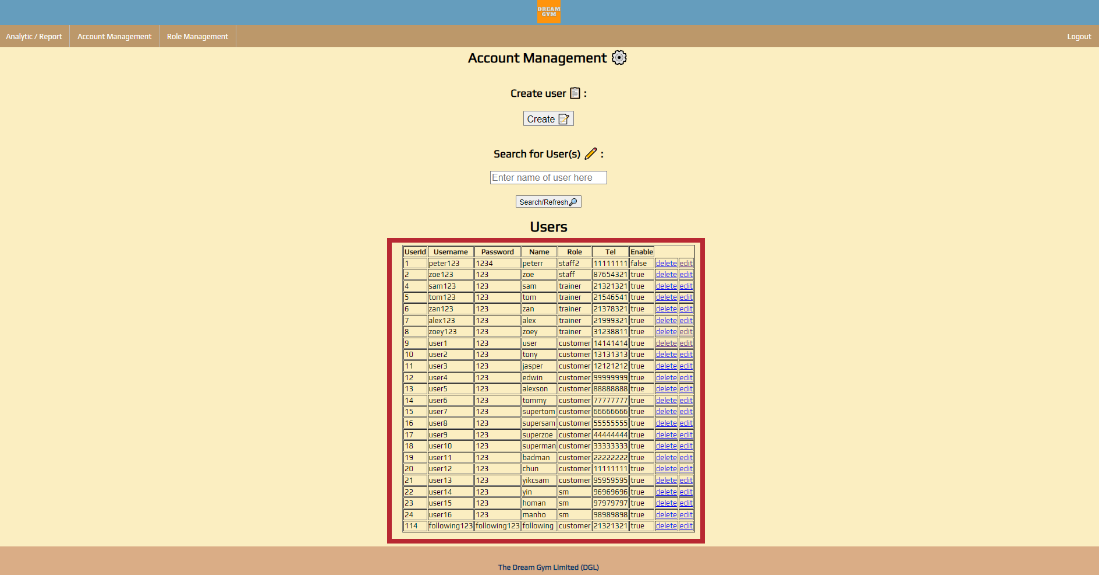
Now, the web-system is developed and achieved those objectives, the system will provide these functions to the users. For customers, the web system can do some searching and booking. For senior management, the web system can do some analysis and reporting, search booking records of the center and trainer, and even see the income and booking rate. Also, they can do some account management, create/edit/delete some accounts for the web system.

# Skill checklist

|  |  |
| --- | --- |
| Skills requirements |  |
| Use JSP/servlets to dynamically generate HTML pages | Skill used |
| Use JSP/servlets to accept user inputs from browser | Skill used |
| Use JSP Action | Skill used |
| Use Custom Tag (Taglib). | - |
| Use JavaBean | Skill used |
| Use JDBC for database connection | Skill used |
| Use session checking | Skill used |
| Use login control | Skill used |
| Apply the MVC model | Skill used |
| Other skills applied | - |

## Use JSP/servlets to dynamically generate HTML pages

The users' record table is generated with JSP.

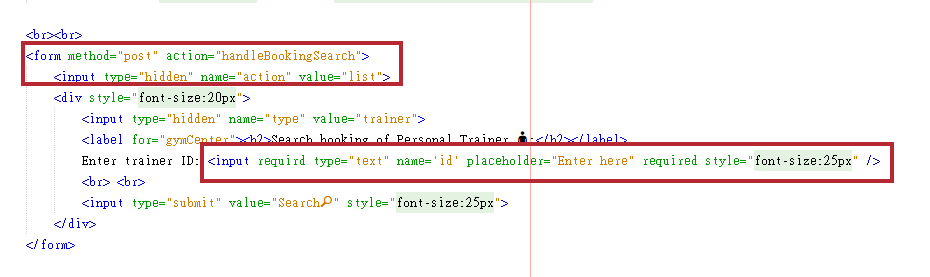


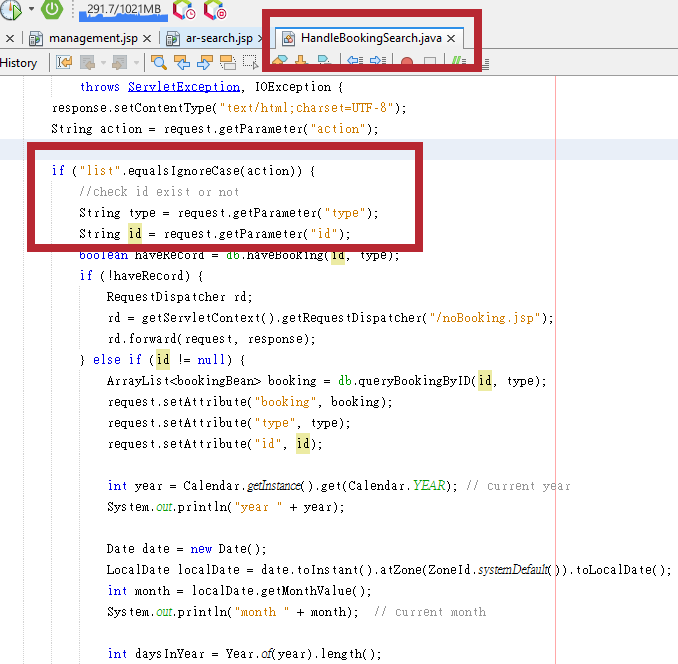


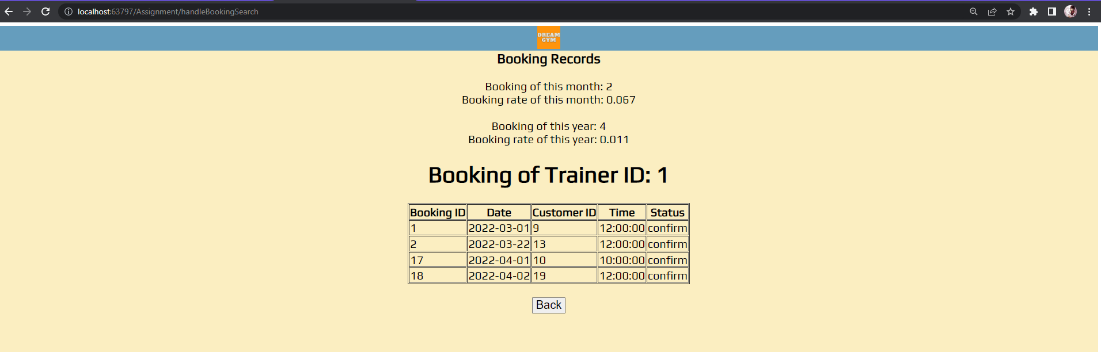
## Use JSP/servlets to accept user inputs from browser

We receive user input and use servlet to process the data and pass records to user.



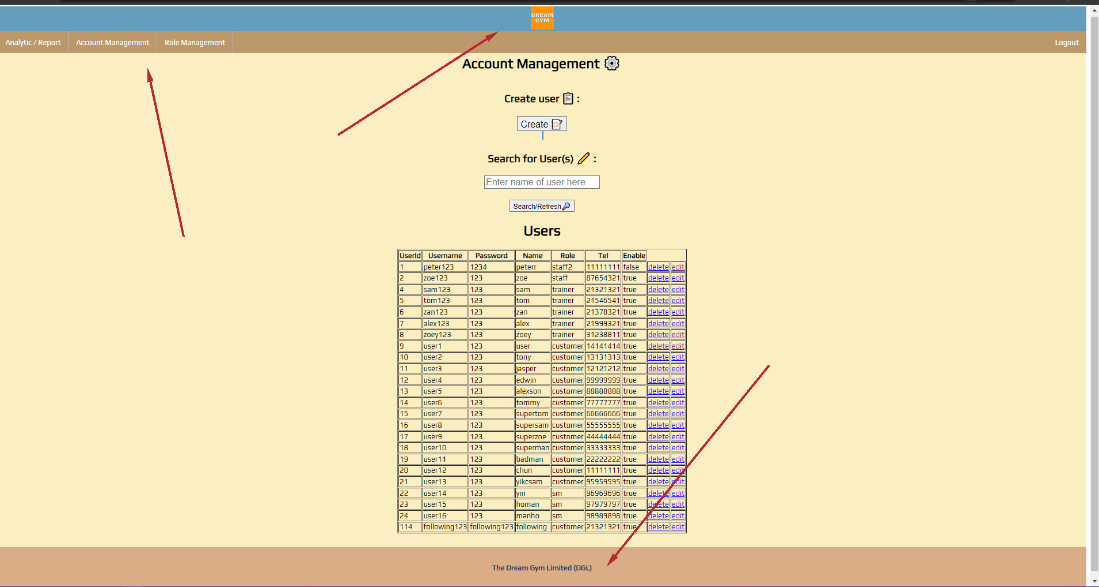


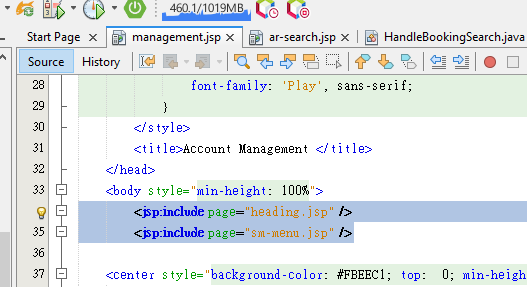




## Use JSP Action

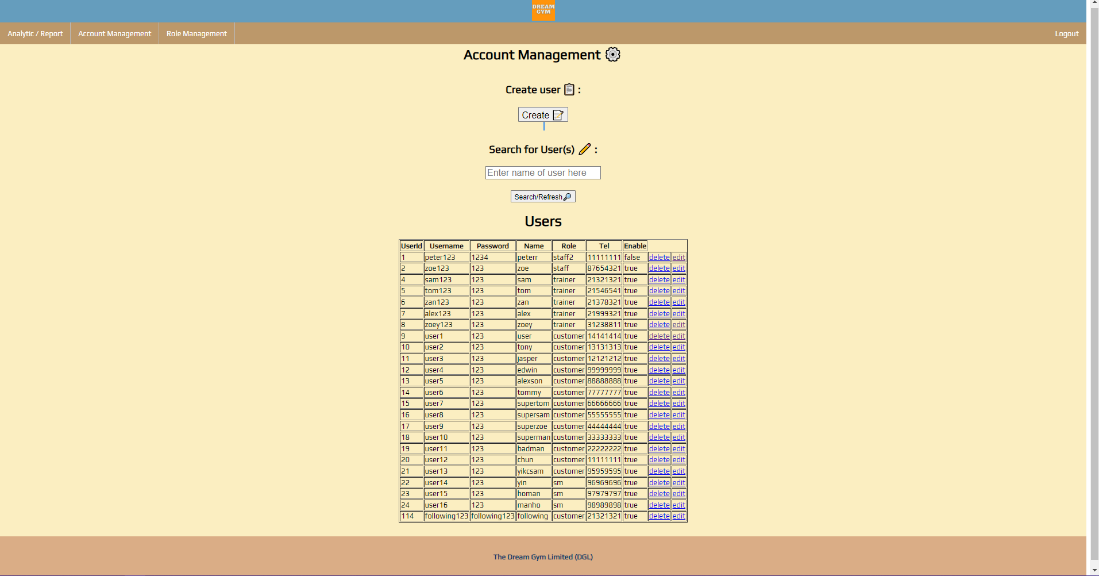
We use [jsp:include] to add header, menu and footer in every page.

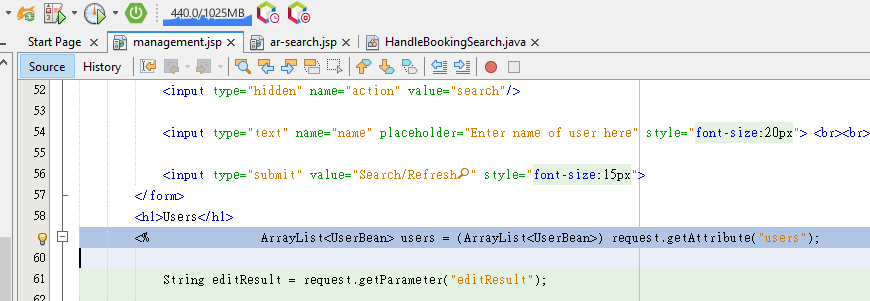


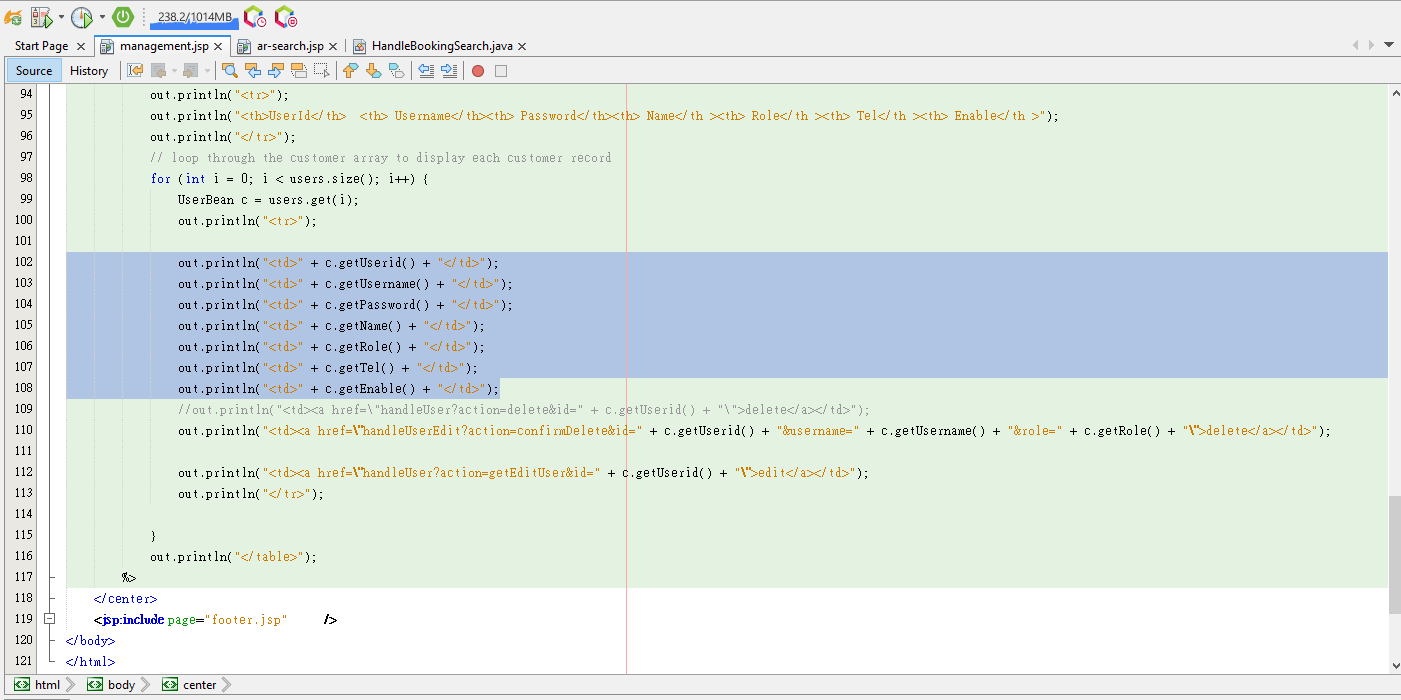


## Use JavaBean

We use JavaBean to get records and show it to users.

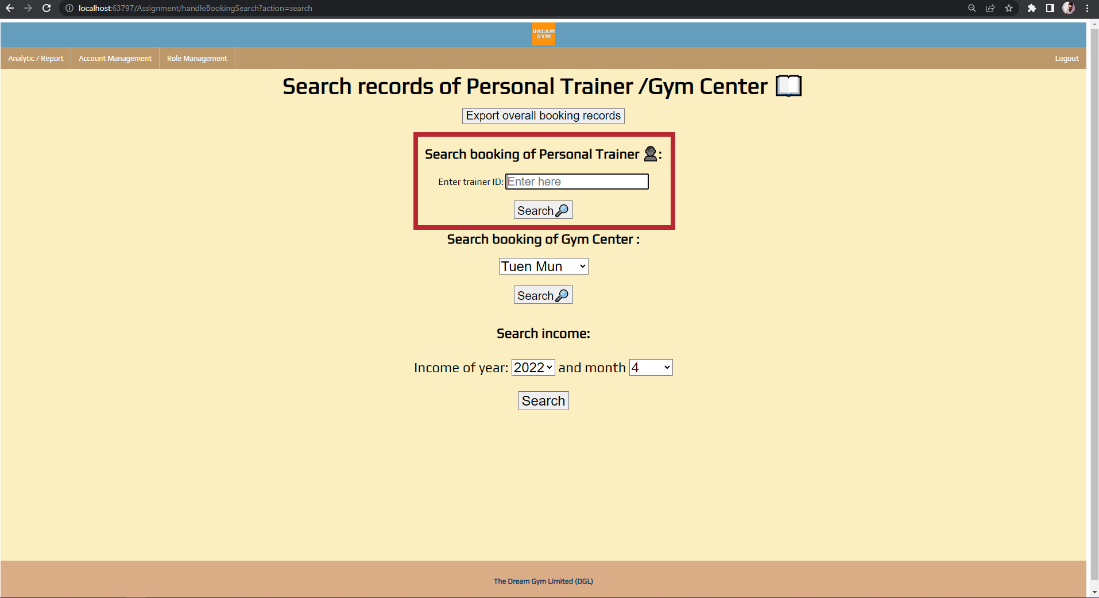


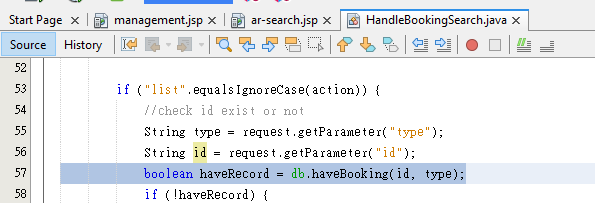


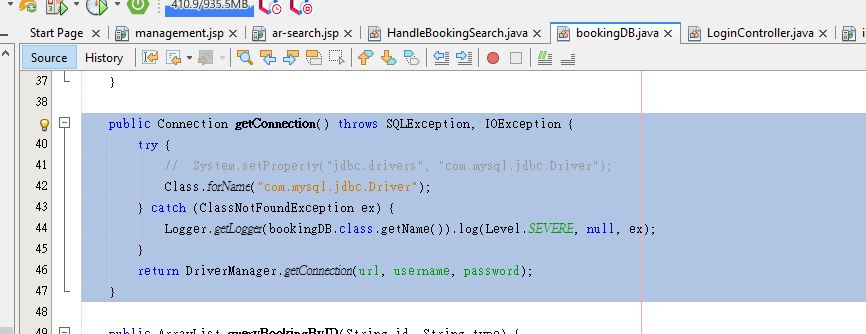


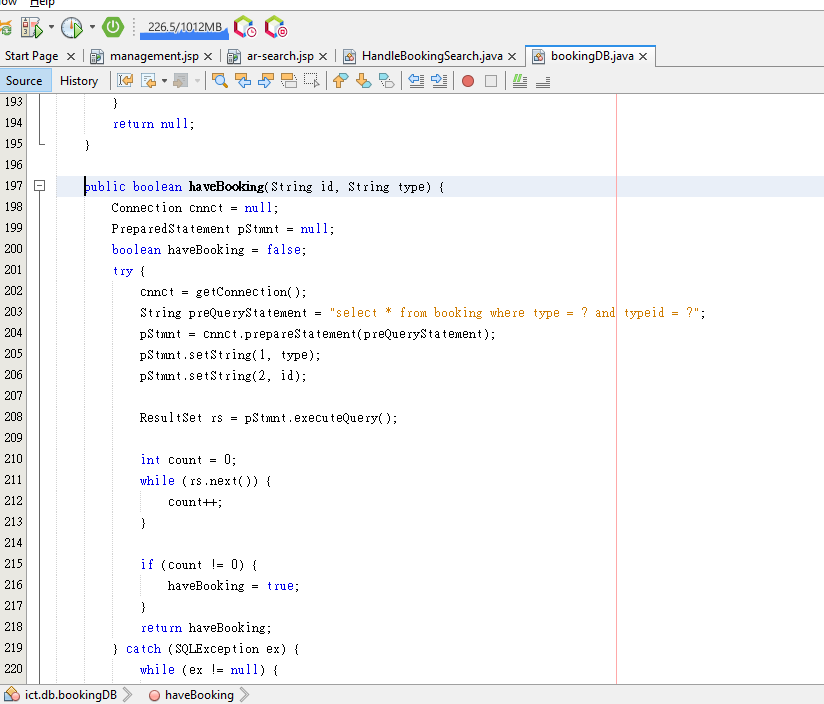
## Use JDBC for database connection

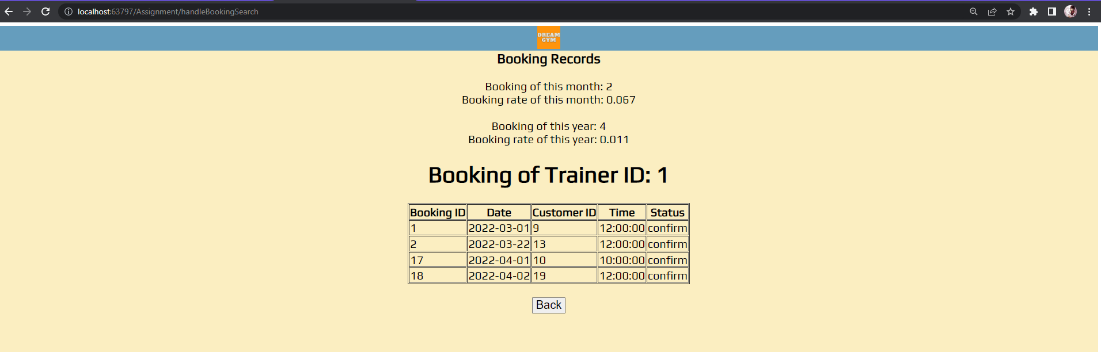
We use JDBC to connect database to get records and show it to users.







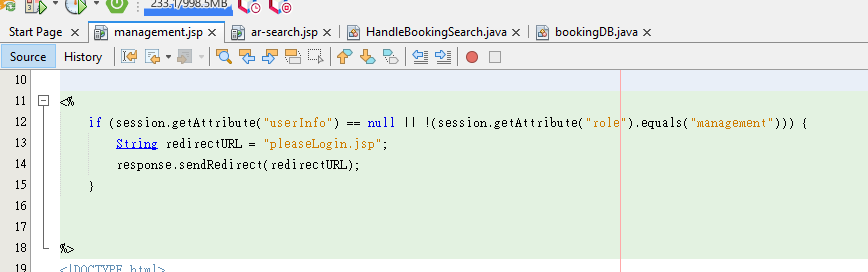




## Use session checking

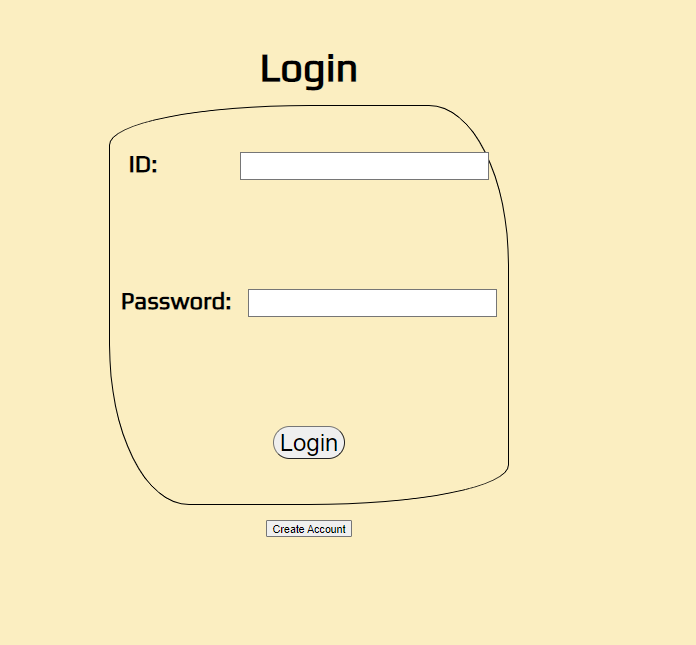
We use session checking to check if user logged in or not. And check if they can visit that page.

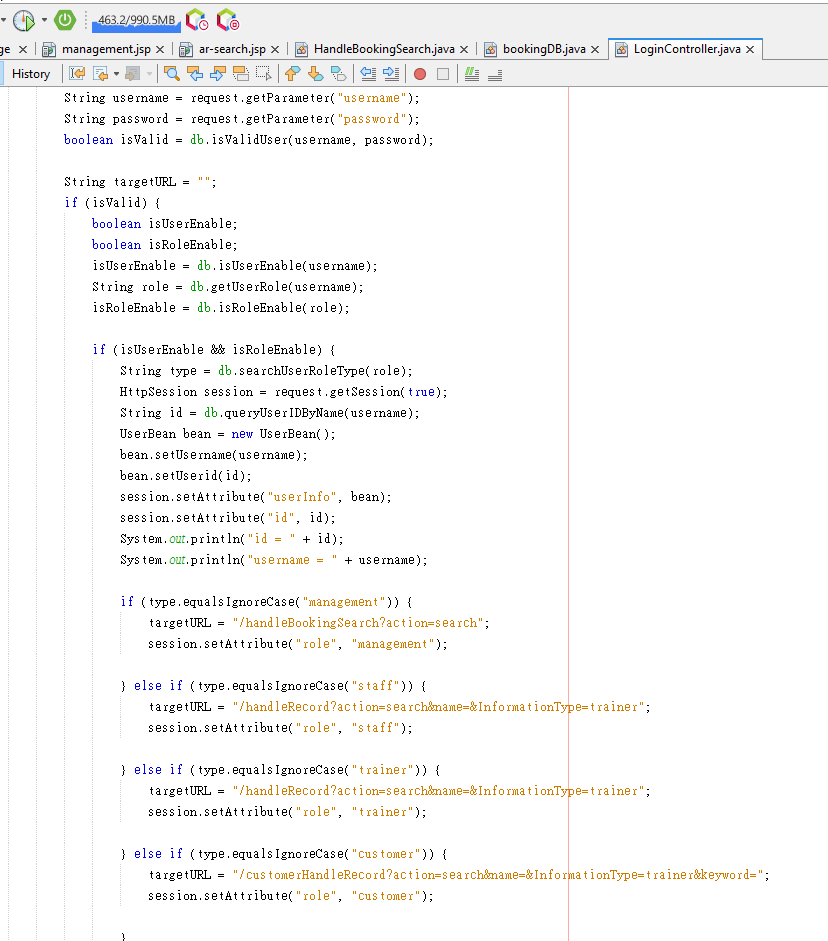




## Use login control

We use login control to let different accounts go to different pages.



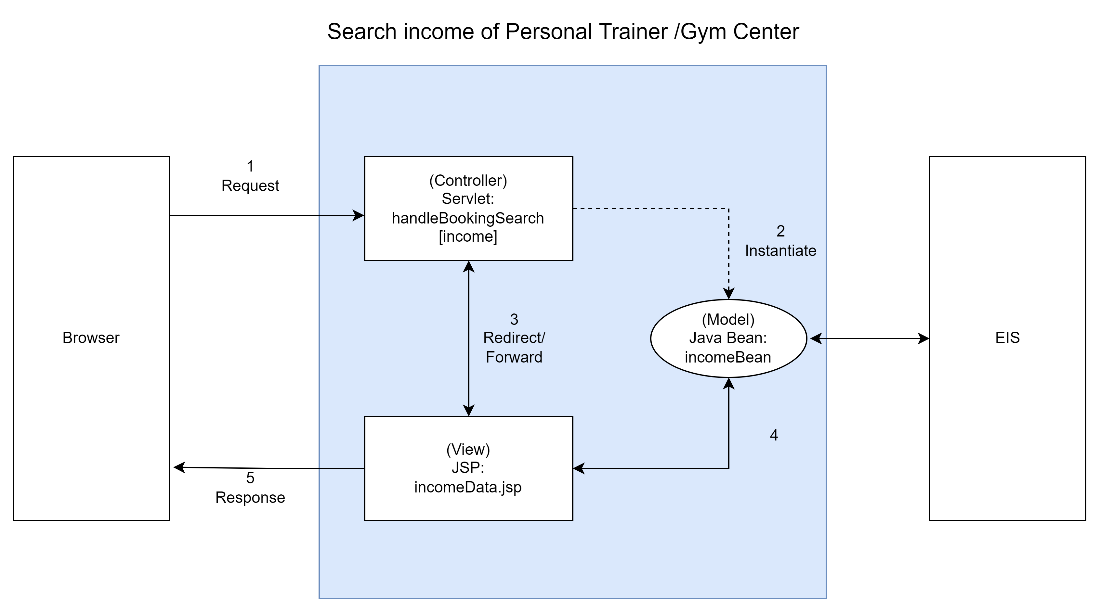


## Apply the MVC model

We applied the MVC model in our page to do some view/edit/delete function.







## Other skills applied

### Google chart

We use google chart to show some data.



### JavaScript- Table to CSV (found on internet)

We use JavaScript to save a table as a csv file.



