Project Name “Fitness – Online Gym Management System”

Shuai Shao

Summary:

This project is to build a gym system regarding administrator management, supervisor management, gym member information update, booking sections, reviewing and rating, supplier and equipment management.

Functionalities:

- login into 4 different roles with their username, password and implement their own operations.
- Store data into MySQL database.
- Administrator, Supervisor, Member and Supplier have to login with their username and password according to the related values from MySQL database.
- Gym reviews, rate, avg rate can be found on main page.
- Supervisor, member and supplier accounts can be updated/deleted/added.
- Member can register accounts from home page.
- When registering, member username and password are restricted using regex.
- Member can rate and make reviews.
- Member can reserve date for going to the gym.
- Member can choose section and time when reserving.
- Supervisor can view all member and equipment information and update/add/delete them.
- Supervisor can search for member/equipment information according to name/ID.
- When member finishes gym section, supervisor can delete his reservation information.
- Member can view his information and update.
- Supplier can view equipment information and update/add/delete.
- Used interceptor and validation to filter data.
- All succeeded operations will lead to a “success” page.
- All failed operations will lead to a ‘failure” page.

Technology:

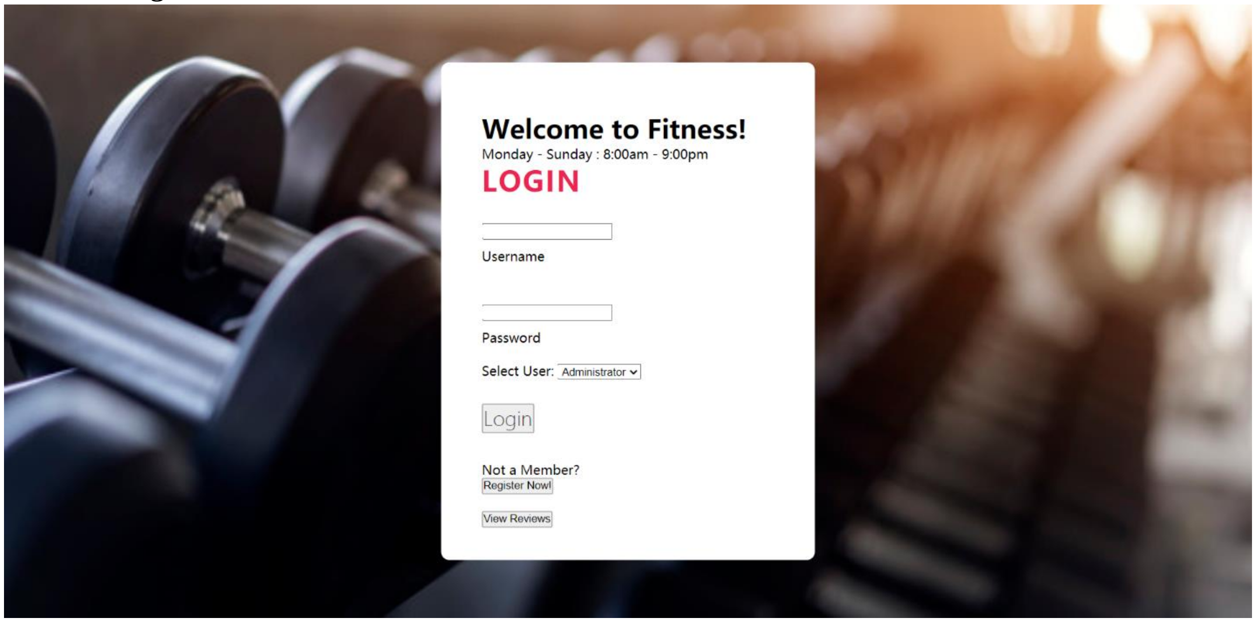
- IntelliJ, Spring, Spring MVC, Hibernate, MySQL, HTML, CSS, jQuery, interceptor, regex, etc.

Roles supported:

For this system, we have administrator role, supervisor role, member role and supplier role.

Tasks:

1. Main page:
   User can choose different account roles when login.
   User can view reviews.
   User can register as Member.



Welcome to Fitness!

Monday - Sunday : 8:00am - 9:00pm

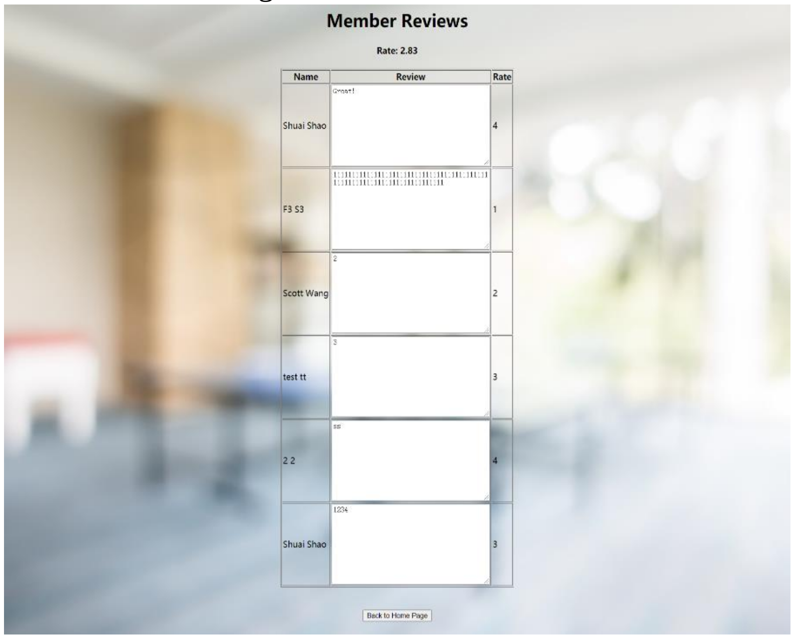
LOGIN

Username

Password

Select User: Administrator

2. View Reviews:
   Review, rate and avg rate are shown here.

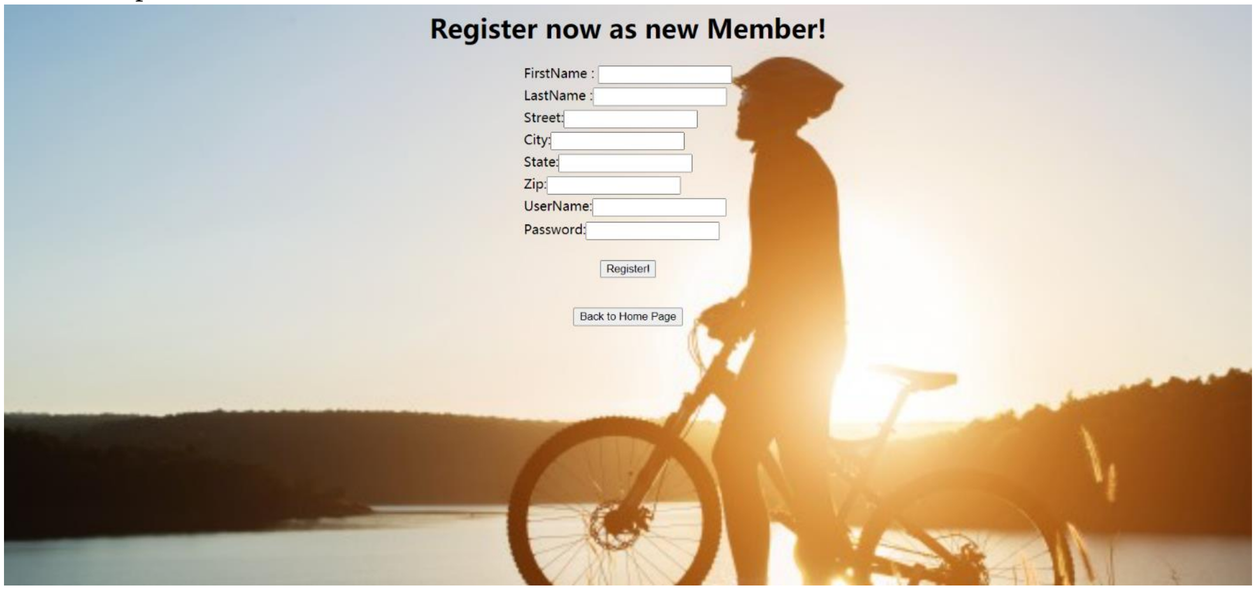


Member Reviews

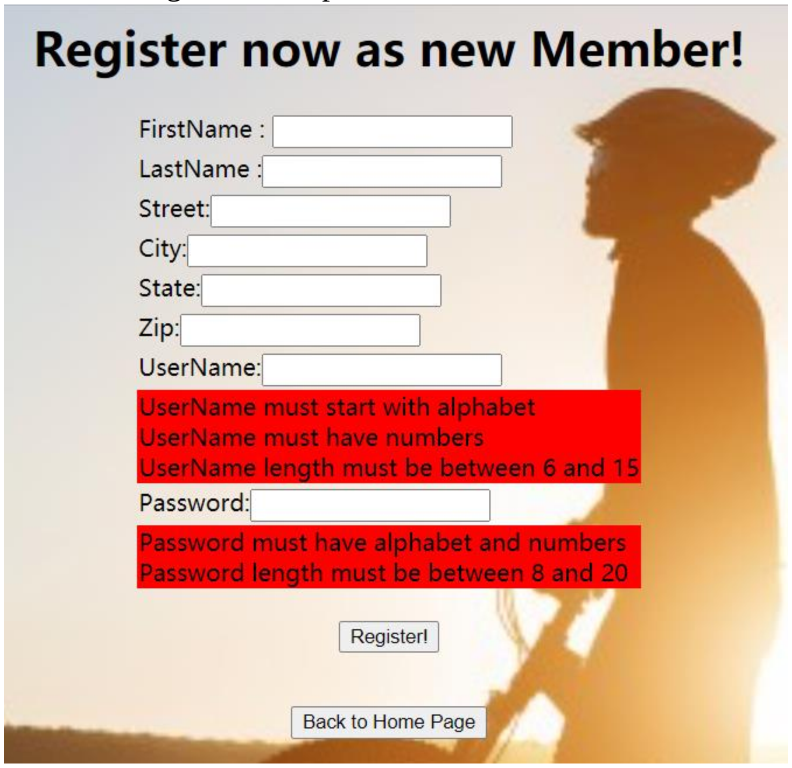
Rate: 2.83

Name	Review	Rate
Shuai Shao	Great!	4
F3 S3	[illegible]	1
Scott Wang	2	2
test tt	3	3
2 2	ss	4
Shuai Shao	1234	3

3. Member register:
   User can register as member here.
   Null and unsafe values will fail. Filtered by interceptor.
   When clicking on username/password, a notice will show up, informing username/password standard.



After clicking username/password:

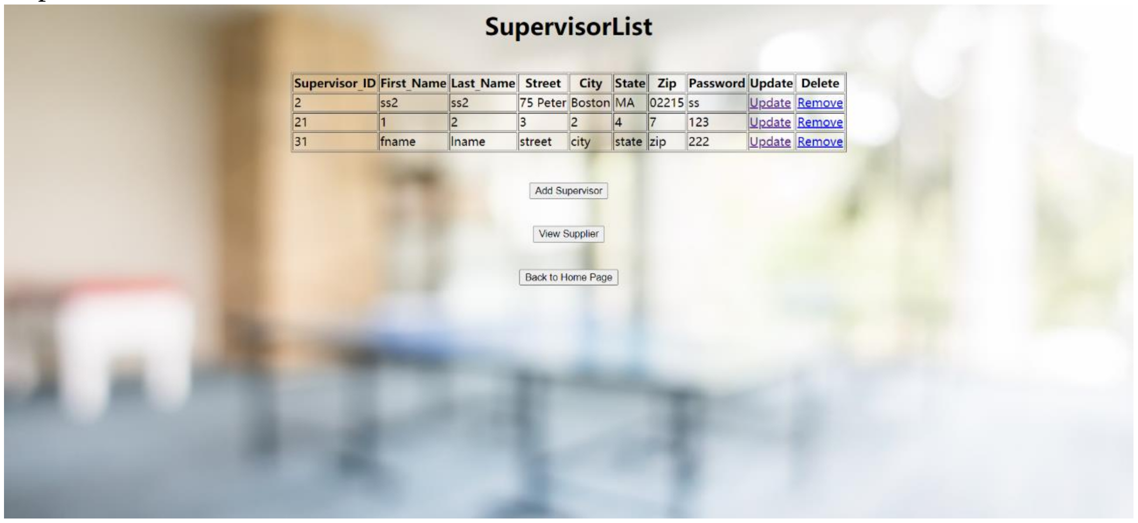


UserName must start with alphabet
UserName must have numbers
UserName length must be between 6 and 15

Password must have alphabet and numbers
Password length must be between 8 and 20

4. Administrator login:
   Admin login according to his/her username and password.
   Admin can view, add, update, delete supervisors. Null and unsafe values will fail. Filtered by interceptor.
   Admin can jump to supplier page, add, update and delete them. Null and unsafe values will fail. Filtered by interceptor.

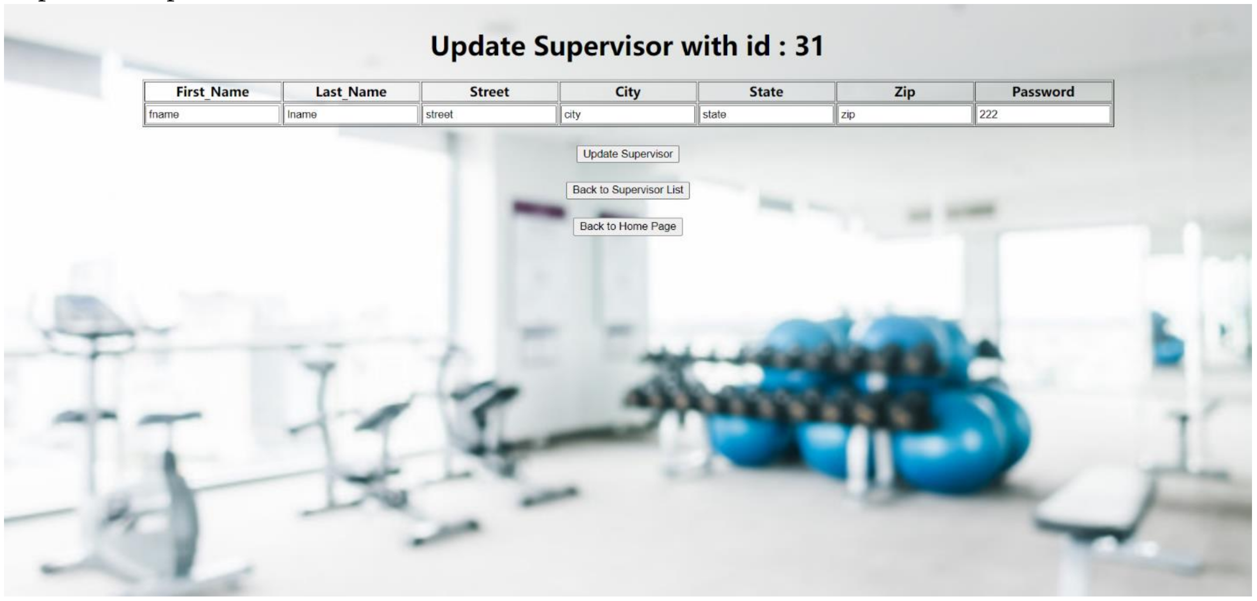
   Supervisor list:



SupervisorList

Supervisor_ID	First_Name	Last_Name	Street	City	State	Zip	Password	Update	Delete
2	ss2	ss2	75 Peter	Boston	MA	02215	ss	Update	Remove
21	1	2	3	2	4	7	123	Update	Remove
31	fname	lname	street	city	state	zip	222	Update	Remove

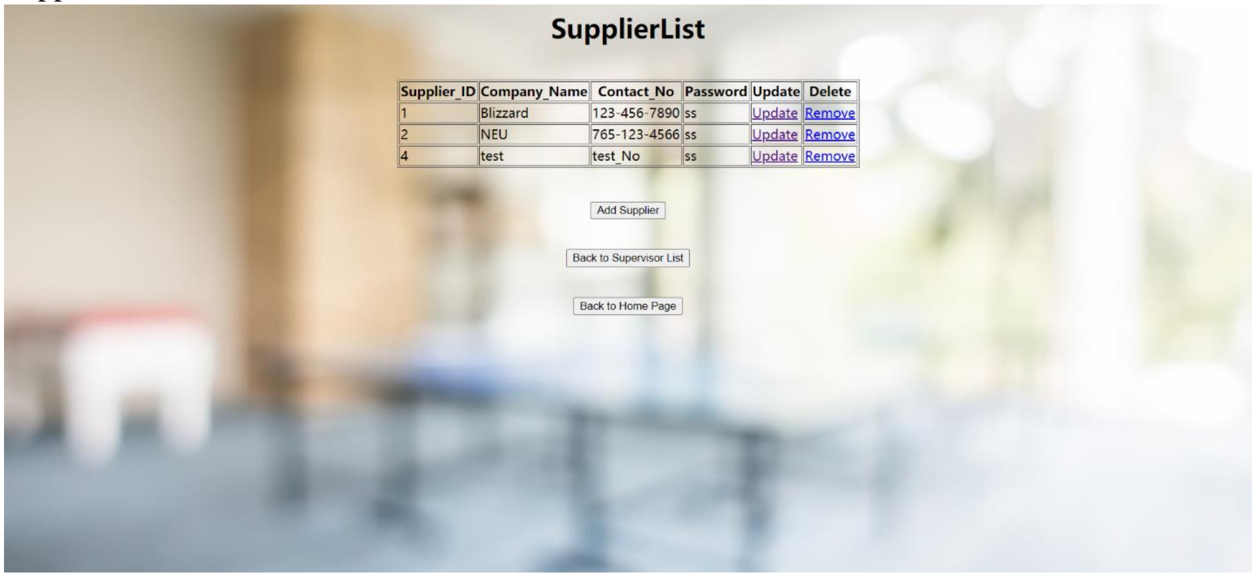
   Supervisor update:



Update Supervisor with id : 31

First_Name	Last_Name	Street	City	State	Zip	Password
fname	lname	street	city	state	zip	222

Supplier list:

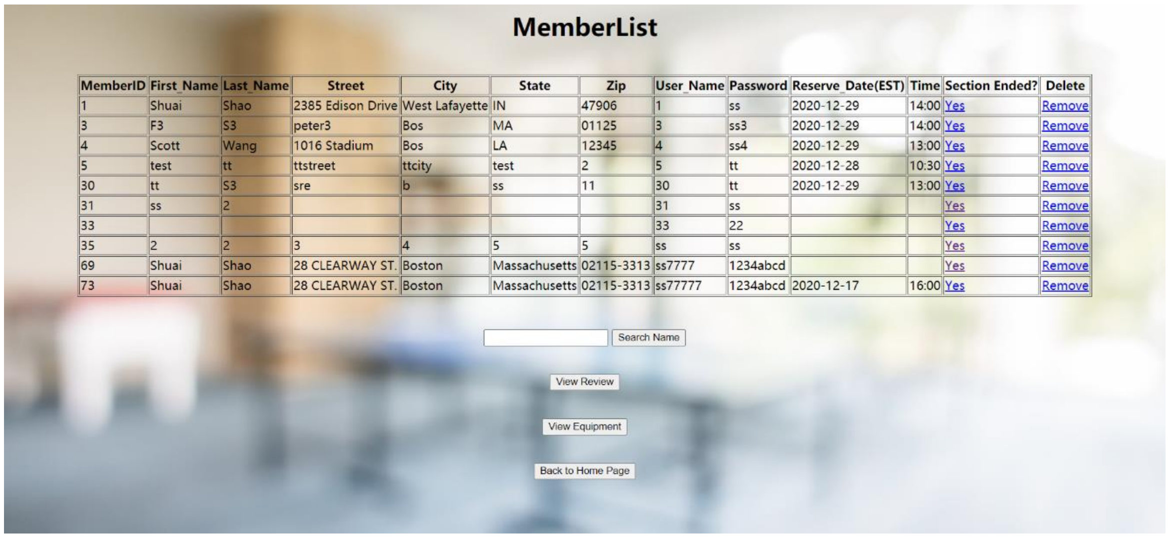


SupplierList

Supplier_ID	Company_Name	Contact_No	Password	Update	Delete
1	Blizzard	123-456-7890	ss	Update	Remove
2	NEU	765-123-4566	ss	Update	Remove
4	test	test_No	ss	Update	Remove

5. Supervisor login:
   Supervisor login according to his/her username and account.
   Supervisor can view, delete, search for member according to first/last name.
   Supervisor can view and delete equipment.
   Supervisor can remove member reservation info when reservation finishes.

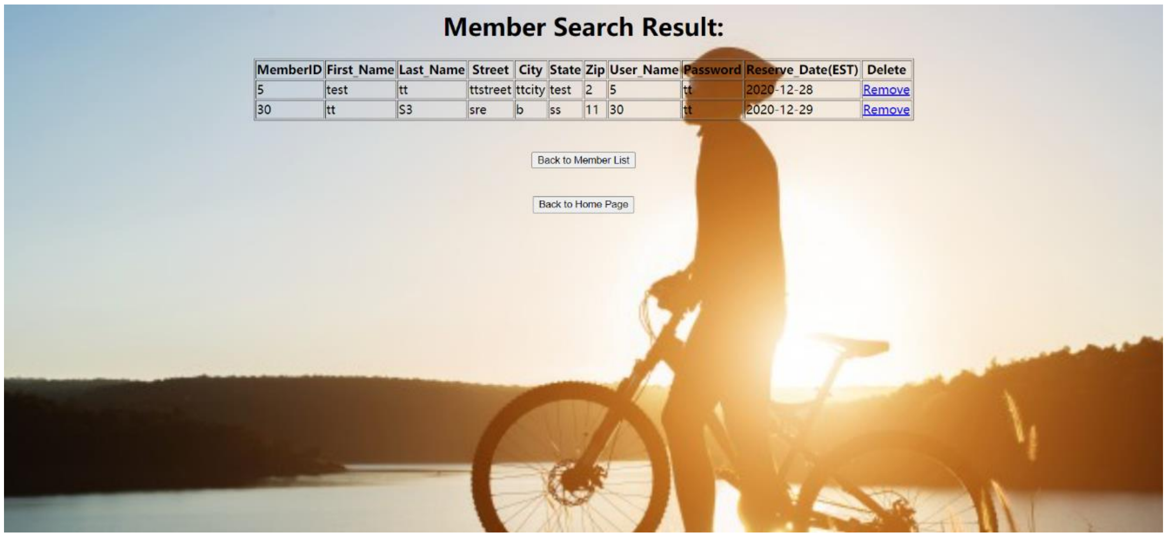
   Member list:



MemberList

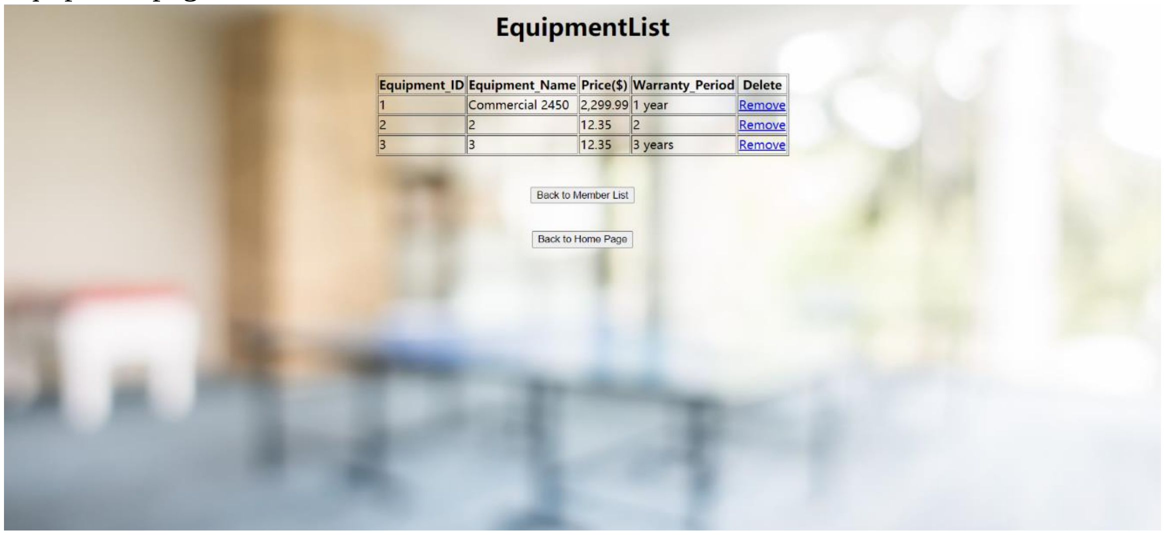
MemberID	First_Name	Last_Name	Street	City	State	Zip	User_Name	Password	Reserve_Date(EST)	Time	Section Ended?	Delete
1	Shuai	Shao	2385 Edison Drive	West Lafayette	IN	47906	1	ss	2020-12-29	14:00	Yes	Remove
3	F3	S3	peter3	Bos	MA	01125	3	ss3	2020-12-29	14:00	Yes	Remove
4	Scott	Wang	1016 Stadium	Bos	LA	12345	4	ss4	2020-12-29	13:00	Yes	Remove
5	test	tt	ttstreet	ttcity	test	2	5	tt	2020-12-28	10:30	Yes	Remove
30	tt	S3	sre	b	ss	11	30	tt	2020-12-29	13:00	Yes	Remove
31	ss	2					31	ss			Yes	Remove
33							33	22			Yes	Remove
35	2	2	3	4	5	5	ss	ss			Yes	Remove
69	Shuai	Shao	28 CLEARWAY ST.	Boston	Massachusetts	02115-3313	ss7777	1234abcd			Yes	Remove
73	Shuai	Shao	28 CLEARWAY ST.	Boston	Massachusetts	02115-3313	ss77777	1234abcd	2020-12-17	16:00	Yes	Remove

   Member search page:


Member Search Result:

MemberID	First_Name	Last_Name	Street	City	State	Zip	User_Name	Password	Reserve_Date(EST)	Delete
5	test	tt	ttstreet	ttcity	test	2	5	tt	2020-12-28	Remove
30	tt	S3	sre	b	ss	11	30	tt	2020-12-29	Remove

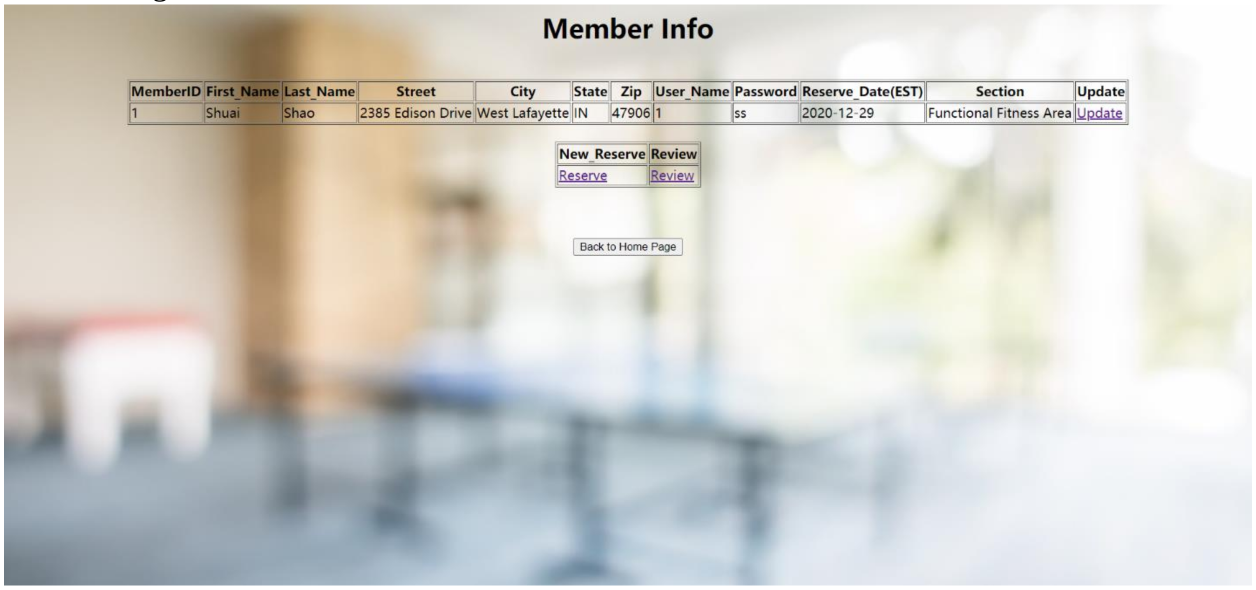
Equipment page:


EquipmentList

Equipment_ID	Equipment_Name	Price($)	Warranty_Period	Delete
1	Commercial 2450	2,299.99	1 year	Remove
2	2	12.35	2	Remove
3	3	12.35	3 years	Remove

6. Member login:
   Member can login according to username and password.
   Member can update information. Null and unsafe values will fail. Filtered by interceptor.
   Member can reserve date, time, section for exercising.
   Member can make reviews and rate.

Member Page:

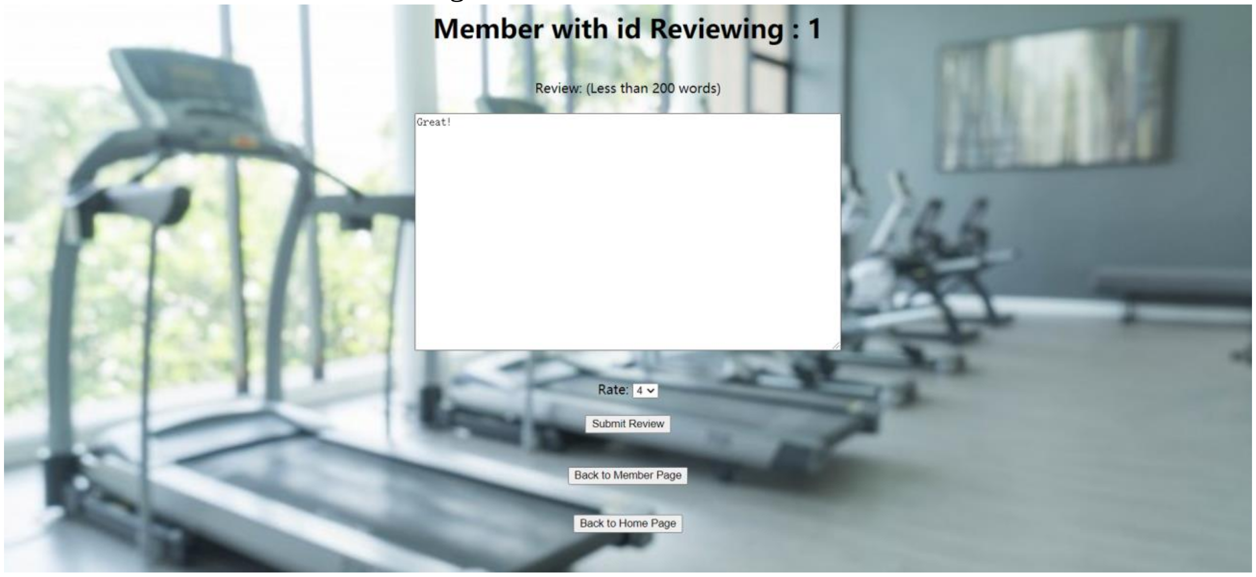


Member Info

MemberID	First_Name	Last_Name	Street	City	State	Zip	User_Name	Password	Reserve_Date(EST)	Section	Update
1	Shuai	Shao	2385 Edison Drive	West Lafayette	IN	47906	1	ss	2020-12-29	Functional Fitness Area	Update

New_Reserve	Review
Reserve	Review

Member review page:
null values will fail when reviewing.



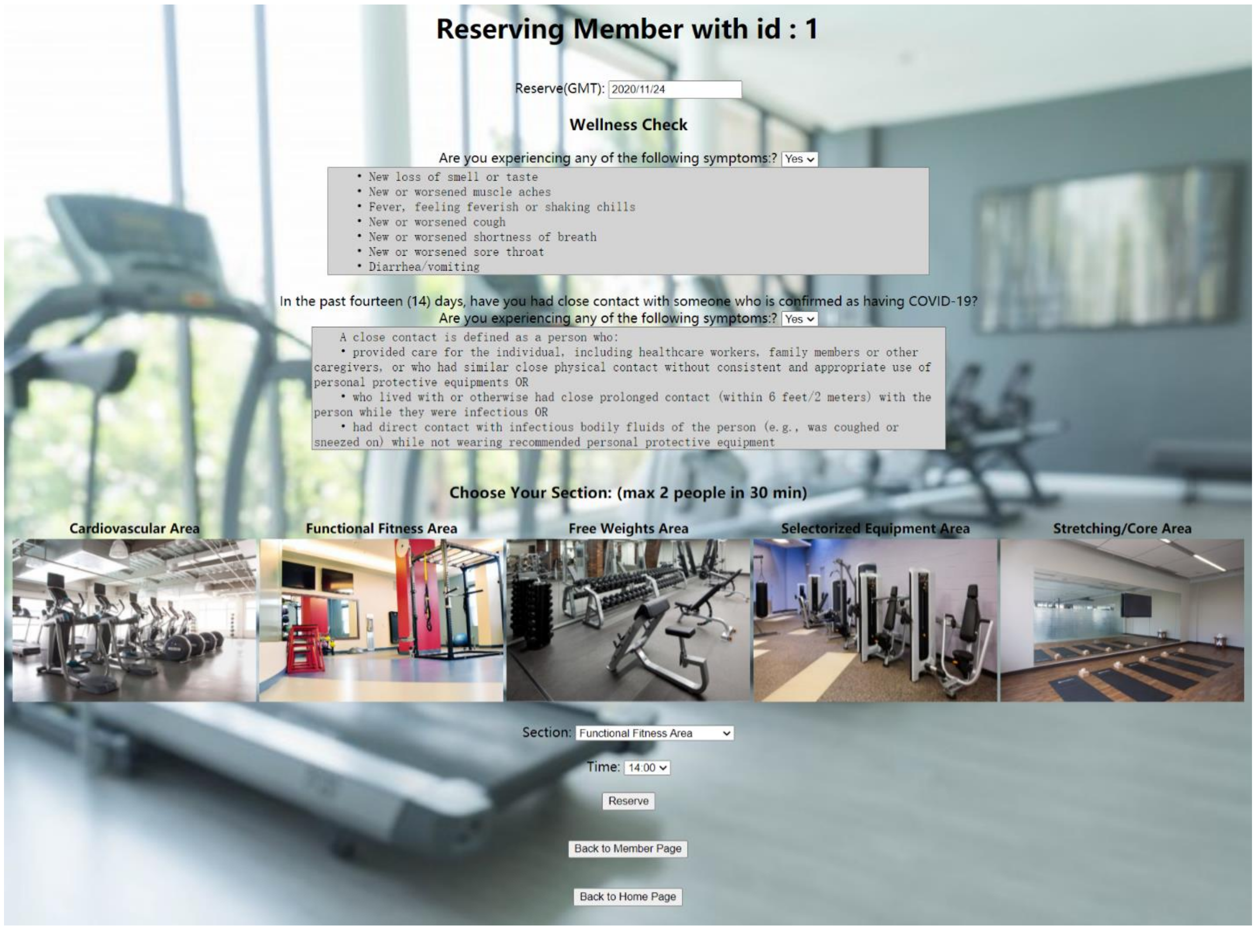
Member with id Reviewing : 1

Review: (Less than 200 words)

Great!

Rate: 4

Member reserve page:


Reserving Member with id : 1

Reserve(GMT): 2020/11/24

Wellness Check

Are you experiencing any of the following symptoms:? Yes

- New loss of smell or taste
- New or worsened muscle aches
- Fever, feeling feverish or shaking chills
- New or worsened cough
- New or worsened shortness of breath
- New or worsened sore throat
- Diarrhea/vomiting

In the past fourteen (14) days, have you had close contact with someone who is confirmed as having COVID-19?
Are you experiencing any of the following symptoms:? Yes

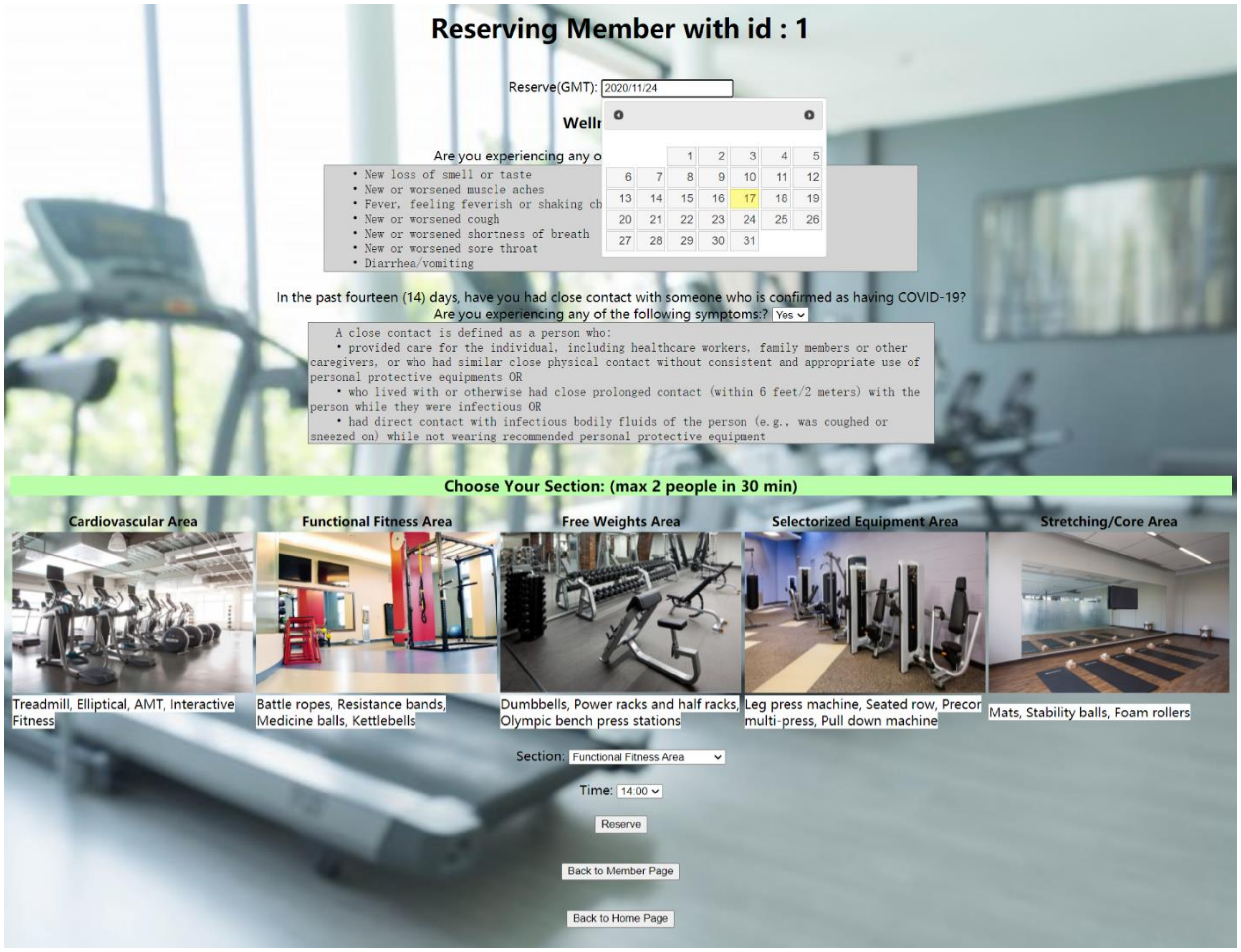
A close contact is defined as a person who:
- provided care for the individual, including healthcare workers, family members or other caregivers, or who had similar close physical contact without consistent and appropriate use of personal protective equipments OR
- who lived with or otherwise had close prolonged contact (within 6 feet/2 meters) with the person while they were infectious OR
- had direct contact with infectious bodily fluids of the person (e.g., was coughed or sneezed on) while not wearing recommended personal protective equipment

Choose Your Section: (max 2 people in 30 min)

Section: Functional Fitness Area

Time: 14:00

Date will appear after clicking date label.
Wellness check must be checked as “No”, otherwise reservation will not be permitted.
Section details will appear after clicking on section pictures.
Section background will be highlighted when selecting section/time.



Reserving Member with id : 1

Reserve(GMT): 2020/11/24

Welln

Are you experiencing any o

- New loss of smell or taste
- New or worsened muscle aches
- Fever, feeling feverish or shaking ch
- New or worsened cough
- New or worsened shortness of breath
- New or worsened sore throat
- Diarrhea/vomiting

In the past fourteen (14) days, have you had close contact with someone who is confirmed as having COVID-19?
Are you experiencing any of the following symptoms:? Yes

A close contact is defined as a person who:
• provided care for the individual, including healthcare workers, family members or other caregivers, or who had similar close physical contact without consistent and appropriate use of personal protective equipments OR
• who lived with or otherwise had close prolonged contact (within 6 feet/2 meters) with the person while they were infectious OR
• had direct contact with infectious bodily fluids of the person (e.g., was coughed or sneezed on) while not wearing recommended personal protective equipment

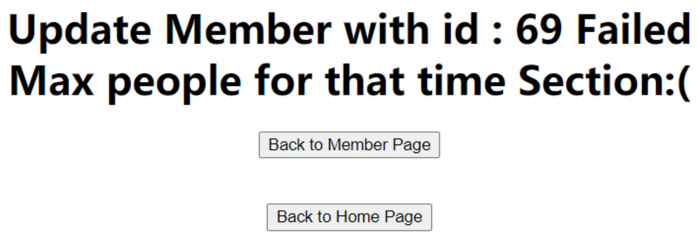
Choose Your Section: (max 2 people in 30 min)

Cardiovascular Area	Functional Fitness Area	Free Weights Area	Selectorized Equipment Area	Stretching/Core Area
Treadmill, Elliptical, AMT, Interactive Fitness	Battle ropes, Resistance bands, Medicine balls, Kettlebells	Dumbbells, Power racks and half racks, Olympic bench press stations	Leg press machine, Seated row, Precor multi-press, Pull down machine	Mats, Stability balls, Foam rollers

Section: Functional Fitness Area

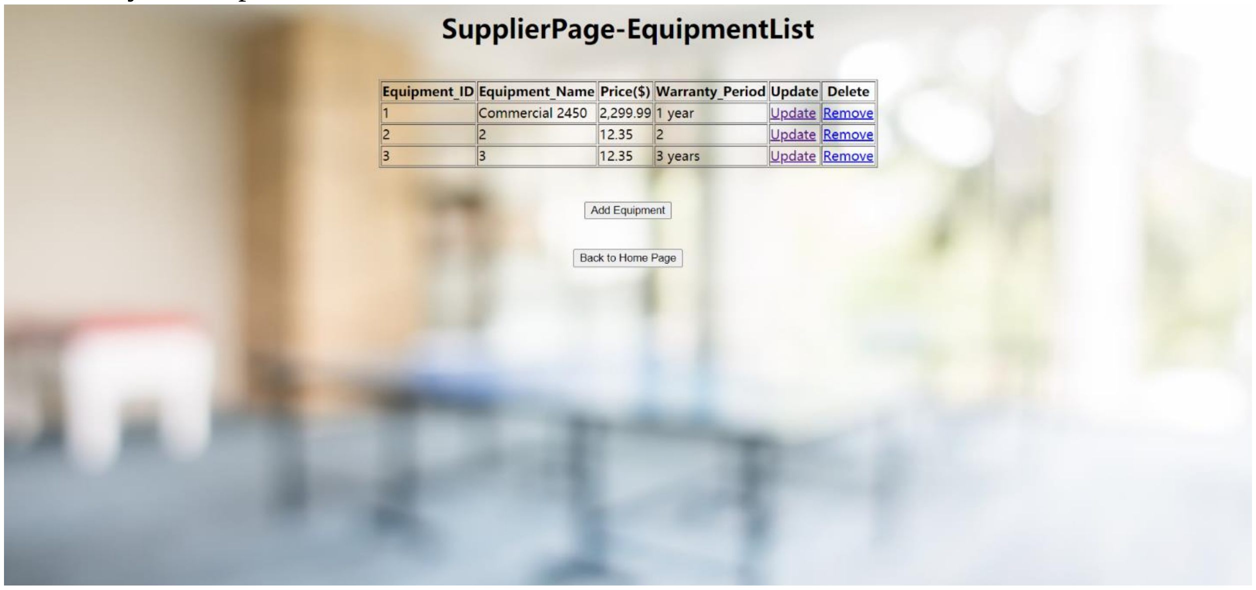
Time: 14:00

Date must be later than today, otherwise reservation fails.
There can only be 2 member reservations in same section at the same date/time. The third reservation will fail.



Update Member with id : 69 Failed
Max people for that time Section:(

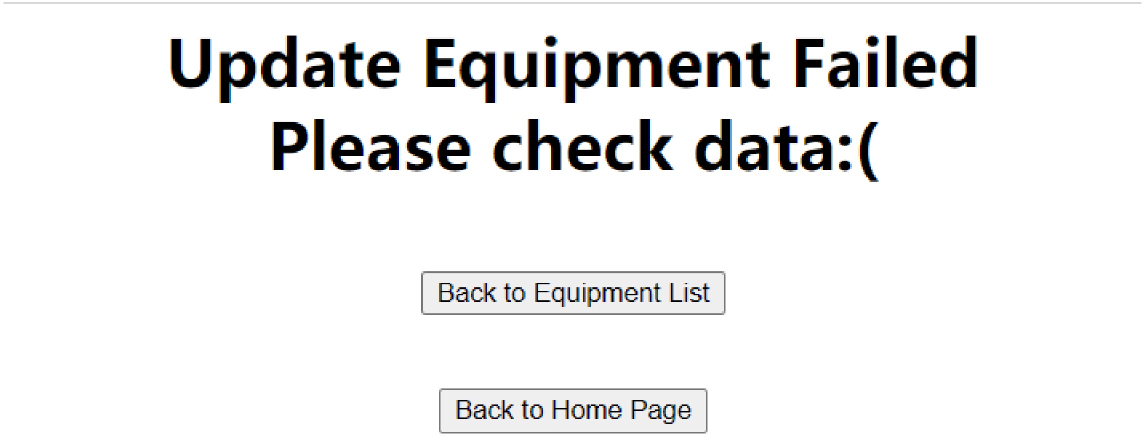
7. Supper login:
   Supplier login according to his/her username and password.
   Supplier can view, add, update, delete equipment. Null and unsafe values will fail. Filtered by interceptor.



SupplierPage-EquipmentList

Equipment_ID	Equipment_Name	Price($)	Warranty_Period	Update	Delete
1	Commercial 2450	2,299.99	1 year	Update	Remove
2	2	12.35	2	Update	Remove
3	3	12.35	3 years	Update	Remove

Price must be integer/double, otherwise update fails.
If price is longer than 3 decimals when updating/adding, it will be rounded and stored to 2 decimals.



Update Equipment Failed
Please check data:(

Controller code:

```
package edu.neu.project.controller;

import edu.neu.project.dao.*;
import edu.neu.project.pojo.*;


import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


//import org.springframework.beans.factory.annotation.Required;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;


@Controller
public class TodoController extends HttpServlet {


//admin login, show supervisor list
@RequestMapping(value = "/admin.htm",method = RequestMethod.GET)
public String getSuperList(HttpServletResponse response, HttpServletRequest
request){
    if(request.getAttribute("unsafe_request") == "true"){
        return ("Error-Home");
    }

    try {
        String up = request.getParameter("password");
        String user=request.getParameter("userAccount");

        //administrator check
        if(user.equals("Ad")){
            int id = Integer.parseInt(request.getParameter("id"));
            AdminDao adminDao=new AdminDao();
            Admin admin = adminDao.AdCheck(id, up);
            if(admin==null){
                return "Error-Home";
            }
            else {
                SupervisorDao supervisorDao = new SupervisorDao();
                List<Supervisor> supervisorList =
supervisorDao.getSupervisors();
                request.setAttribute("superList", supervisorList);
```

```java
                return "Page-SupervisorList";
            }

            //supervisor check
        }else if(user.equals("Sup")){
            int id = Integer.parseInt(request.getParameter("id"));
            SupervisorDao supervisorDao = new SupervisorDao();
            Supervisor supervisor = supervisorDao.Adcheck(id, up);
            if(supervisor ==null){
                return "Error-Home";
            }else {
                MemberDao memberDao = new MemberDao();
                List<Member> memberList = supervisorDao.MemList();
                request.setAttribute("memberList", memberList);
                return "Page-MemberList";
            }
        }

        //Member check
        else if(user.equals("Member")){
            String uname =request.getParameter("id");
            MemberDao memberDao=new MemberDao();
            Member member=memberDao.Memcheck(uname,up);
            if(member==null){
                return "Error-Home";
            }else {

                List<Member> memberList = memberDao.Memlogin(uname);
                request.setAttribute("member", memberList);
                return "Page-Member";
            }
        }

        //supplier
        else if(user.equals("Supplier")){
            int id = Integer.parseInt(request.getParameter("id"));
            SupplierDao supplierDao = new SupplierDao();
            Supplier supplier = supplierDao.Supcheck(id, up);
            if(supplier ==null){
                return "Error-Home";
            }else {
//                List<Supplier> supplierList=supplierDao.Supplierlogin(id);
//                request.setAttribute("supplier",supplierList);
            EquipmentDao equipmentDao = new EquipmentDao();
            List<Equipment> equipmentList = equipmentDao.getEquipments();
                request.setAttribute("equipList", equipmentList);
                return "Page-Supplier";
            }
        }
        //else error
        else {
            return "Error-Home";

        }
    }catch (Exception e) {
        return "Error-Home";
    }
```

```java
}


//add supervisor
    @PostMapping("/addSup.htm")
    public ModelAndView addAdmin(@ModelAttribute("newSup") Supervisor
supervisor, HttpServletResponse response, HttpServletRequest request)throws
ServletException, IOException {

        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println(1);
            return new ModelAndView("Error-Admin");
        }

    if(request.getParameter("password").equals("")){
        return new ModelAndView("Error-Admin");
    }
        ModelAndView mv;

        SupervisorDao supervisorDao = new SupervisorDao();
        int result = supervisorDao.addSuper(supervisor);

        if(result == 1) {
           // mv = new ModelAndView("redirect:/supervisor.htm");
            mv =new ModelAndView("Success-Admin");
            System.out.println("New Supervisor Id: " + supervisor.getId());
        } else mv =  new ModelAndView("Error-Admin");

        return mv;

    }

    //delete supervisor
    @GetMapping("/removeSuper.htm")
    public ModelAndView removeAdmin(@RequestParam("id") Integer supId) {
        ModelAndView mv;

        SupervisorDao supervisorDao = new SupervisorDao();
        int result = supervisorDao.deleteSuper(supId);

        if(result == 1) {
            //mv = new ModelAndView("redirect:/admin.htm");
            mv = new ModelAndView("Success-Admin");
            System.out.println("Deleted Supervisor Id: " + supId);
        } else mv =  new ModelAndView("Error-Admin");

        return mv;
    }

    //update click
    @GetMapping("/updateSuper.htm")
    public ModelAndView updateAdminGet(@RequestParam("id") Integer supId) {
        ModelAndView mv;

        SupervisorDao supervisorDao = new SupervisorDao();
        Supervisor supervisor = supervisorDao.getSuper(supId);
        if(supervisor != null) {
            mv = new ModelAndView("Update-Supervisor", "existingSupervisor",
supervisor);
        } else mv =  new ModelAndView("Error-Admin");

        return mv;
    }

    //update supervisor
    @PostMapping("/updateSuper2.htm")
    public ModelAndView updateAdminPost(@ModelAttribute("existingSupervisor")
Supervisor supervisor, HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Admin");
        }

        ModelAndView mv;
        if(supervisor.getPassword().equals("")){
         return new ModelAndView("Error-Admin");
        }
        SupervisorDao supervisorDao =new SupervisorDao();
        int result = supervisorDao.updateSuper(supervisor.getId(),
supervisor.getfName(), supervisor.getlName(), supervisor.getStreet(),
supervisor.getCity(), supervisor.getState(), supervisor.getZip(),
supervisor.getPassword());

        if(result == 1) {
            //mv = new ModelAndView("redirect:/supervisor.htm");
            mv = new ModelAndView("Success-Admin");
            System.out.println("Updated Supervisor Id: " +
supervisor.getId());
        } else mv =  new ModelAndView("Error-Admin");

        return mv;
    }

//switch to add sup page
@PostMapping("/addSup2.htm")
    public ModelAndView add(@ModelAttribute("newSup") Supervisor supervisor){
        return new ModelAndView("AddSupervisor");
    }

//return to supervisor list
    @PostMapping("/returnSup.htm")
    public ModelAndView returnAdmin(ModelMap model){
        Supervisor supervisor = new Supervisor();
        model.addAttribute("newSuper", supervisor);

        SupervisorDao supervisorDao = new SupervisorDao();
        List<Supervisor> supervisorList = supervisorDao.getSupervisors();
        return new ModelAndView("Page-SupervisorList", "superList",
supervisorList);
    }
```

```java
//Register for member
    @PostMapping("/addMem2.htm")
    public ModelAndView addMem2(@ModelAttribute("newMem") Member
member,HttpServletResponse response,HttpServletRequest request)throws
ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Register");
        }

        //uName start with alphabet, must have numbers, >5, <=15
        if(!request.getParameter("uName").matches("^[A-z][A-z0-9]{5,15}$")){
            System.out.println(1);
            return new ModelAndView("Error-Register");
        }

        //password, must have alphabet and numbers, >7, <=20
        if(!request.getParameter("password").matches("^[A-z0-9][A-z0-
9]{7,20}$")){
            System.out.println(2);
            return new ModelAndView("Error-Register");
        }
        ModelAndView mv;

        MemberDao memberDao = new MemberDao();
        int result = memberDao.addMember(member);

        if(result == 1) {
            // mv = new ModelAndView("redirect:/admin.htm");
            mv =new ModelAndView("Success-Register");
            System.out.println("New Member Id: " + member.getId());
        } else mv =  new ModelAndView("Error-Register");

        return mv;

    }


//Supervisor remove member
    @GetMapping("/removeMember.htm")
    public ModelAndView removeMember(@RequestParam("id") Integer memberId) {
        ModelAndView mv;

        MemberDao memberDao = new MemberDao();
        int result = memberDao.deleteMember(memberId);

        if(result == 1) {
            //mv = new ModelAndView("redirect:/admin.htm");
            mv = new ModelAndView("Success-Supervisor");
            System.out.println("Deleted Member Id: " + memberId);
        } else mv =  new ModelAndView("Error-Sup");

        return mv;
    }
    //supervisor view Review
    @RequestMapping(value = "/deleteRev.htm",method = RequestMethod.POST)
    public String ViewRev(HttpServletResponse response, HttpServletRequest
request) {
        MemberDao memberDao = new MemberDao();
        List<Member> memReview = memberDao.MemReview();
        double memAvg= memberDao.MemRate();
        System.out.println(memAvg);
        request.setAttribute("revAvg", memAvg);
        request.setAttribute("reviewList", memReview);
        return "Update-deleteRev";
    }


    //Supervisor remove review
    @GetMapping("/removeRev.htm")
    public ModelAndView removeRev(@RequestParam("id") Integer memberId) {
        ModelAndView mv;

        SupervisorDao supervisorDao = new SupervisorDao();
        int result = supervisorDao.deleteRev(memberId);

        if(result == 1) {
            //mv = new ModelAndView("redirect:/admin.htm");
            mv = new ModelAndView("Success-Supervisor");
            System.out.println("Deleted Review of: " + memberId);
        } else mv =  new ModelAndView("Error-Sup");

        return mv;
    }


    //Supervisor remove reserve
    @GetMapping("/removeReserve.htm")
    public ModelAndView removeReserve(@RequestParam("id") Integer memberId) {
        ModelAndView mv;

        SupervisorDao supervisorDao = new SupervisorDao();
        int result = supervisorDao.deleteReserve(memberId);

        if(result == 1) {
            //mv = new ModelAndView("redirect:/admin.htm");
            mv = new ModelAndView("Success-Supervisor");
            System.out.println("Deleted Reserve of: " + memberId);
        } else mv =  new ModelAndView("Error-Sup");

        return mv;
    }





    //Member update switch page
    @GetMapping("/updateMember.htm")
    public ModelAndView updateMemberGet(@RequestParam("id") Integer memberId)
{
        ModelAndView mv;

        MemberDao memberDao = new MemberDao();
        Member member = memberDao.getMember(memberId);
        if(member != null) {
            mv = new ModelAndView("Update-Member", "existingmember", member);
        } else mv =  new ModelAndView("Error-Member");

        return mv;
    }


    //update member by Member
    @PostMapping("/updateMember2.htm")
    public ModelAndView updateMemberPost(@ModelAttribute("existingMember")
Member member,HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Member","existingmember", member);
        }

        //uName start with alphabet, must have numbers, >5, <=15
        if(!request.getParameter("uName").matches("^[A-z][A-z0-9]{5,15}$")){
            System.out.println(1);
            return new ModelAndView("Error-Member","existingmember", member);
        }

        //password, must have alphabet and numbers, >7, <=20
        if(!request.getParameter("password").matches("^[A-z0-9][A-z0-
9]{7,20}$")){
            System.out.println(2);
            return new ModelAndView("Error-Member","existingmember", member);
        }

        ModelAndView mv;
        try {
            if (member.getPassword().equals("")) {
                return new ModelAndView("Error-Member", "existingmember",
member);
            }
            MemberDao memberDao = new MemberDao();
            int result = memberDao.updateMember(member.getId(),
member.getfName(), member.getlName(), member.getStreet(), member.getCity(),
member.getState(), member.getZip(), member.getuName(), member.getPassword());

            if (result == 1) {
                //mv = new ModelAndView("redirect:/admin.htm");
                mv = new ModelAndView("Success-Member", "existingmember",
member);
                System.out.println("Updated Member Id: " + member.getId());
            } else mv = new ModelAndView("Error-Member", "existingmember",
member);
        }catch (Exception e){
            System.out.println(1);
            mv = new ModelAndView("Error-Member", "existingmember", member);
        }
        return mv;
    }

    //switch to register member
    @PostMapping("/addMem.htm")
    public ModelAndView addMem(@ModelAttribute("newMem")Member member){
        return new ModelAndView("AddMember");
    }


    //return to Member List
    @PostMapping("/returnAd_Mem.htm")
    public ModelAndView returnAd_Mem(ModelMap model){
        Member member = new Member();
        model.addAttribute("newMem", member);

        MemberDao memberDao = new MemberDao();

        List<Member> members = memberDao.getMembers();
        return new ModelAndView("Page-MemberList", "memberList", members);
    }



    //return to Member
    @PostMapping("/returnMem.htm")
    public ModelAndView returnMem(@RequestParam("id") Integer
memberId,ModelMap model){
        Member member = new Member(); // SimpleFormController
        model.addAttribute("newMem", member);

        MemberDao memberDao = new MemberDao();
        List<Member> members = memberDao.Memlogin2(memberId);
        return new ModelAndView("Page-Member", "member", members);
    }



    //return to Home
@PostMapping("/returnHome.htm")
public ModelAndView returnHome(ModelMap model){
return new ModelAndView("Home");
}


//search for Member
    @RequestMapping(value = "/searchMem.htm",method = RequestMethod.GET)
    public String MemSearch(HttpServletResponse response, HttpServletRequest
request){
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return ("Error-Sup");
        }
    try {
            //int id = Integer.parseInt(request.getParameter("MemId"));
            String name=request.getParameter("MemId");
                MemberDao memberDao=new MemberDao();
                List<Member> memList = memberDao.MemSearch(name);
                request.setAttribute("memList", memList);
                return "Search-Member";

        }catch (Exception e) {
            return "Error-Sup";
        }
    }


    //member reserve page jump
    @GetMapping("/memberDate.htm")
    public ModelAndView updateMemberDate(@RequestParam("id") Integer
memberId) {
        ModelAndView mv;

        MemberDao memberDao = new MemberDao();
        Member member = memberDao.getMember(memberId);
        if(member != null) {
            mv = new ModelAndView("Member-Reserve", "existingmember",
member);
        } else mv =  new ModelAndView("Error-Member");

        return mv;
    }


    //member reserve
    @PostMapping("/updateMemberDate.htm")
    public ModelAndView updateMemberDate(@ModelAttribute("existingMember")
Member member,HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
//          if(request.getAttribute("unsafe_request") == "true"){
//              System.out.println("Interceptor");
//              return new ModelAndView("Error-Member","existingmember",
member);
//          }
        MemberDao memberDao=new MemberDao();
        long
check=memberDao.checkRev(member.getSection(),member.getDate(),member.getTime(
));
        System.out.println(check);
        ModelAndView mv;

        Date date1=new Date();
        System.out.println(date1);
        System.out.println(member.getDate());


            try {
            //check date
            if(date1.compareTo(member.getDate())>0) {
            System.out.println("date Error");
            return new ModelAndView("Error-Member", "existingmember",
member);
            }

                //check same date, time, each section cannot contain>2 people
                if(check>1){
                    System.out.println("check");
                    System.out.println(check);
                    return  new ModelAndView("Error-
Reserve","existingmember", member);
                }

                //check wellness
                String check1=request.getParameter("check1");
                String check2=request.getParameter("check2");
//            MemberDao memberDao=new MemberDao();
                System.out.println(member.getDate());

            if(check1.equals("check1No")&&check2.equals("check2No")) {
                int result =
memberDao.updateMemberDate(member.getId(),member.getfName(),member.getlName()
,member.getStreet(),member.getCity(),member.getState(),member.getZip(),member
.getPassword(),member.getDate(),member.getSection(),member.getTime());
                mv = new ModelAndView("Success-Member","existingmember",
member);
                System.out.println("Updated Member Id: " + member.getId());
            } else mv =  new ModelAndView("Error-Member","existingmember",
member);
        }catch (Exception e){
            return  new ModelAndView("Error-Member","existingmember",
member);
        }
        return mv;
    }



    //member review page jump
    @GetMapping("/memberReview.htm")
    public ModelAndView updateMemberReview(@RequestParam("id") Integer
memberId) {
        ModelAndView mv;

        MemberDao memberDao = new MemberDao();
        Member member = memberDao.getMember(memberId);
        if(member != null) {
            mv = new ModelAndView("Member-Review", "existingmember", member);
        } else mv =  new ModelAndView("Error-Member");

        return mv;
    }

    //member review
    @PostMapping("/updateMemberReview.htm")
    public ModelAndView updateMemberReview(@ModelAttribute("existingMember")
Member member,HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Member","existingmember", member);
        }

    ModelAndView mv;

        try {
            MemberDao memberDao=new MemberDao();
            int result =
memberDao.updateMemberReview(member.getId(),member.getfName(),member.getlName
(),member.getStreet(),member.getCity(),member.getState(),member.getZip(),memb
er.getPassword(),member.getReview(),member.getRate());

            if(result == 1) {
                //mv = new ModelAndView("redirect:/admin.htm");
                mv = new ModelAndView("Success-Member","existingmember",
member);
                System.out.println("Updated Member Id: " + member.getId());
            } else mv =  new ModelAndView("Error-Member","existingmember",
member);
        }catch (Exception e){
            return new ModelAndView("Error-Member","existingmember", member);
        }
        return mv;
    }


    //view Review and rate
    @RequestMapping(value = "/memReview.htm",method = RequestMethod.POST)
    public String getReviewList(HttpServletResponse response,
HttpServletRequest request) {
        MemberDao memberDao = new MemberDao();
        List<Member> memReview = memberDao.MemReview();
        double memAvg= memberDao.MemRate();
        System.out.println(memAvg);
        request.setAttribute("revAvg", memAvg);
        request.setAttribute("reviewList", memReview);
        return "Page-ReviewAndRate";
    }





    //update supplier switch page
    @GetMapping("/updateSupplier.htm")
    public ModelAndView updateSupplierGet(@RequestParam("id") Integer
supplierId) {
        ModelAndView mv;

        SupplierDao supplierDao = new SupplierDao();
        Supplier supplier = supplierDao.getSupplier(supplierId);

        if(supplier != null) {
            mv = new ModelAndView("Update-Supplier", "existingSupplier",
supplier);
        } else mv =  new ModelAndView("Error-Supplier");
        return mv;
    }

    //update supplier
    @PostMapping("/updateSupplier2.htm")
    public ModelAndView
updateSupplierPost(@ModelAttribute("existingSupplier") Supplier supplier,
HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Supplier");
        }

        ModelAndView mv;
        if(supplier.getPassword().equals("")){
            return new ModelAndView("Error-Supplier");
        }
        SupplierDao supplierDao =new SupplierDao();
        int result = supplierDao.updateSupplier(supplier.getId(),
supplier.getCompany(), supplier.getContact(), supplier.getPassword());

        if(result == 1) {
            //mv = new ModelAndView("redirect:/supervisor.htm");
            mv = new ModelAndView("Success-Supplier");
            System.out.println("Updated Supplier Id: " + supplier.getId());
        } else mv =  new ModelAndView("Error-Supplier");

        return mv;
    }

    //remove supplier
    @GetMapping("/removeSupplier.htm")
    public ModelAndView removeSupplier(@RequestParam("id") Integer
supplierId) {
        ModelAndView mv;

        SupplierDao supplierDao = new SupplierDao();
        int result = supplierDao.deleteSupplier(supplierId);

        if(result == 1) {
            //mv = new ModelAndView("redirect:/admin.htm");
            mv = new ModelAndView("Success-Supplier");
            System.out.println("Deleted Supplier Id: " + supplierId);
        } else mv =  new ModelAndView("Error-Supplier");

        return mv;
    }

    //return supplier
    @PostMapping("/returnSupplier.htm")
    public ModelAndView returnSupplier(@RequestParam("id") Integer
supplierId,ModelMap model){
        Member member = new Member(); // SimpleFormController
        model.addAttribute("newSupplier", supplierId);
        SupplierDao supplierDao = new SupplierDao();
        List<Supplier> suppliers = supplierDao.Supplierlogin(supplierId);
        return new ModelAndView("Page-Supplier", "supplier", suppliers);
    }

    //view all suppliers
    @PostMapping("/viewSupplier.htm")
    public String viewSupplier(ModelMap model,HttpServletRequest
request,HttpServletResponse response) {
        SupplierDao supplierDao = new SupplierDao();
        List<Supplier> supplierList = supplierDao.getSuppliers();
        request.setAttribute("supplierList", supplierList);
        return("Page-SupplierList");
    }

    //return to supplier list
    @PostMapping("/returnSupplierList.htm")
    public ModelAndView returnSupplierList(ModelMap model){
        Supplier supplier = new Supplier(); // SimpleFormController
        model.addAttribute("newSupplier", supplier);

        SupplierDao supplierDao = new SupplierDao();
        List<Supplier> supplierList = supplierDao.getSuppliers();
        return new ModelAndView("Page-SupplierList", "supplierList",
supplierList);
    }

    //add supplier switch page
    @PostMapping("/addSupplier.htm")
    public ModelAndView addSupplier(@ModelAttribute("newSupplier") Supplier
supplier){
        return new ModelAndView("AddSupplier");
    }


    //add supplier
    @PostMapping("/addSupplier2.htm")
    public ModelAndView addSupplier(@ModelAttribute("newSupplier") Supplier
supplier, HttpServletResponse response, HttpServletRequest request)throws
ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Supplier");
        }

    if(request.getParameter("password").equals("")){
            return new ModelAndView("Error-Supplier");
        }
        ModelAndView mv;

        SupplierDao supplierDao = new SupplierDao();
        int result = supplierDao.addSupplier(supplier);

        if(result == 1) {
            // mv = new ModelAndView("redirect:/supervisor.htm");
            mv =new ModelAndView("Success-Supplier");
            System.out.println("New Supplier Id: " + supplier.getId());
        } else mv =  new ModelAndView("Error-Supplier");

        return mv;
    }



    //update equipment switch page
    @GetMapping("/updateEquipment.htm")
    public ModelAndView updateEquipment(@RequestParam("id") Integer
equipmentId) {
        ModelAndView mv;

        EquipmentDao equipmentDao = new EquipmentDao();
        Equipment equipment = equipmentDao.getEquipment(equipmentId);

        if(equipment != null) {
            mv = new ModelAndView("Update-Equipment", "existingEquipment",
equipment);
        } else mv =  new ModelAndView("Error-Equipment");

        return mv;
    }

    //update equipment
    @PostMapping("/updateEquipment2.htm")
    public ModelAndView updateEquipPost(@ModelAttribute("existingEquipment")
Equipment equipment, HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Equipment");
        }

        ModelAndView mv;
        try {
//          if(equipment.getPassword().equals("")){
//              return new ModelAndView("Error-Supplier");
//          }
            int result=0;
            double price=Double.parseDouble(equipment.getPrice());
//              if(price%1!=0) {
                EquipmentDao equipmentDao = new EquipmentDao();
                result = equipmentDao.updateEquipment(equipment.getId(),
equipment.geteName(), String.format("%.2f", price), equipment.getWarranty());

                if (result == 1) {
                    //mv = new ModelAndView("redirect:/supervisor.htm");
                    mv = new ModelAndView("Success-Equipment");
                    System.out.println("Updated Equipment Id: " +
equipment.getId());
                } else mv = new ModelAndView("Error-Equipment");
//              }else {
//                  mv = new ModelAndView("Error-Equipment");
//              }

        }catch (NumberFormatException e){
```

```java
            mv=  new ModelAndView("Error-Equipment");
        }
        return mv;
    }

    //remove equipment
    @GetMapping("/removeEquipment.htm")
    public ModelAndView removeEquip(@RequestParam("id") Integer equipmentId)
{
        ModelAndView mv;

        EquipmentDao equipmentDao = new EquipmentDao();
        int result = equipmentDao.deleteEquipment(equipmentId);

        if(result == 1) {
            //mv = new ModelAndView("redirect:/admin.htm");
            mv = new ModelAndView("Success-Equipment");
            System.out.println("Deleted Equipment Id: " + equipmentId);
        } else mv =  new ModelAndView("Error-Equipment");

        return mv;
    }

    //return to equip list
    @PostMapping("/returnEquipList.htm")
    public ModelAndView returnEquipList(ModelMap model){
        Equipment equipment = new Equipment(); // SimpleFormController
        model.addAttribute("newEquip", equipment);

        EquipmentDao equipmentDao = new EquipmentDao();
        List<Equipment> equipmentList = equipmentDao.getEquipments();
        return new ModelAndView("Page-Supplier", "equipList", equipmentList);
    }


    //switch to add equip page
    @PostMapping("/addEquip.htm")
    public ModelAndView addEquip(@ModelAttribute("newEquip") Equipment
equipment){
        return new ModelAndView("AddEquipment");
    }

    //add equipment
    @PostMapping("/addEquip2.htm")
    public ModelAndView addEquip(@ModelAttribute("newEquip") Equipment
equipment, HttpServletResponse response, HttpServletRequest request)throws
ServletException, IOException {
        if(request.getAttribute("unsafe_request") == "true"){
            System.out.println("Interceptor");
            return new ModelAndView("Error-Equipment");
        }

        ModelAndView mv;
        try {
            int result=0;
            double price=Double.parseDouble(equipment.getPrice());
//              if(price%1!=0) {
```

```
                EquipmentDao equipmentDao = new EquipmentDao();
                result = equipmentDao.addEquipment(equipment);
                if (result == 1) {
                    // mv = new ModelAndView("redirect:/supervisor.htm");
                    mv = new ModelAndView("Success-Equipment");
                    System.out.println("New Equipment Id: " +
equipment.getId());
                } else mv = new ModelAndView("Error-Equipment");
//              }else {
//                  mv = new ModelAndView("Error-Equipment");
//              }
        }catch (Exception e){
            mv = new ModelAndView("Error-Equipment");
        }

        return mv;
    }

    //supervisor view equip
    @PostMapping("/viewEquip.htm")
    public String viewEquip(ModelMap model,HttpServletRequest
request,HttpServletResponse response) {
        EquipmentDao equipmentDao = new EquipmentDao();
        List<Equipment> equipmentList = equipmentDao.getEquipments();
        request.setAttribute("equipList", equipmentList);
        return "Page-EquipList";
    }


    //supervisor remove equipment
    @GetMapping("/removeEquip.htm")
    public ModelAndView removeEquipment(@RequestParam("id") Integer
equipmentId) {
        ModelAndView mv;

        EquipmentDao equipmentDao = new EquipmentDao();
        int result = equipmentDao.deleteEquipment(equipmentId);

        if(result == 1) {
            //mv = new ModelAndView("redirect:/admin.htm");
            mv = new ModelAndView("Success-Equipment");
            System.out.println("Deleted Equipment Id: " + equipmentId);
        } else mv =  new ModelAndView("Error-Equipment");

        return mv;
    }
}
```