INFO6250 36648 Web Development Tools SEC 04 - Spring 2017

Venue Management Tool (BeOnGame) **Vinoth Punniyamoorthy(001225656)**

Abstract:

Online portal for finding and reserving available venue for specific type of outdoor / Indoor game like Badminton, Tennis, Table Tennis, etc. The application can aid to reserve the venue for the day based on end user. On the available list of Venues provided by the service providers, end user can book and reserve the venues. On blocking a venue, the number of slots available for the day goes down and email confirmation is sent to the end-user based on the reservation.

Roles:

1. **Admin:** Admin registers the service providers.
2. **Service Providers**: Adds or Removes the venues slots per sport.
3. **Users:** A user can search and reserve an available sports services based on the game he wants to play.

Features:

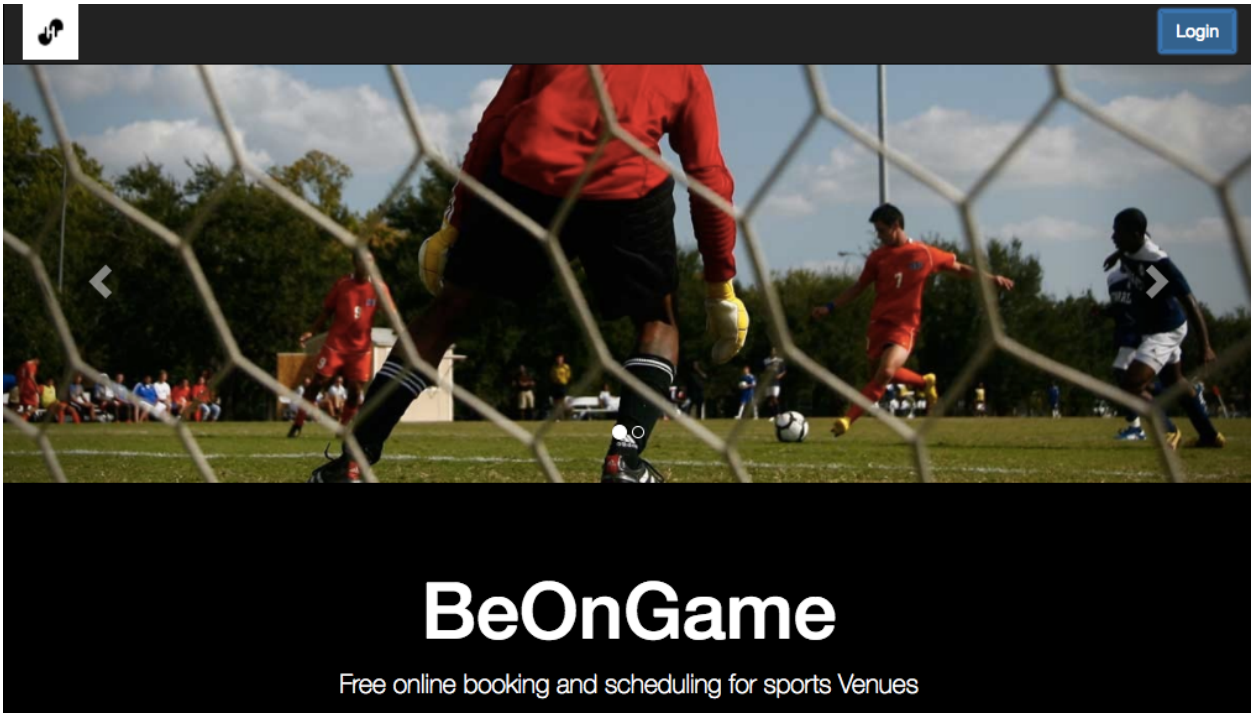
- Manage and show availability of venue to play specific indoor and outdoor sports.
- Provide a frictionless booking process for your users.
- Control your venue from anywhere
- Customize your venue setup.

Key Concepts Included:

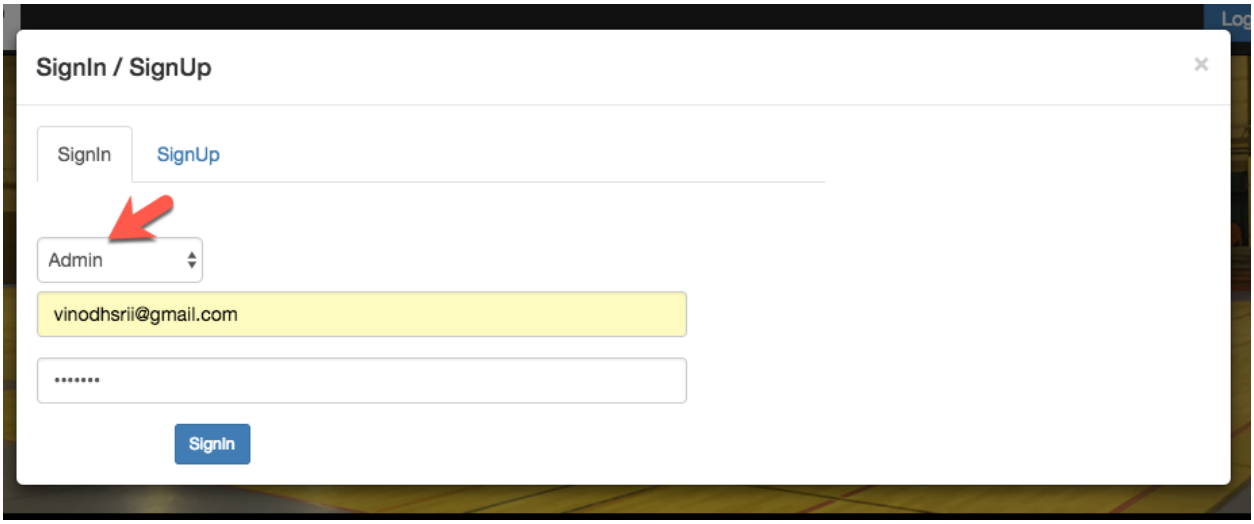
- Annotated POJO Model
- Annotated Controllers
- Hibernate
- JSP View Pages
- Velocity View Page
- Ajax's Concepts including JSON
- Authorization Securities.

Work Flow:

Home Screen:

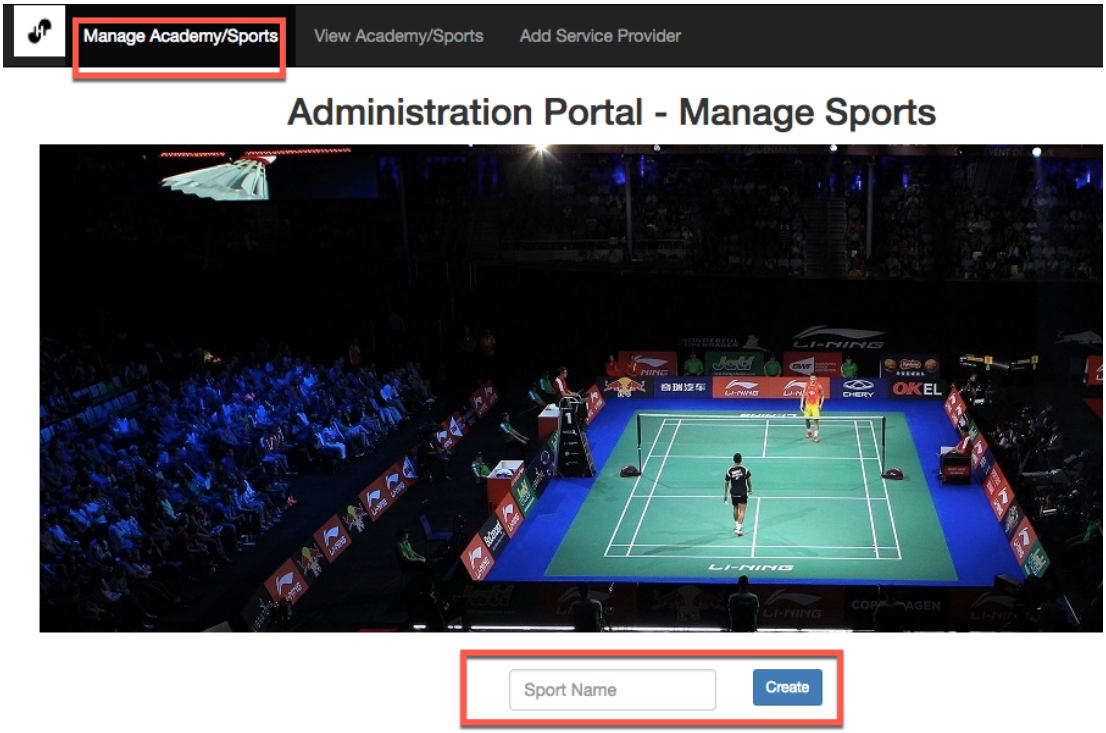


Step 1: Admin Login

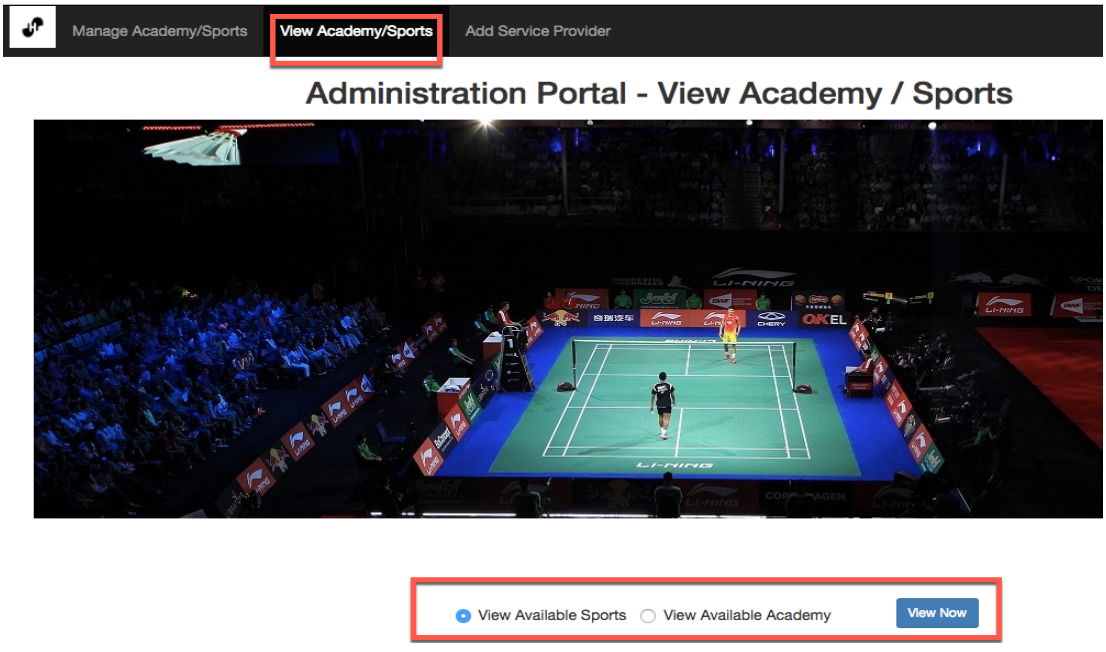


Step 2: Admin has 3 functionalities

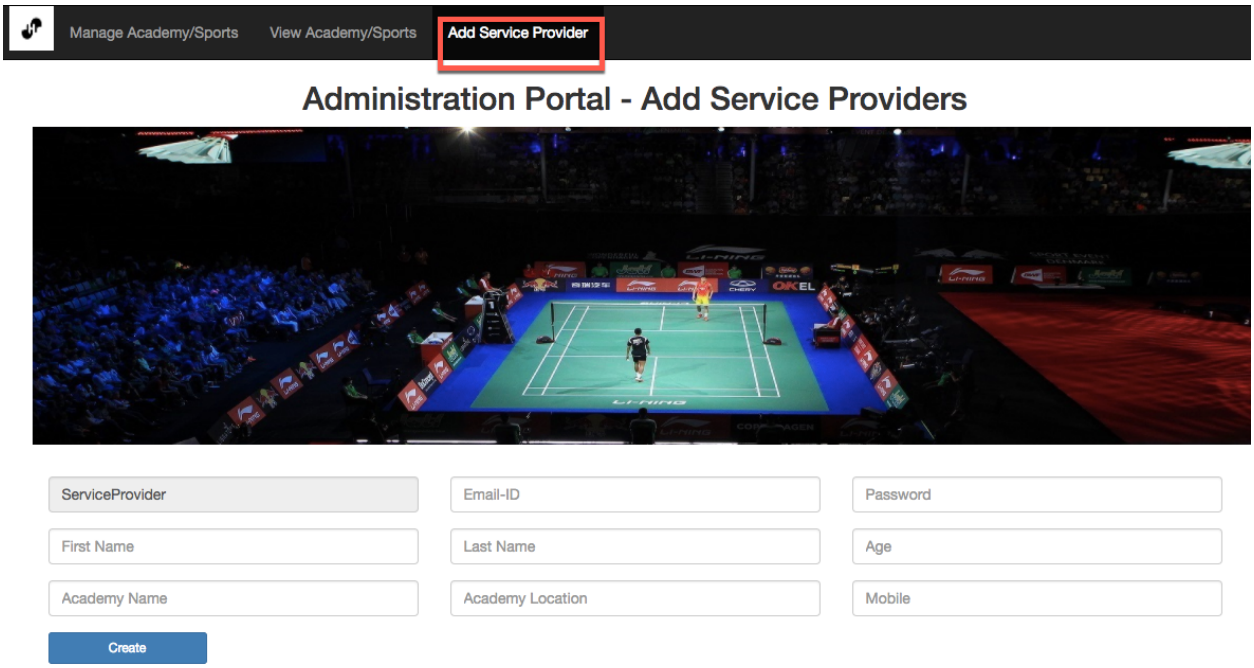
a. **Manage Sports Academy -** Adds Sports that Service Providers can include.



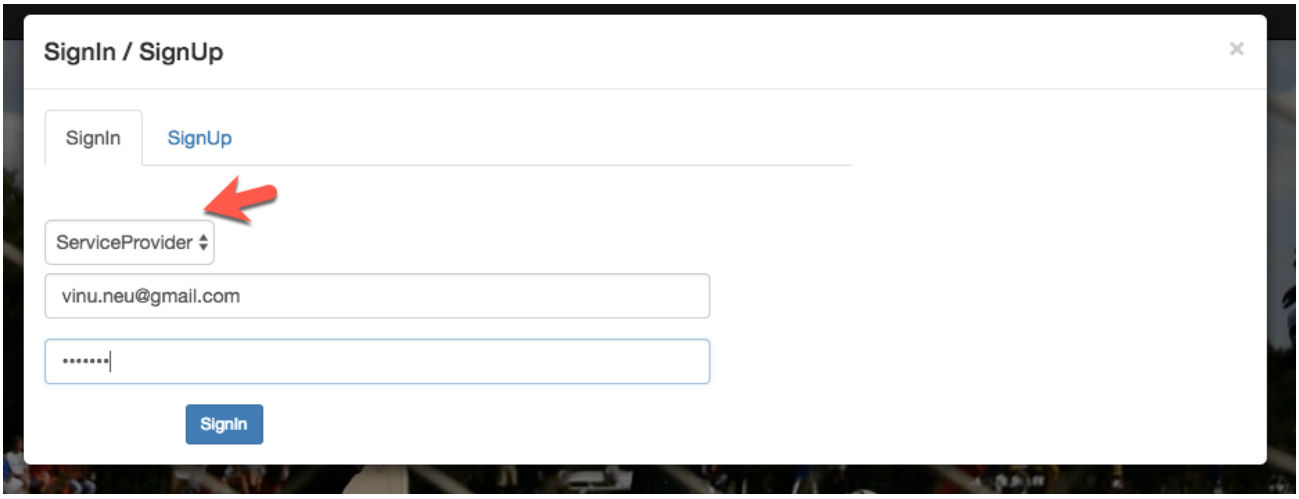
b. **View Academy / Sports –** Can have a look at all Sports and Academies.



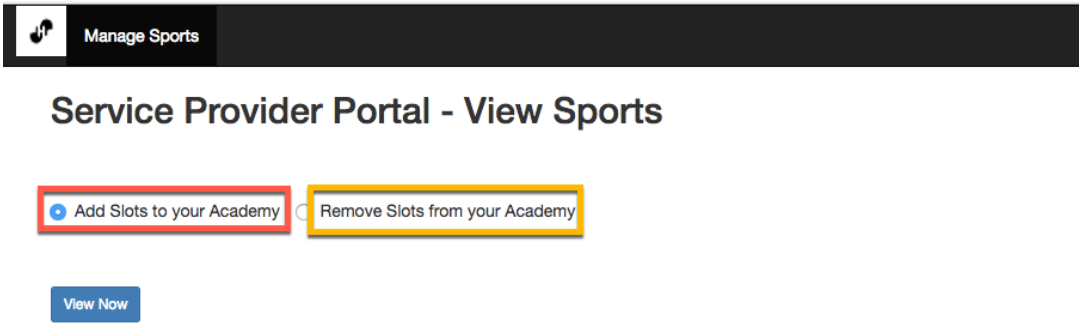
c. **Add Service Provider -** Adds Service Providers to the Portal / get registered.



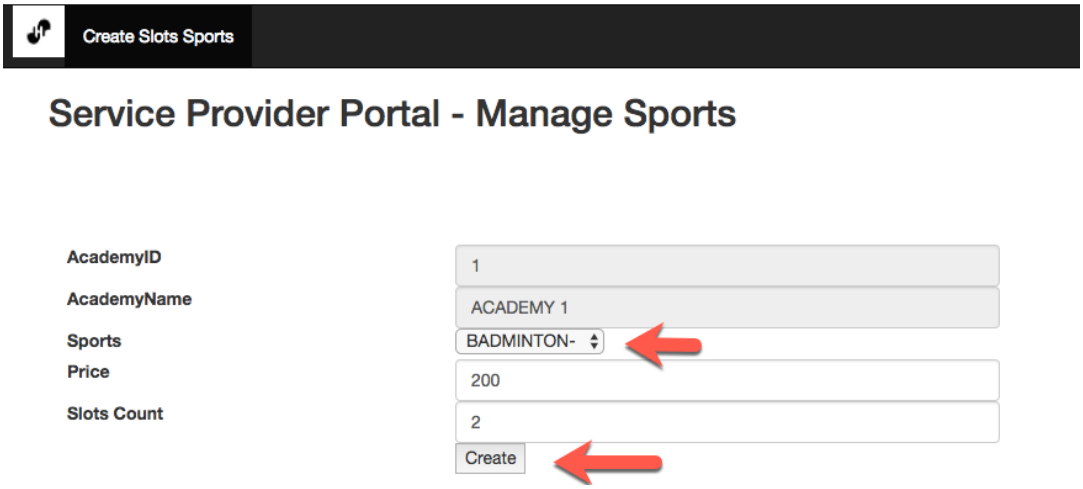
Step 3: Service Provider Login



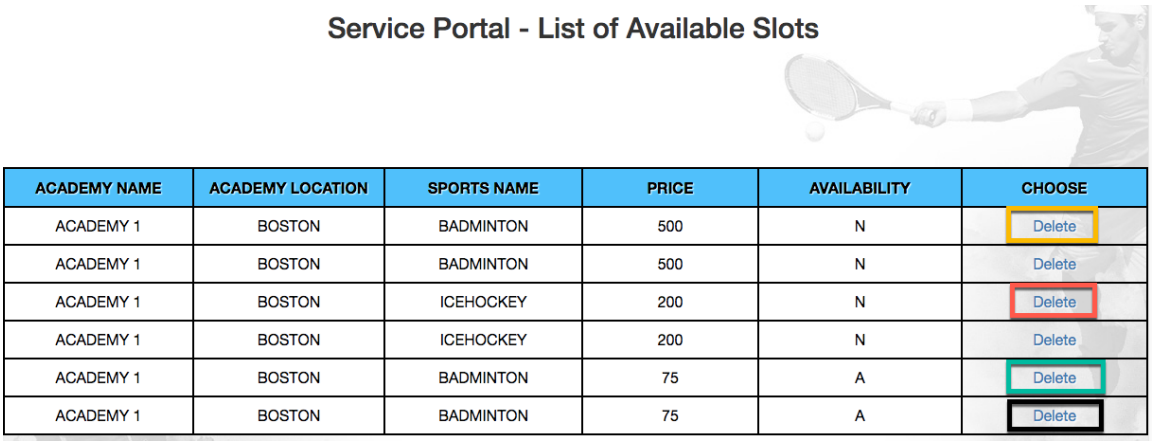
Step 4: Service Provider has 2 Functionalities



a. Add Slots / Sports to his /her Academy

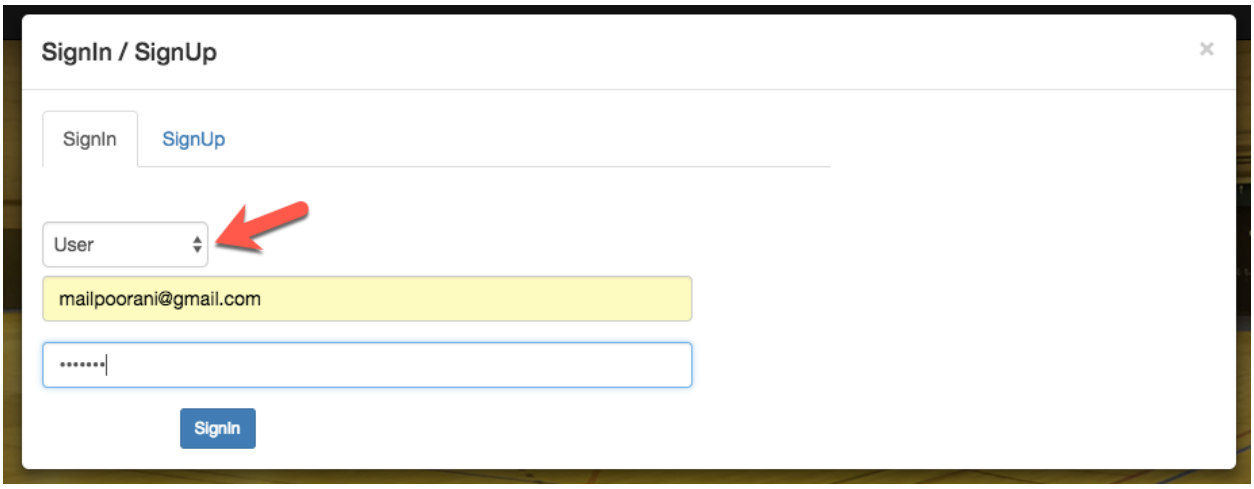


b. Remove Slots /sports to his/her Academy



ACADEMY NAME	ACADEMY LOCATION	SPORTS NAME	PRICE	AVAILABILITY	CHOOSE
ACADEMY 1	BOSTON	BADMINTON	500	N	Delete
ACADEMY 1	BOSTON	BADMINTON	500	N	Delete
ACADEMY 1	BOSTON	ICEHOCKEY	200	N	Delete
ACADEMY 1	BOSTON	ICEHOCKEY	200	N	Delete
ACADEMY 1	BOSTON	BADMINTON	75	A	Delete
ACADEMY 1	BOSTON	BADMINTON	75	A	Delete

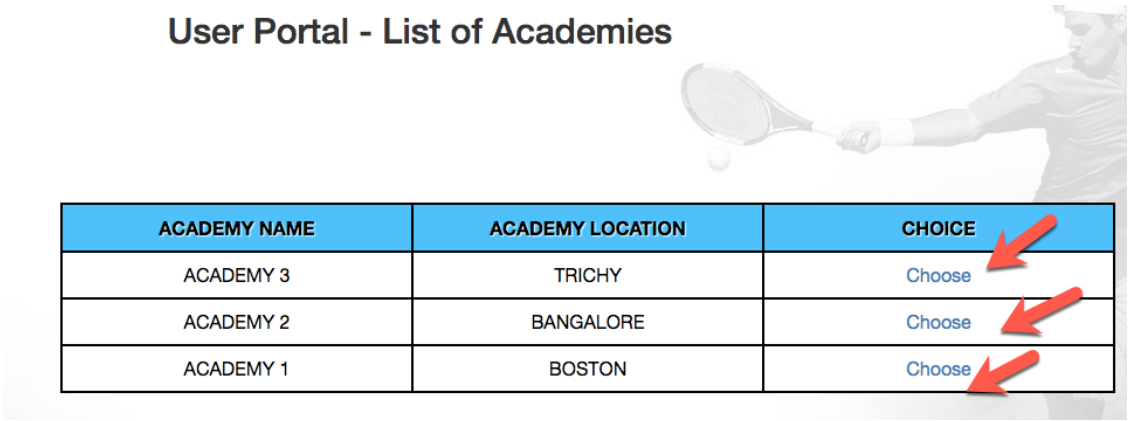
Step 4: End-user Login



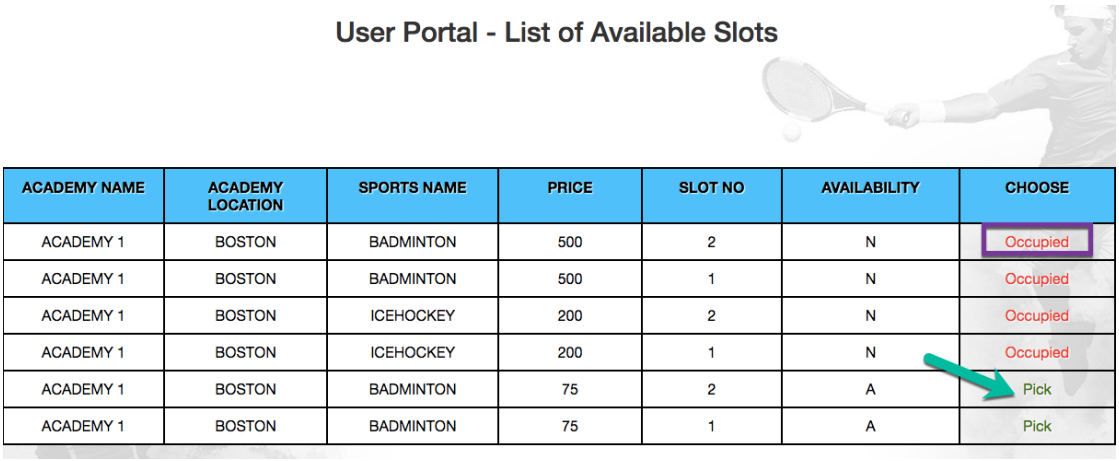
Step 6: End-user has 3 Functions

a. **Choose a sport of his/her interest.**

b. **Choose the Academy of his/her choice.**



c. Reserve a slot.



ACADEMY NAME	ACADEMY LOCATION	SPORTS NAME	PRICE	SLOT NO	AVAILABILITY	CHOOSE
ACADEMY 1	BOSTON	BADMINTON	500	2	N	Occupied
ACADEMY 1	BOSTON	BADMINTON	500	1	N	Occupied
ACADEMY 1	BOSTON	ICEHOCKEY	200	2	N	Occupied
ACADEMY 1	BOSTON	ICEHOCKEY	200	1	N	Occupied
ACADEMY 1	BOSTON	BADMINTON	75	2	A	Pick
ACADEMY 1	BOSTON	BADMINTON	75	1	A	Pick

Snap Shots of controllers

Admincontroller .docx

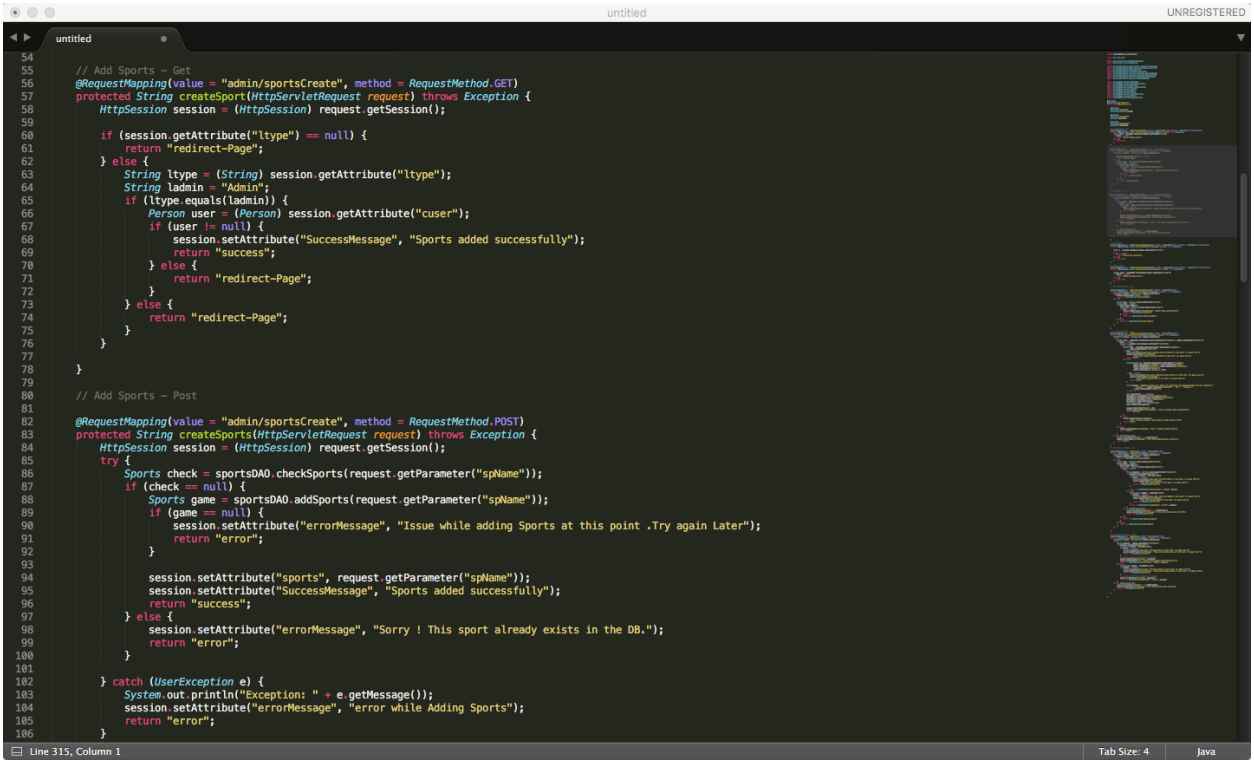
Endusercontroller.docx

service controller.docx

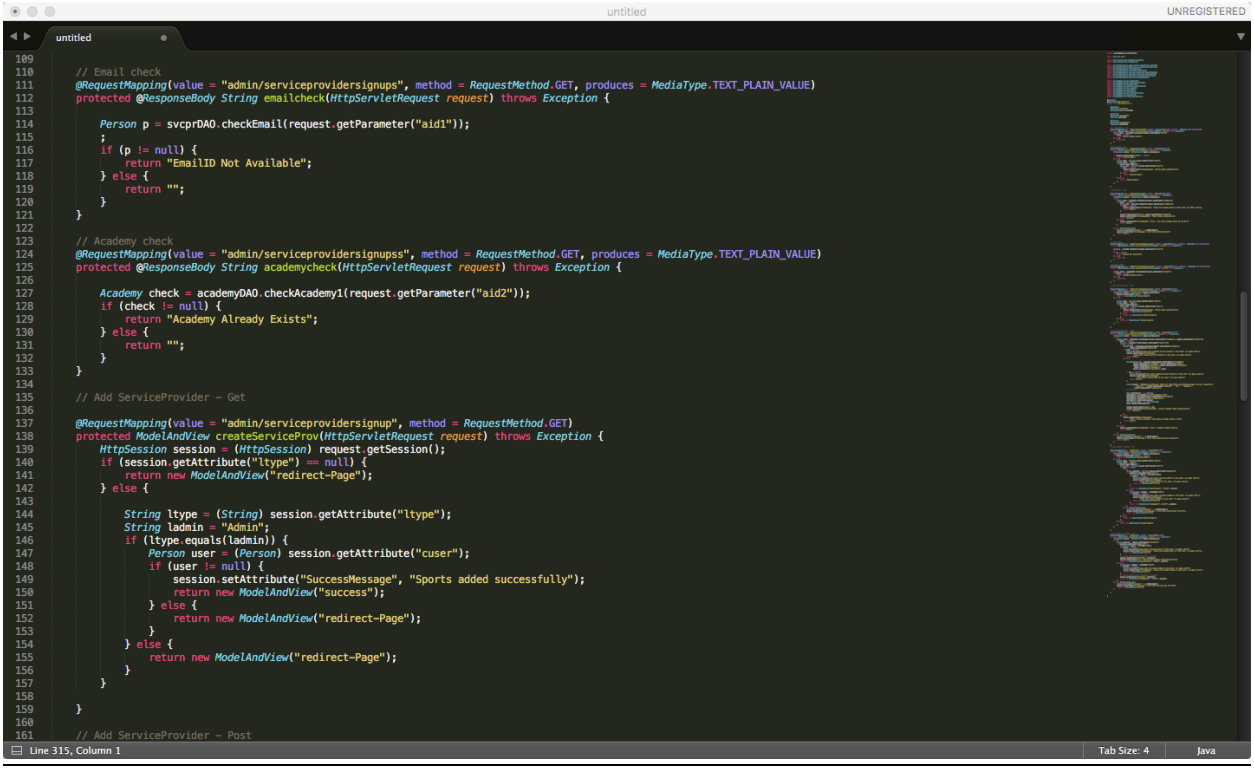
usercontroller .docx

1. Admincontroller

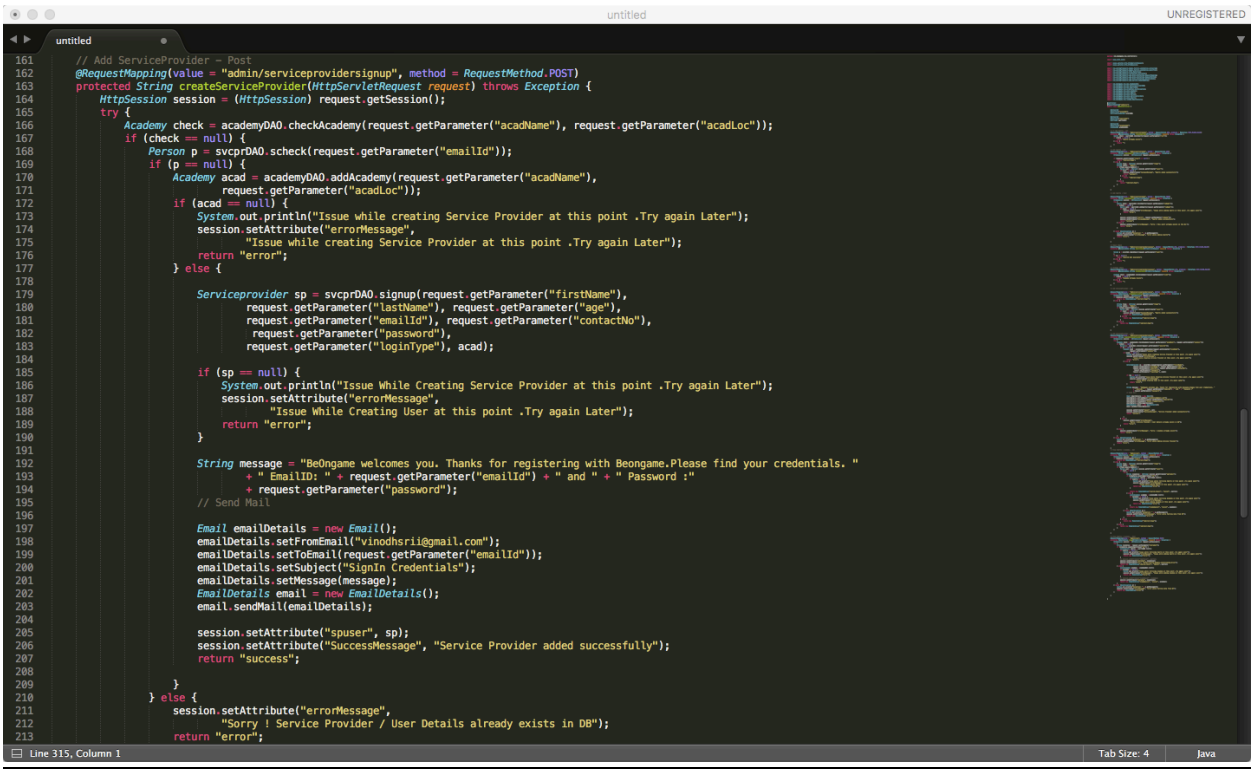
```java
package com.beongame.top.controllers;

import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.servlet.ModelAndView;

import com.beongame.top.dao.AcademyDAO;
import com.beongame.top.dao.ServiceproviderDAO;
import com.beongame.top.dao.SportsDAO;
import com.beongame.top.exception.UserException;
import com.beongame.top.pojo.Academy;
import com.beongame.top.pojo.Email;
import com.beongame.top.pojo.Person;
import com.beongame.top.pojo.Serviceprovider;
import com.beongame.top.pojo.Sports;
import com.beongame.top.random.EmailDetails;

@Controller
@RequestMapping("/admin/*")
public class Admincontroller {

    @Autowired
    @Qualifier("svcprDAO")
    ServiceproviderDAO svcprDAO;

    @Autowired
    @Qualifier("sportsDAO")
    SportsDAO sportsDAO;

    @Autowired
    @Qualifier("academyDAO")
    AcademyDAO academyDAO;

    // // Add Sports - Get
    @RequestMapping(value = "admin/sportsCreates", method = RequestMethod.GET, produces = MediaType.TEXT_PLAIN_VALUE)
    protected @ResponseBody String createSportaj(HttpServletRequest request) throws Exception {
        Sports check = sportsDAO.checkSports(request.getParameter("aid"));
        if (check != null) {
            return "Sports already exists";
        } else {
            return "";
        }
    }
```



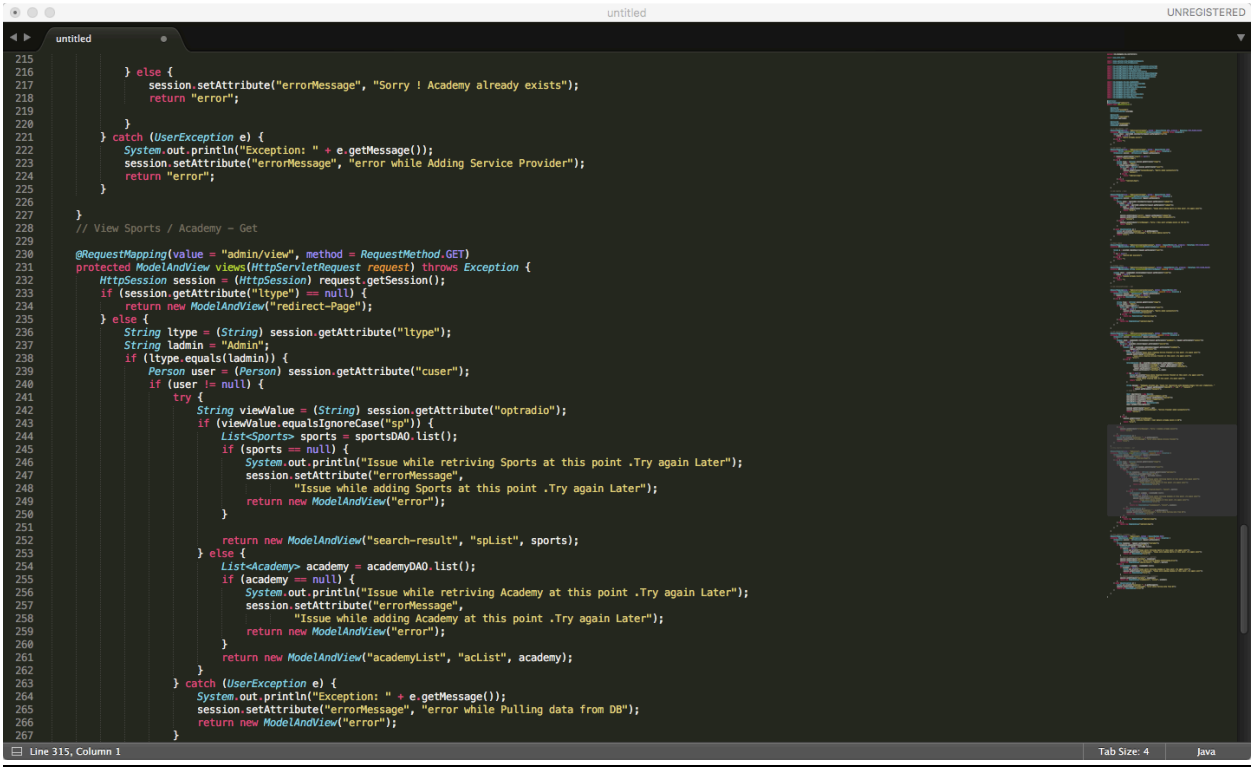
```
untitled                                                                    UNREGISTERED

untitled

54
55      // Add Sports - Get
56      @RequestMapping(value = "admin/sportsCreate", method = RequestMethod.GET)
57      protected String createSport(HttpServletRequest request) throws Exception {
58          HttpSession session = (HttpSession) request.getSession();
59
60          if (session.getAttribute("ltype") == null) {
61              return "redirect-Page";
62          } else {
63              String ltype = (String) session.getAttribute("ltype");
64              String ladmin = "Admin";
65              if (ltype.equals(ladmin)) {
66                  Person user = (Person) session.getAttribute("cuser");
67                  if (user != null) {
68                      session.setAttribute("SuccessMessage", "Sports added successfully");
69                      return "success";
70                  } else {
71                      return "redirect-Page";
72                  }
73              } else {
74                  return "redirect-Page";
75              }
76          }
77
78      }
79
80      // Add Sports - Post
81
82      @RequestMapping(value = "admin/sportsCreate", method = RequestMethod.POST)
83      protected String createSports(HttpServletRequest request) throws Exception {
84          HttpSession session = (HttpSession) request.getSession();
85          try {
86              Sports check = sportsDAO.checkSports(request.getParameter("spName"));
87              if (check == null) {
88                  Sports game = sportsDAO.addSports(request.getParameter("spName"));
89                  if (game == null) {
90                      session.setAttribute("errorMessage", "Issue while adding Sports at this point .Try again Later");
91                      return "error";
92                  }
93
94                  session.setAttribute("sports", request.getParameter("spName"));
95                  session.setAttribute("SuccessMessage", "Sports added successfully");
96                  return "success";
97              } else {
98                  session.setAttribute("errorMessage", "Sorry ! This sport already exists in the DB.");
99                  return "error";
100             }
101
102         } catch (UserException e) {
103             System.out.println("Exception: " + e.getMessage());
104             session.setAttribute("errorMessage", "error while Adding Sports");
105             return "error";
106         }

Line 315, Column 1                                                  Tab Size: 4    Java
```



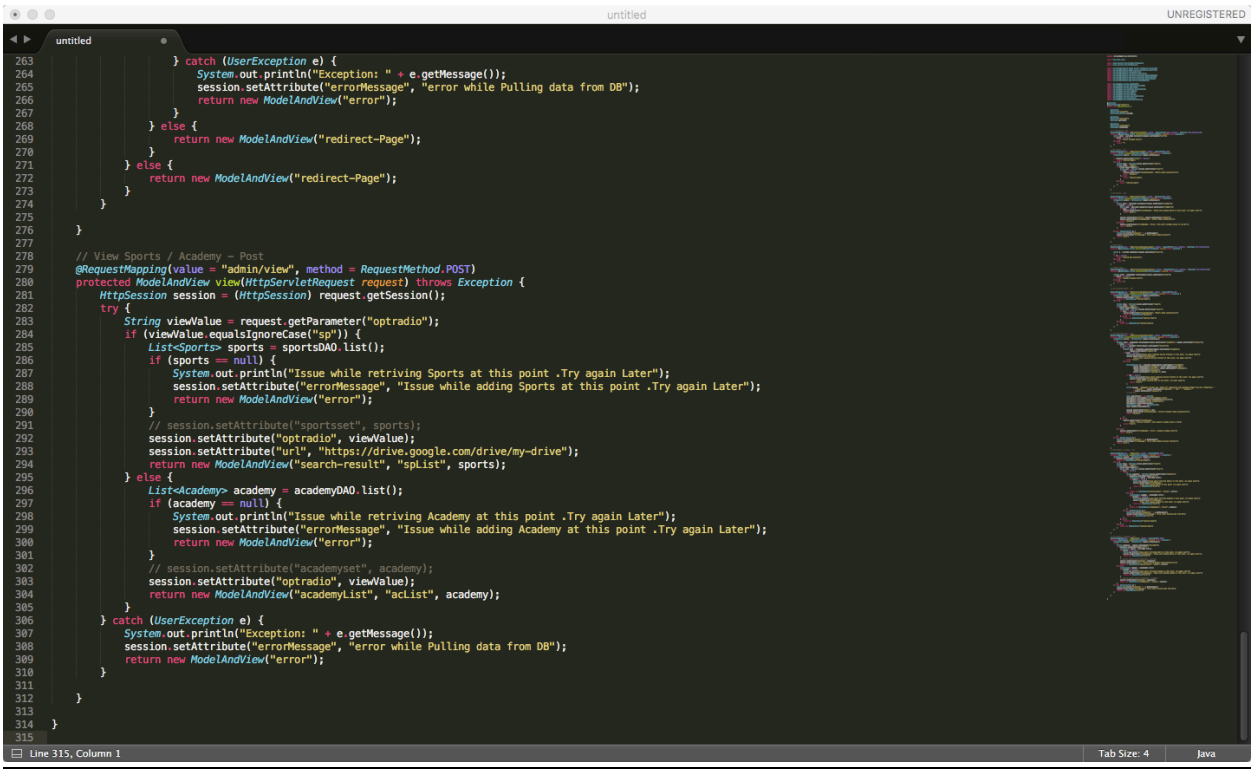
```
untitled                                                                    UNREGISTERED

untitled

109
110     // Email check
111     @RequestMapping(value = "admin/serviceprovidersignups", method = RequestMethod.GET, produces = MediaType.TEXT_PLAIN_VALUE)
112     protected @ResponseBody String emailcheck(HttpServletRequest request) throws Exception {
113
114         Person p = svcprDAO.checkEmail(request.getParameter("aid1"));
115         ;
116         if (p != null) {
117             return "EmailID Not Available";
118         } else {
119             return "";
120         }
121     }
122
123     // Academy check
124     @RequestMapping(value = "admin/serviceprovidersignupss", method = RequestMethod.GET, produces = MediaType.TEXT_PLAIN_VALUE)
125     protected @ResponseBody String academycheck(HttpServletRequest request) throws Exception {
126
127         Academy check = academyDAO.checkAcademy1(request.getParameter("aid2"));
128         if (check != null) {
129             return "Academy Already Exists";
130         } else {
131             return "";
132         }
133     }
134
135     // Add ServiceProvider - Get
136
137     @RequestMapping(value = "admin/serviceprovidersignup", method = RequestMethod.GET)
138     protected ModelAndView createServiceProv(HttpServletRequest request) throws Exception {
139         HttpSession session = (HttpSession) request.getSession();
140         if (session.getAttribute("ltype") == null) {
141             return new ModelAndView("redirect-Page");
142         } else {
143
144             String ltype = (String) session.getAttribute("ltype");
145             String ladmin = "Admin";
146             if (ltype.equals(ladmin)) {
147                 Person user = (Person) session.getAttribute("cuser");
148                 if (user != null) {
149                     session.setAttribute("SuccessMessage", "Sports added successfully");
150                     return new ModelAndView("success");
151                 } else {
152                     return new ModelAndView("redirect-Page");
153                 }
154             } else {
155                 return new ModelAndView("redirect-Page");
156             }
157         }
158
159     }
160
161     // Add ServiceProvider - Post

Line 315, Column 1                                                  Tab Size: 4    Java
```



```
// Add ServiceProvider - Post
@RequestMapping(value = "admin/serviceprovidersignup", method = RequestMethod.POST)
protected String createServiceProvider(HttpServletRequest request) throws Exception {
    HttpSession session = (HttpSession) request.getSession();
    try {
        Academy check = academyDAO.checkAcademy(request.getParameter("acadName"), request.getParameter("acadLoc"));
        if (check == null) {
            Person p = svcprDAO.scheck(request.getParameter("emailId"));
            if (p == null) {
                Academy acad = academyDAO.addAcademy(request.getParameter("acadName"),
                        request.getParameter("acadLoc"));
                if (acad == null) {
                    System.out.println("Issue while creating Service Provider at this point .Try again Later");
                    session.setAttribute("errorMessage",
                            "Issue while creating Service Provider at this point .Try again Later");
                    return "error";
                } else {

                    Serviceprovider sp = svcprDAO.signup(request.getParameter("firstName"),
                            request.getParameter("lastName"), request.getParameter("age"),
                            request.getParameter("emailId"), request.getParameter("contactNo"),
                            request.getParameter("password"),
                            request.getParameter("loginType"), acad);

                    if (sp == null) {
                        System.out.println("Issue While Creating Service Provider at this point .Try again Later");
                        session.setAttribute("errorMessage",
                                "Issue While Creating User at this point .Try again Later");
                        return "error";
                    }

                    String message = "BeOngame welcomes you. Thanks for registering with Beongame.Please find your credentials. "
                            + " EmailID: " + request.getParameter("emailId") + " and " + " Password :"
                            + request.getParameter("password");
                    // Send Mail

                    Email emailDetails = new Email();
                    emailDetails.setFromEmail("vinodhsrii@gmail.com");
                    emailDetails.setToEmail(request.getParameter("emailId"));
                    emailDetails.setSubject("SignIn Credentials");
                    emailDetails.setMessage(message);
                    EmailDetails email = new EmailDetails();
                    email.sendMail(emailDetails);

                    session.setAttribute("spuser", sp);
                    session.setAttribute("SuccessMessage", "Service Provider added successfully");
                    return "success";

                }
            } else {
                session.setAttribute("errorMessage",
                        "Sorry ! Service Provider / User Details already exists in DB");
                return "error";
```



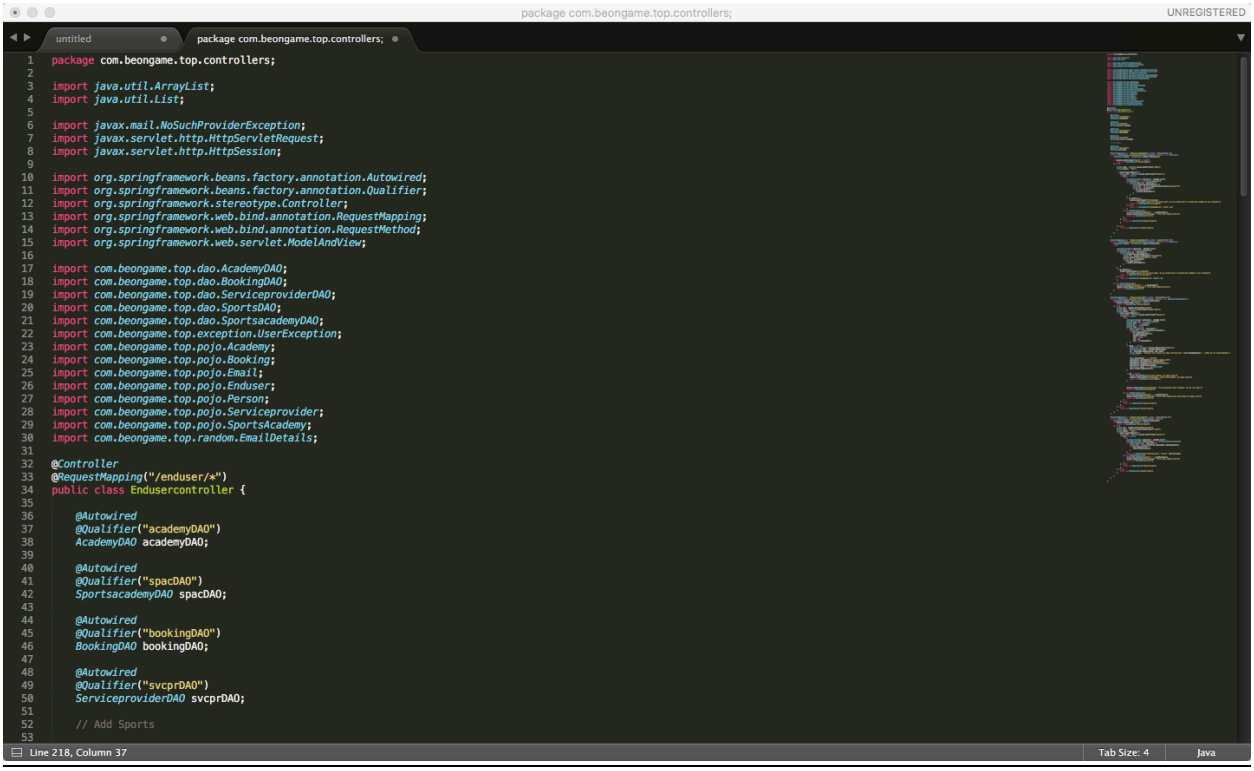
```
            } else {
                session.setAttribute("errorMessage", "Sorry ! Academy already exists");
                return "error";

            }
        } catch (UserException e) {
            System.out.println("Exception: " + e.getMessage());
            session.setAttribute("errorMessage", "error while Adding Service Provider");
            return "error";
        }

    }
    // View Sports / Academy - Get

    @RequestMapping(value = "admin/view", method = RequestMethod.GET)
    protected ModelAndView views(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        if (session.getAttribute("ltype") == null) {
            return new ModelAndView("redirect-Page");
        } else {
            String ltype = (String) session.getAttribute("ltype");
            String ladmin = "Admin";
            if (ltype.equals(ladmin)) {
                Person user = (Person) session.getAttribute("cuser");
                if (user != null) {
                    try {
                        String viewValue = (String) session.getAttribute("optradio");
                        if (viewValue.equalsIgnoreCase("sp")) {
                            List<Sports> sports = sportsDAO.list();
                            if (sports == null) {
                                System.out.println("Issue while retriving Sports at this point .Try again Later");
                                session.setAttribute("errorMessage",
                                        "Issue while adding Sports at this point .Try again Later");
                                return new ModelAndView("error");
                            }

                            return new ModelAndView("search-result", "spList", sports);
                        } else {
                            List<Academy> academy = academyDAO.list();
                            if (academy == null) {
                                System.out.println("Issue while retriving Academy at this point .Try again Later");
                                session.setAttribute("errorMessage",
                                        "Issue while adding Academy at this point .Try again Later");
                                return new ModelAndView("error");
                            }
                            return new ModelAndView("academyList", "acList", academy);
                        }
                    } catch (UserException e) {
                        System.out.println("Exception: " + e.getMessage());
                        session.setAttribute("errorMessage", "error while Pulling data from DB");
                        return new ModelAndView("error");
                    }
```



```
                    } catch (UserException e) {
                        System.out.println("Exception: " + e.getMessage());
                        session.setAttribute("errorMessage", "error while Pulling data from DB");
                        return new ModelAndView("error");
                    }
                } else {
                    return new ModelAndView("redirect-Page");
                }
            } else {
                return new ModelAndView("redirect-Page");
            }
        }

    }

    // View Sports / Academy - Post
    @RequestMapping(value = "admin/view", method = RequestMethod.POST)
    protected ModelAndView view(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        try {
            String viewValue = request.getParameter("optradio");
            if (viewValue.equalsIgnoreCase("sp")) {
                List<Sports> sports = sportsDAO.list();
                if (sports == null) {
                    System.out.println("Issue while retriving Sports at this point .Try again Later");
                    session.setAttribute("errorMessage", "Issue while adding Sports at this point .Try again Later");
                    return new ModelAndView("error");
                }
                // session.setAttribute("sportsset", sports);
                session.setAttribute("optradio", viewValue);
                session.setAttribute("url", "https://drive.google.com/drive/my-drive");
                return new ModelAndView("search-result", "spList", sports);
            } else {
                List<Academy> academy = academyDAO.list();
                if (academy == null) {
                    System.out.println("Issue while retriving Academy at this point .Try again Later");
                    session.setAttribute("errorMessage", "Issue while adding Academy at this point .Try again Later");
                    return new ModelAndView("error");
                }
                // session.setAttribute("academyset", academy);
                session.setAttribute("optradio", viewValue);
                return new ModelAndView("academyList", "acList", academy);
            }
        } catch (UserException e) {
            System.out.println("Exception: " + e.getMessage());
            session.setAttribute("errorMessage", "error while Pulling data from DB");
            return new ModelAndView("error");
        }

    }
}

```

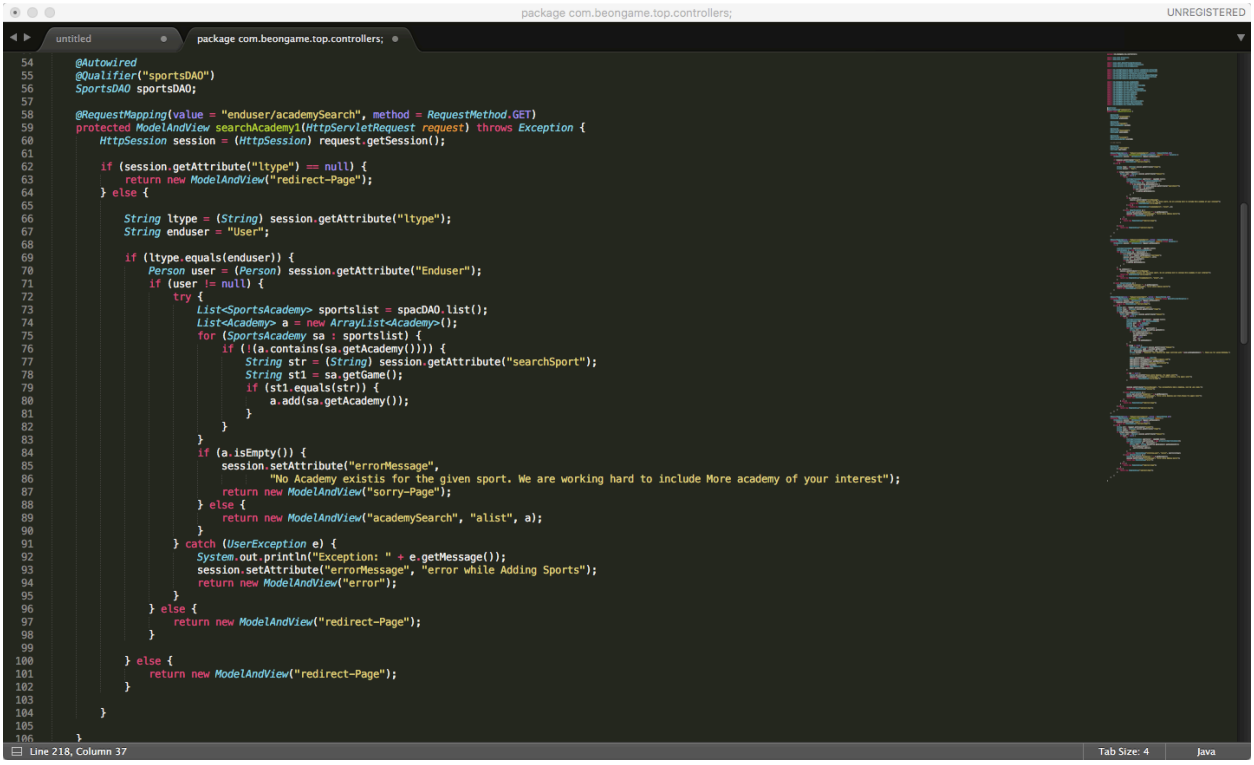
Line 315, Column 1 — Tab Size: 4 — Java

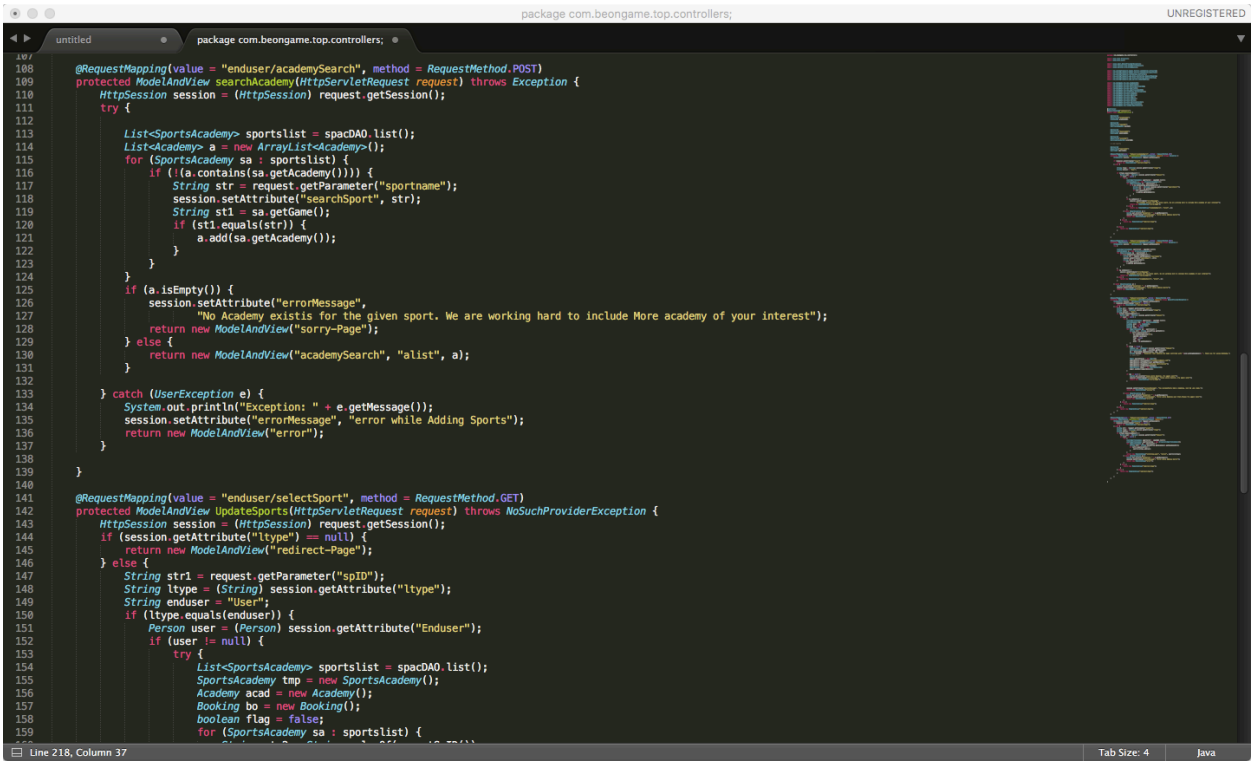
2. Endusercontroller

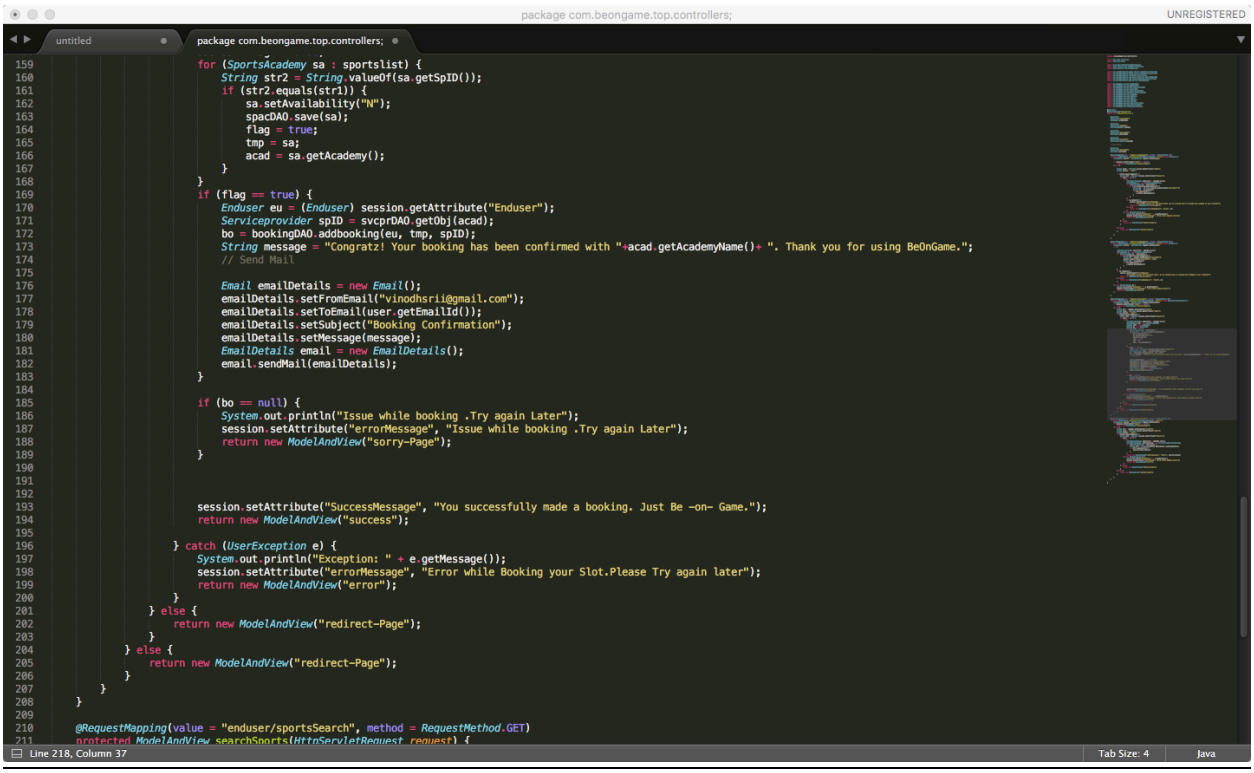


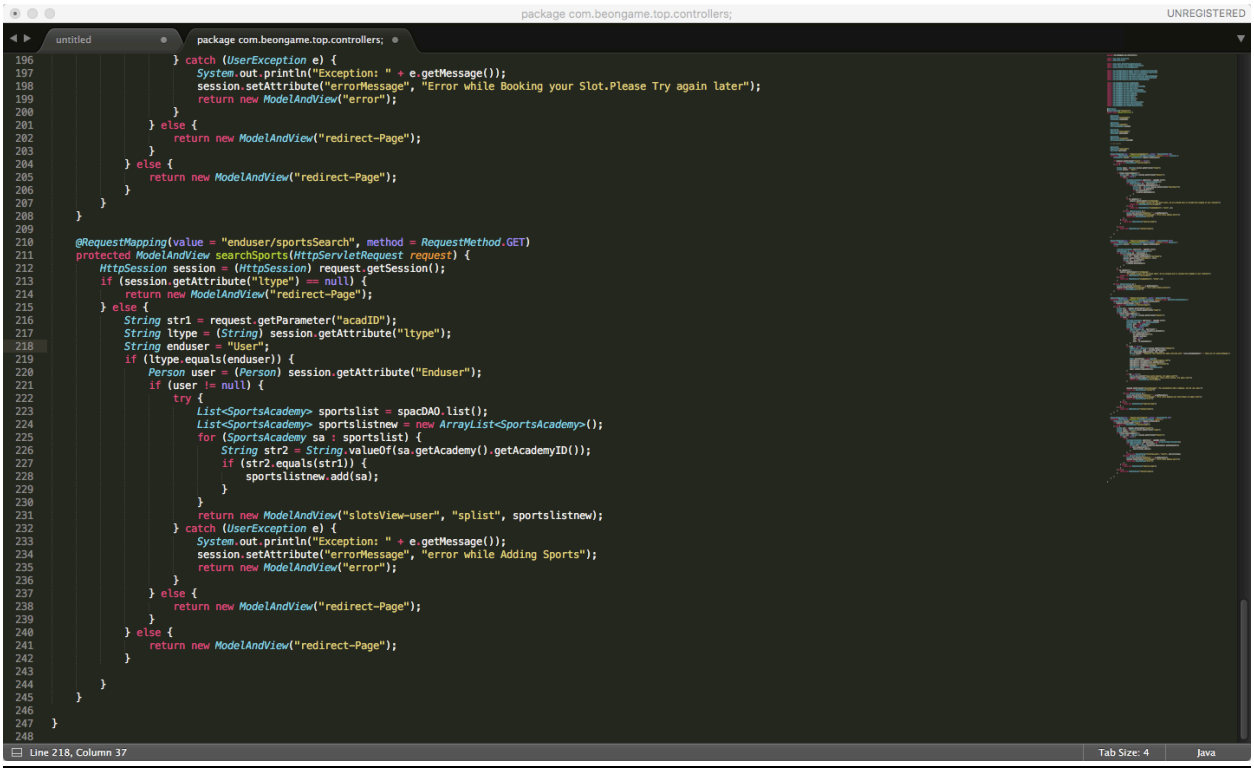
```
package com.beongame.top.controllers;

import java.util.ArrayList;
import java.util.List;

import javax.mail.NoSuchProviderException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.beongame.top.dao.AcademyDAO;
import com.beongame.top.dao.BookingDAO;
import com.beongame.top.dao.ServiceproviderDAO;
import com.beongame.top.dao.SportsDAO;
import com.beongame.top.dao.SportsacademyDAO;
import com.beongame.top.exception.UserException;
import com.beongame.top.pojo.Academy;
import com.beongame.top.pojo.Booking;
import com.beongame.top.pojo.Email;
import com.beongame.top.pojo.Enduser;
import com.beongame.top.pojo.Person;
import com.beongame.top.pojo.Serviceprovider;
import com.beongame.top.pojo.SportsAcademy;
import com.beongame.top.random.EmailDetails;

@Controller
@RequestMapping("/enduser/*")
public class Endusercontroller {

    @Autowired
    @Qualifier("academyDAO")
    AcademyDAO academyDAO;

    @Autowired
    @Qualifier("spacDAO")
    SportsacademyDAO spacDAO;

    @Autowired
    @Qualifier("bookingDAO")
    BookingDAO bookingDAO;

    @Autowired
    @Qualifier("svcprDAO")
    ServiceproviderDAO svcprDAO;

    // Add Sports

```

Line 218, Column 37 — Tab Size: 4 — Java

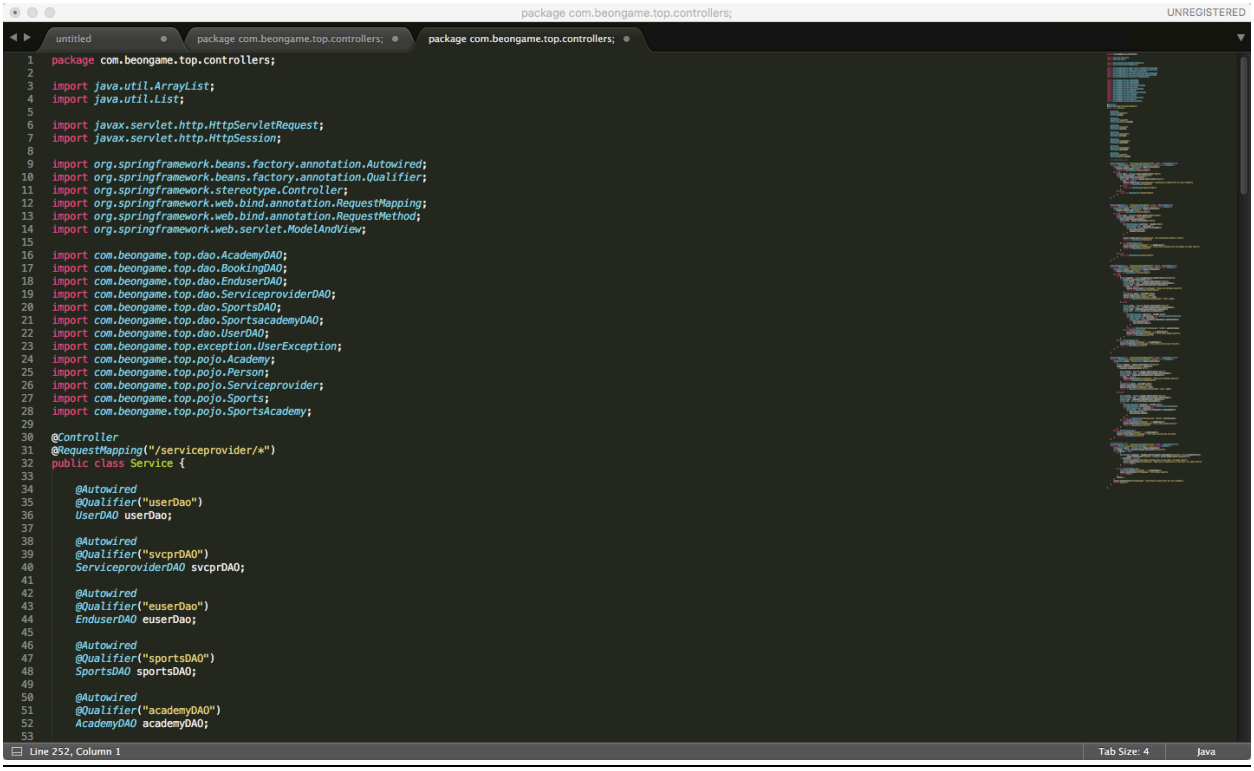

package com.beongame.top.controllers;
UNREGISTERED
untitled
package com.beongame.top.controllers;
@Autowired
@Qualifier("sportsDAO")
SportsDAO sportsDAO;
@RequestMapping(value = "enduser/academySearch", method = RequestMethod.GET)
protected ModelAndView searchAcademy1(HttpServletRequest request) throws Exception {
HttpSession session = (HttpSession) request.getSession();
if (session.getAttribute("ltype") == null) {
return new ModelAndView("redirect-Page");
} else {
String ltype = (String) session.getAttribute("ltype");
String enduser = "User";
if (ltype.equals(enduser)) {
Person user = (Person) session.getAttribute("Enduser");
if (user != null) {
try {
List<SportsAcademy> sportslist = spacDAO.list();
List<Academy> a = new ArrayList<Academy>();
for (SportsAcademy sa : sportslist) {
if (!(a.contains(sa.getAcademy()))) {
String str = (String) session.getAttribute("searchSport");
String st1 = sa.getGame();
if (st1.equals(str)) {
a.add(sa.getAcademy());
}
}
}
if (a.isEmpty()) {
session.setAttribute("errorMessage",
"No Academy existis for the given sport. We are working hard to include More academy of your interest");
return new ModelAndView("sorry-Page");
} else {
return new ModelAndView("academySearch", "alist", a);
}
} catch (UserException e) {
System.out.println("Exception: " + e.getMessage());
session.setAttribute("errorMessage", "error while Adding Sports");
return new ModelAndView("error");
}
} else {
return new ModelAndView("redirect-Page");
}
} else {
return new ModelAndView("redirect-Page");
}
}
}
Line 218, Column 37
Tab Size: 4
Java


package com.beongame.top.controllers;
UNREGISTERED
untitled
package com.beongame.top.controllers;
@RequestMapping(value = "enduser/academySearch", method = RequestMethod.POST)
protected ModelAndView searchAcademy(HttpServletRequest request) throws Exception {
HttpSession session = (HttpSession) request.getSession();
try {
List<SportsAcademy> sportslist = spacDAO.list();
List<Academy> a = new ArrayList<Academy>();
for (SportsAcademy sa : sportslist) {
if (!(a.contains(sa.getAcademy()))) {
String str = request.getParameter("sportname");
session.setAttribute("searchSport", str);
String st1 = sa.getGame();
if (st1.equals(str)) {
a.add(sa.getAcademy());
}
}
}
if (a.isEmpty()) {
session.setAttribute("errorMessage",
"No Academy existis for the given sport. We are working hard to include More academy of your interest");
return new ModelAndView("sorry-Page");
} else {
return new ModelAndView("academySearch", "alist", a);
}
} catch (UserException e) {
System.out.println("Exception: " + e.getMessage());
session.setAttribute("errorMessage", "error while Adding Sports");
return new ModelAndView("error");
}
}
@RequestMapping(value = "enduser/selectSport", method = RequestMethod.GET)
protected ModelAndView UpdateSports(HttpServletRequest request) throws NoSuchProviderException {
HttpSession session = (HttpSession) request.getSession();
if (session.getAttribute("ltype") == null) {
return new ModelAndView("redirect-Page");
} else {
String str1 = request.getParameter("spID");
String ltype = (String) session.getAttribute("ltype");
String enduser = "User";
if (ltype.equals(enduser)) {
Person user = (Person) session.getAttribute("Enduser");
if (user != null) {
try {
List<SportsAcademy> sportslist = spacDAO.list();
SportsAcademy tmp = new SportsAcademy();
Academy acad = new Academy();
Booking bo = new Booking();
boolean flag = false;
for (SportsAcademy sa : sportslist) {
Line 218, Column 37
Tab Size: 4
Java

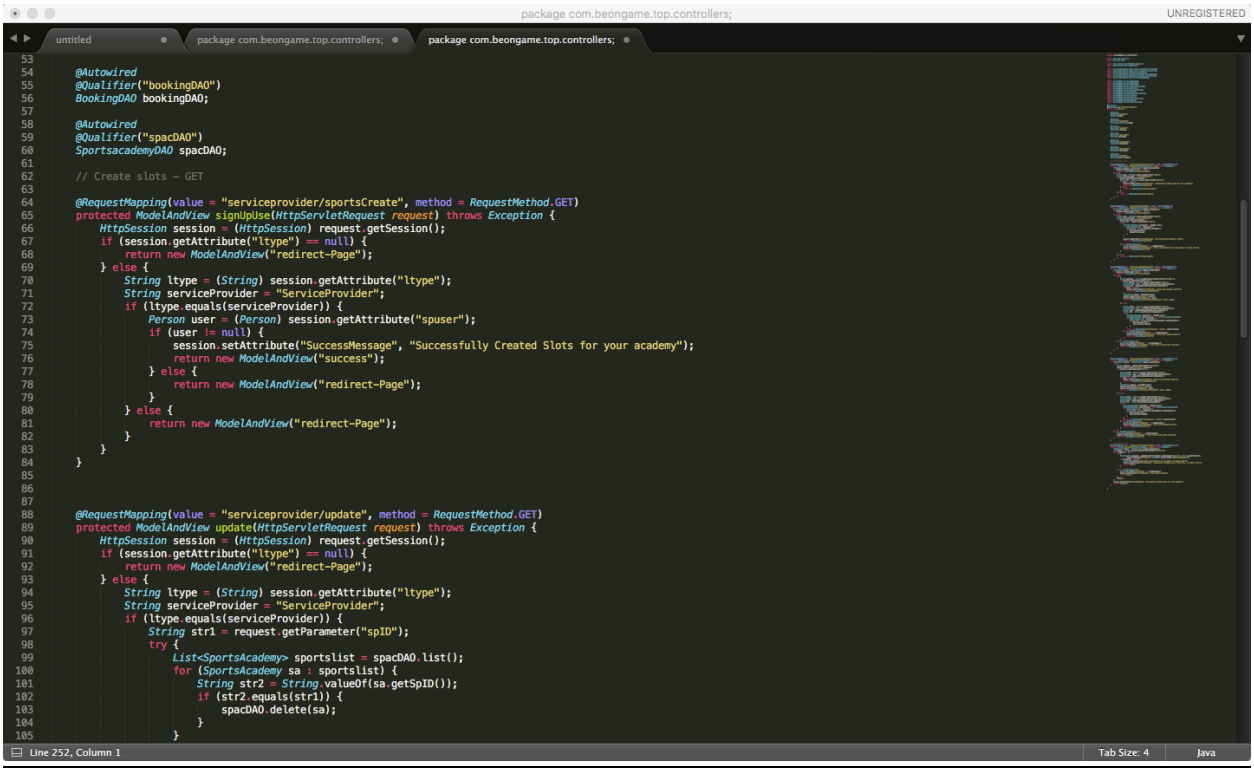

package com.beongame.top.controllers;
UNREGISTERED
untitled
package com.beongame.top.controllers;
for (SportsAcademy sa : sportslist) {
String str2 = String.valueOf(sa.getSpID());
if (str2.equals(str1)) {
sa.setAvailability("N");
spacDAO.save(sa);
flag = true;
tmp = sa;
acad = sa.getAcademy();
}
}
if (flag == true) {
Enduser eu = (Enduser) session.getAttribute("Enduser");
Serviceprovider spID = svcprDAO.getObj(acad);
bo = bookingDAO.addbooking(eu, tmp, spID);
String message = "Congratz! Your booking has been confirmed with "+acad.getAcademyName()+ ". Thank you for using BeOnGame.";
// Send Mail
Email emailDetails = new Email();
emailDetails.setFromEmail("vinodhsrii@gmail.com");
emailDetails.setToEmail(user.getEmailId());
emailDetails.setSubject("Booking Confirmation");
emailDetails.setMessage(message);
EmailDetails email = new EmailDetails();
email.sendMail(emailDetails);
}
if (bo == null) {
System.out.println("Issue while booking .Try again Later");
session.setAttribute("errorMessage", "Issue while booking .Try again Later");
return new ModelAndView("sorry-Page");
}
session.setAttribute("SuccessMessage", "You successfully made a booking. Just Be -on- Game.");
return new ModelAndView("success");
} catch (UserException e) {
System.out.println("Exception: " + e.getMessage());
session.setAttribute("errorMessage", "Error while Booking your Slot.Please Try again later");
return new ModelAndView("error");
}
} else {
return new ModelAndView("redirect-Page");
}
} else {
return new ModelAndView("redirect-Page");
}
}
}
@RequestMapping(value = "enduser/sportsSearch", method = RequestMethod.GET)
protected ModelAndView searchSports(HttpServletRequest request) {
Line 218, Column 37
Tab Size: 4
Java


package com.beongame.top.controllers;
UNREGISTERED
untitled
package com.beongame.top.controllers;
} catch (UserException e) {
System.out.println("Exception: " + e.getMessage());
session.setAttribute("errorMessage", "Error while Booking your Slot.Please Try again later");
return new ModelAndView("error");
}
} else {
return new ModelAndView("redirect-Page");
}
} else {
return new ModelAndView("redirect-Page");
}
}
}
@RequestMapping(value = "enduser/sportsSearch", method = RequestMethod.GET)
protected ModelAndView searchSports(HttpServletRequest request) {
HttpSession session = (HttpSession) request.getSession();
if (session.getAttribute("ltype") == null) {
return new ModelAndView("redirect-Page");
} else {
String str1 = request.getParameter("acadID");
String ltype = (String) session.getAttribute("ltype");
String enduser = "User";
if (ltype.equals(enduser)) {
Person user = (Person) session.getAttribute("Enduser");
if (user != null) {
try {
List<SportsAcademy> sportslist = spacDAO.list();
List<SportsAcademy> sportslistnew = new ArrayList<SportsAcademy>();
for (SportsAcademy sa : sportslist) {
String str2 = String.valueOf(sa.getAcademy().getAcademyID());
if (str2.equals(str1)) {
sportslistnew.add(sa);
}
}
return new ModelAndView("slotsView-user", "splist", sportslistnew);
} catch (UserException e) {
System.out.println("Exception: " + e.getMessage());
session.setAttribute("errorMessage", "error while Adding Sports");
return new ModelAndView("error");
}
} else {
return new ModelAndView("redirect-Page");
}
} else {
return new ModelAndView("redirect-Page");
}
}
}
}
Line 218, Column 37
Tab Size: 4
Java

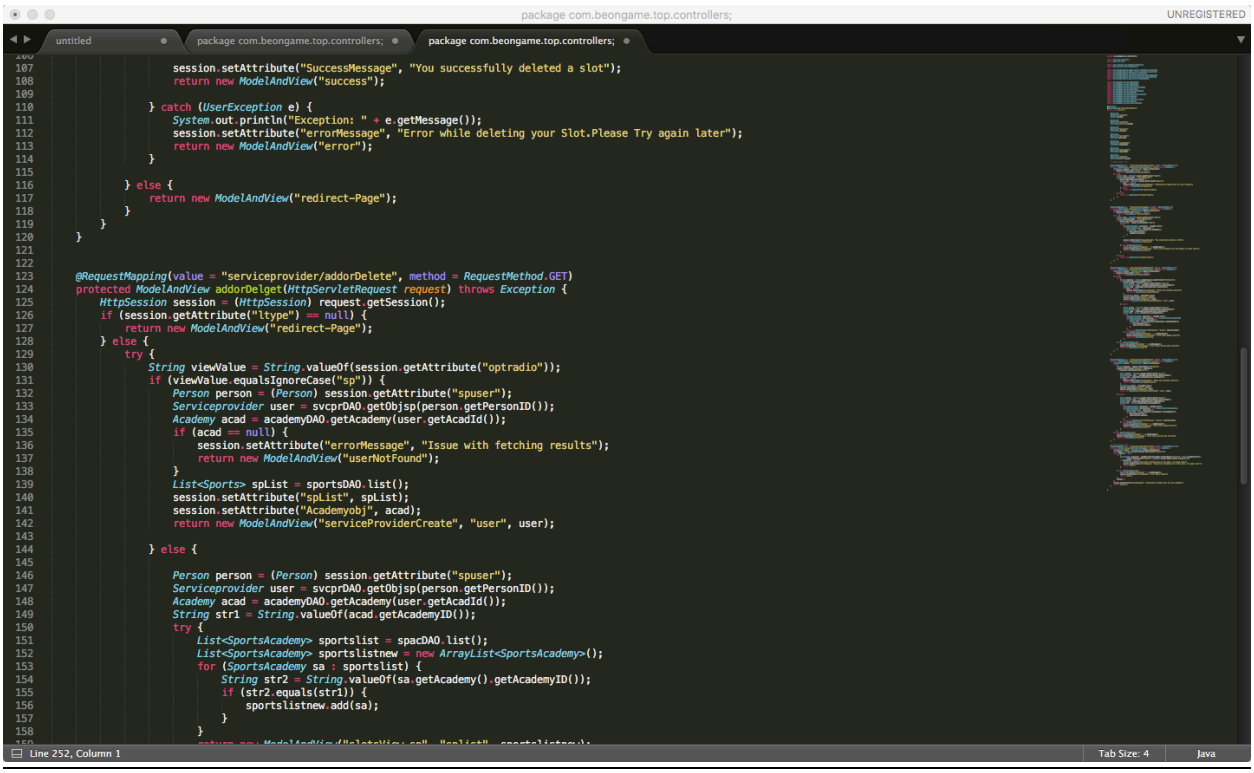
3. Servicecontroller

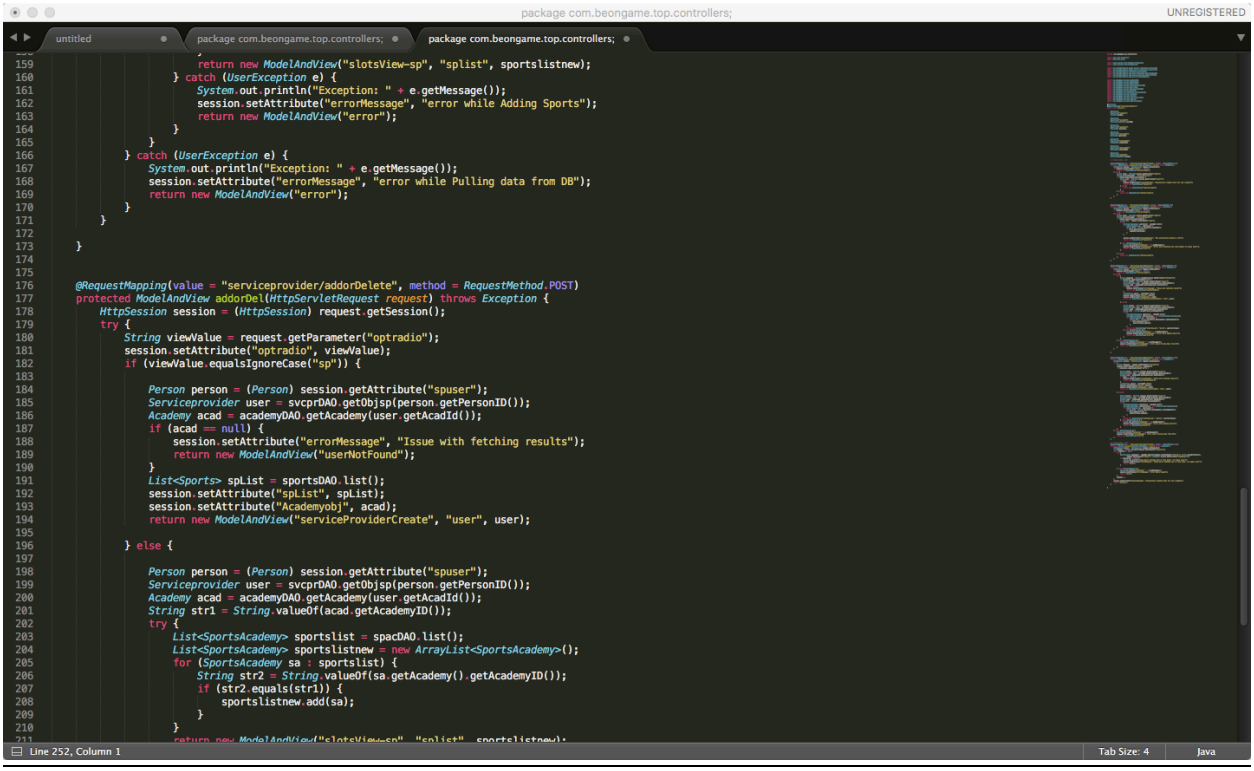


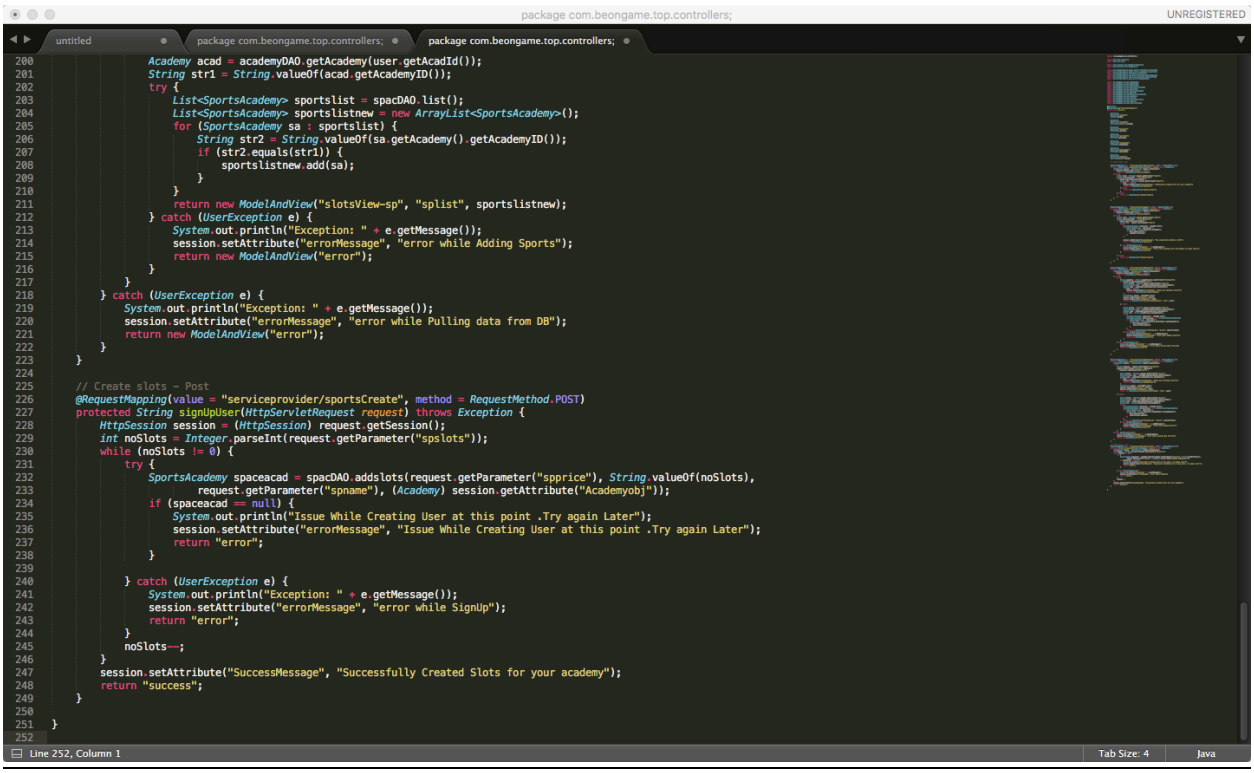
```
package com.beongame.top.controllers;

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.beongame.top.dao.AcademyDAO;
import com.beongame.top.dao.BookingDAO;
import com.beongame.top.dao.EnduserDAO;
import com.beongame.top.dao.ServiceproviderDAO;
import com.beongame.top.dao.SportsDAO;
import com.beongame.top.dao.SportsacademyDAO;
import com.beongame.top.dao.UserDAO;
import com.beongame.top.exception.UserException;
import com.beongame.top.pojo.Academy;
import com.beongame.top.pojo.Person;
import com.beongame.top.pojo.Serviceprovider;
import com.beongame.top.pojo.Sports;
import com.beongame.top.pojo.SportsAcademy;

@Controller
@RequestMapping("/serviceprovider/*")
public class Service {

    @Autowired
    @Qualifier("userDao")
    UserDAO userDao;

    @Autowired
    @Qualifier("svcprDAO")
    ServiceproviderDAO svcprDAO;

    @Autowired
    @Qualifier("euserDao")
    EnduserDAO euserDao;

    @Autowired
    @Qualifier("sportsDAO")
    SportsDAO sportsDAO;

    @Autowired
    @Qualifier("academyDAO")
    AcademyDAO academyDAO;
```



```
    @Autowired
    @Qualifier("bookingDAO")
    BookingDAO bookingDAO;

    @Autowired
    @Qualifier("spacDAO")
    SportsacademyDAO spacDAO;

    // Create slots - GET

    @RequestMapping(value = "serviceprovider/sportsCreate", method = RequestMethod.GET)
    protected ModelAndView signUpUse(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        if (session.getAttribute("ltype") == null) {
            return new ModelAndView("redirect-Page");
        } else {
            String ltype = (String) session.getAttribute("ltype");
            String serviceProvider = "ServiceProvider";
            if (ltype.equals(serviceProvider)) {
                Person user = (Person) session.getAttribute("spuser");
                if (user != null) {
                    session.setAttribute("SuccessMessage", "Successfully Created Slots for your academy");
                    return new ModelAndView("success");
                } else {
                    return new ModelAndView("redirect-Page");
                }
            } else {
                return new ModelAndView("redirect-Page");
            }
        }
    }


    @RequestMapping(value = "serviceprovider/update", method = RequestMethod.GET)
    protected ModelAndView update(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        if (session.getAttribute("ltype") == null) {
            return new ModelAndView("redirect-Page");
        } else {
            String ltype = (String) session.getAttribute("ltype");
            String serviceProvider = "ServiceProvider";
            if (ltype.equals(serviceProvider)) {
                String str1 = request.getParameter("spID");
                try {
                    List<SportsAcademy> sportslist = spacDAO.list();
                    for (SportsAcademy sa : sportslist) {
                        String str2 = String.valueOf(sa.getSpID());
                        if (str2.equals(str1)) {
                            spacDAO.delete(sa);
                        }
                    }
```

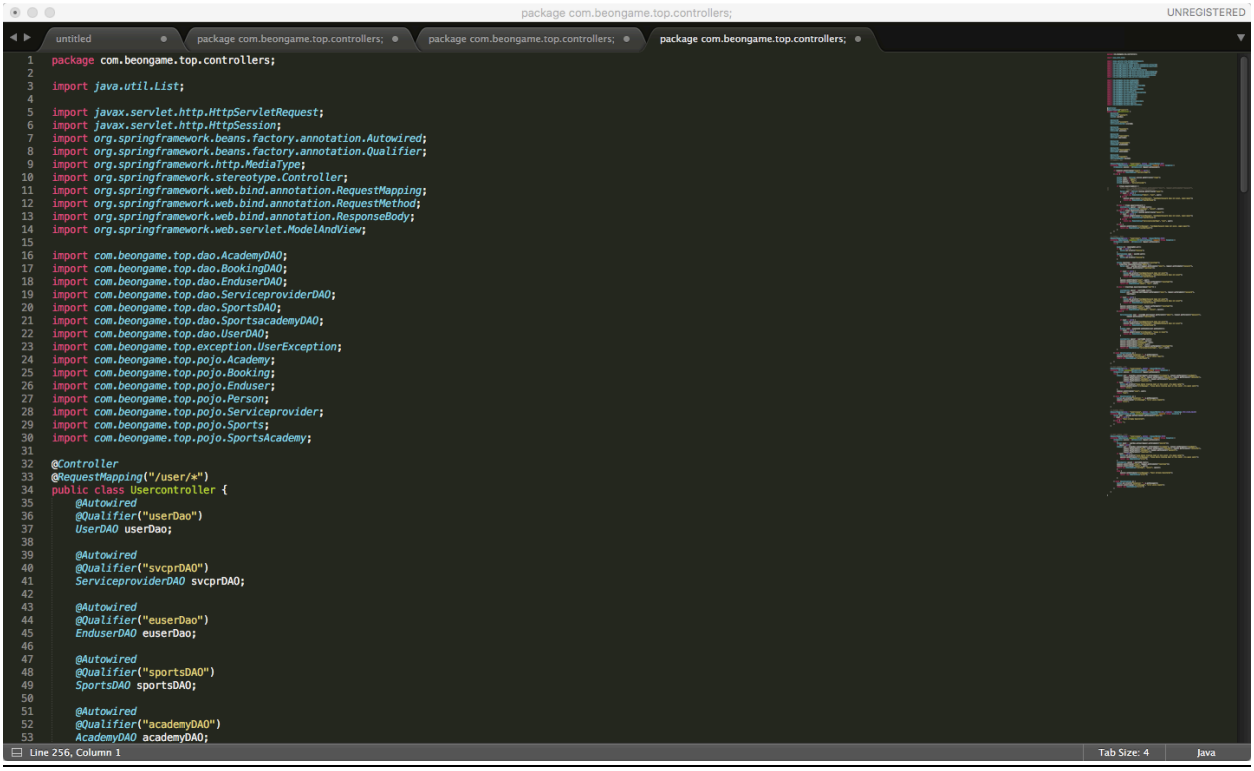

package com.beongame.top.controllers;
UNREGISTERED
untitled
package com.beongame.top.controllers;
package com.beongame.top.controllers;
Line 252, Column 1
Tab Size: 4
Java


package com.beongame.top.controllers;
UNREGISTERED
untitled
package com.beongame.top.controllers;
package com.beongame.top.controllers;
Line 252, Column 1
Tab Size: 4
Java

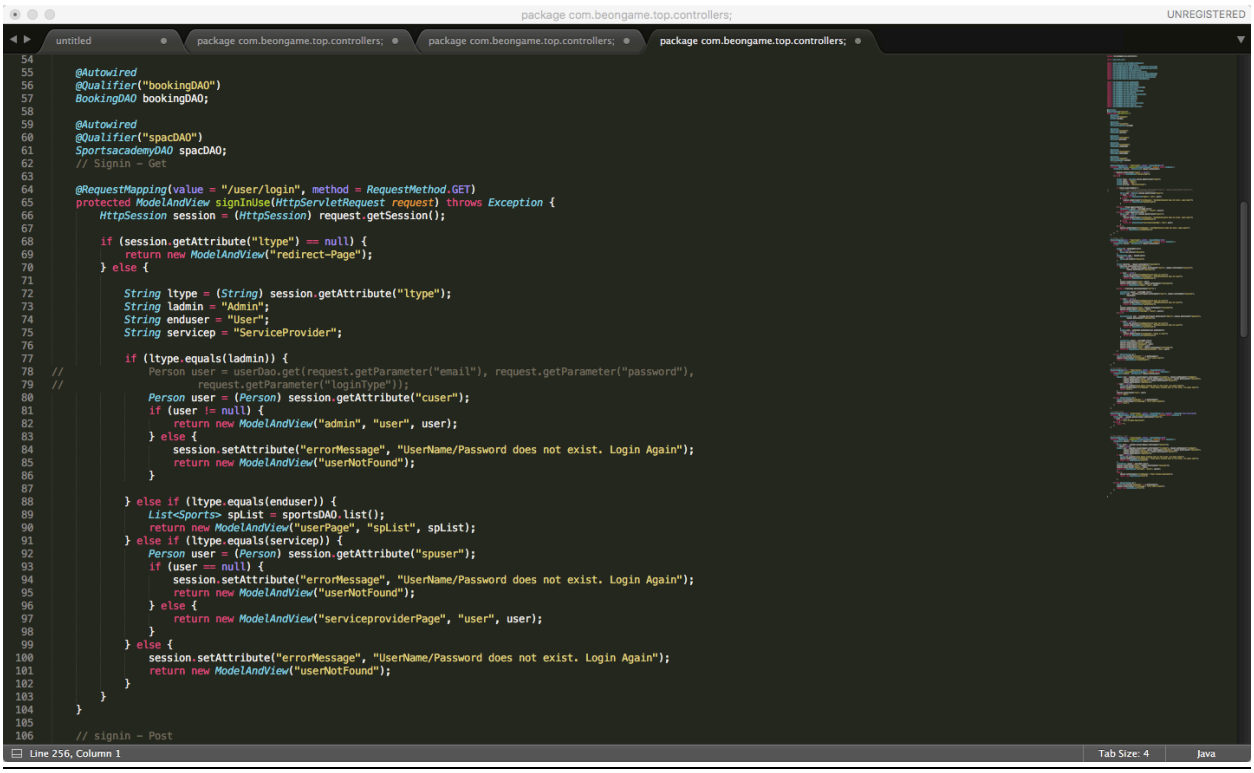



```
                Academy acad = academyDAO.getAcademy(user.getAcadId());
                String str1 = String.valueOf(acad.getAcademyID());
                try {
                    List<SportsAcademy> sportslist = spacDAO.list();
                    List<SportsAcademy> sportslistnew = new ArrayList<SportsAcademy>();
                    for (SportsAcademy sa : sportslist) {
                        String str2 = String.valueOf(sa.getAcademy().getAcademyID());
                        if (str2.equals(str1)) {
                            sportslistnew.add(sa);
                        }
                    }
                    return new ModelAndView("slotsView-sp", "splist", sportslistnew);
                } catch (UserException e) {
                    System.out.println("Exception: " + e.getMessage());
                    session.setAttribute("errorMessage", "error while Adding Sports");
                    return new ModelAndView("error");
                }
            }
        } catch (UserException e) {
            System.out.println("Exception: " + e.getMessage());
            session.setAttribute("errorMessage", "error while Pulling data from DB");
            return new ModelAndView("error");
        }
    }

    // Create slots - Post
    @RequestMapping(value = "serviceprovider/sportsCreate", method = RequestMethod.POST)
    protected String signUpUser(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        int noSlots = Integer.parseInt(request.getParameter("spslots"));
        while (noSlots != 0) {
            try {
                SportsAcademy spaceacad = spacDAO.addslots(request.getParameter("spprice"), String.valueOf(noSlots),
                        request.getParameter("spname"), (Academy) session.getAttribute("Academyobj"));
                if (spaceacad == null) {
                    System.out.println("Issue While Creating User at this point .Try again Later");
                    session.setAttribute("errorMessage", "Issue While Creating User at this point .Try again Later");
                    return "error";
                }

            } catch (UserException e) {
                System.out.println("Exception: " + e.getMessage());
                session.setAttribute("errorMessage", "error while SignUp");
                return "error";
            }
            noSlots--;
        }
        session.setAttribute("SuccessMessage", "Successfully Created Slots for your academy");
        return "success";
    }

}
```

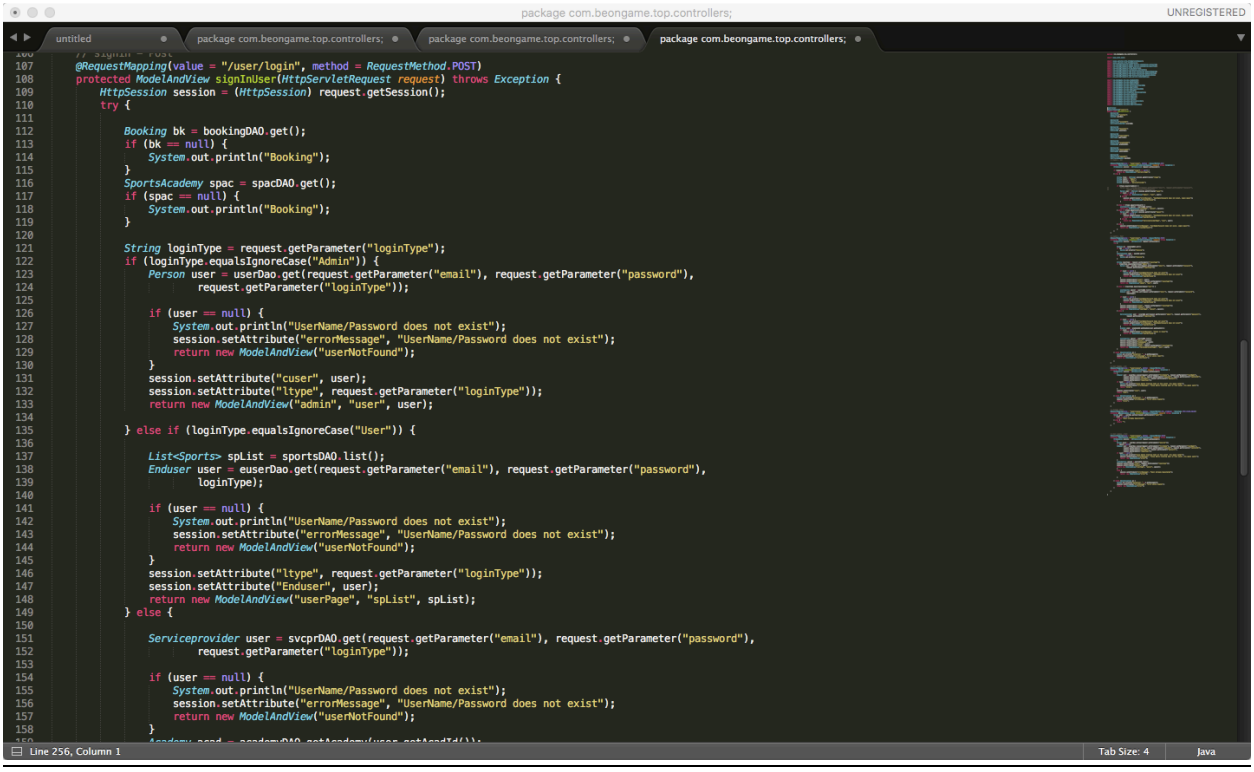
4. Usercontroller



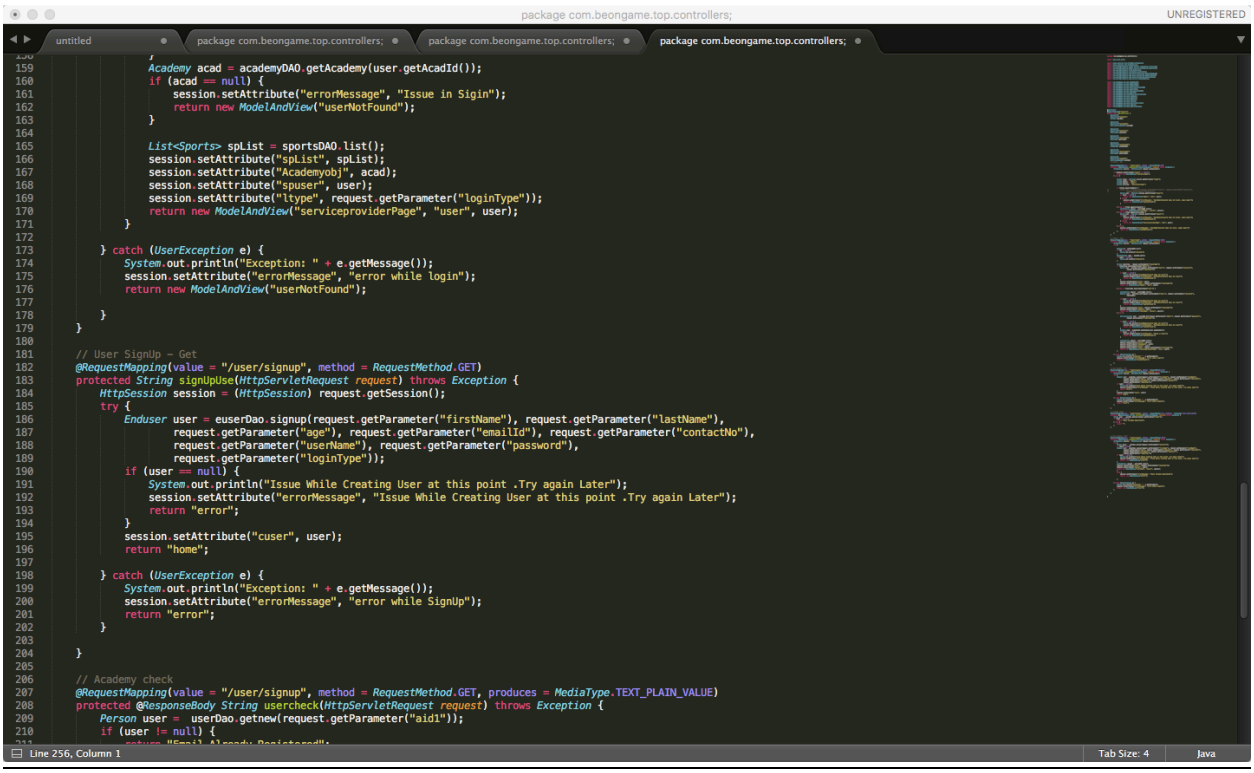

```
package com.beongame.top.controllers;

import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.http.MediaType;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.servlet.ModelAndView;

import com.beongame.top.dao.AcademyDAO;
import com.beongame.top.dao.BookingDAO;
import com.beongame.top.dao.EnduserDAO;
import com.beongame.top.dao.ServiceproviderDAO;
import com.beongame.top.dao.SportsDAO;
import com.beongame.top.dao.SportsacademyDAO;
import com.beongame.top.dao.UserDAO;
import com.beongame.top.exception.UserException;
import com.beongame.top.pojo.Academy;
import com.beongame.top.pojo.Booking;
import com.beongame.top.pojo.Enduser;
import com.beongame.top.pojo.Person;
import com.beongame.top.pojo.Serviceprovider;
import com.beongame.top.pojo.Sports;
import com.beongame.top.pojo.SportsAcademy;

@Controller
@RequestMapping("/user/*")
public class Usercontroller {
    @Autowired
    @Qualifier("userDao")
    UserDAO userDao;

    @Autowired
    @Qualifier("svcprDAO")
    ServiceproviderDAO svcprDAO;

    @Autowired
    @Qualifier("euserDao")
    EnduserDAO euserDao;

    @Autowired
    @Qualifier("sportsDAO")
    SportsDAO sportsDAO;

    @Autowired
    @Qualifier("academyDAO")
    AcademyDAO academyDAO;
```




```
    @Autowired
    @Qualifier("bookingDAO")
    BookingDAO bookingDAO;

    @Autowired
    @Qualifier("spacDAO")
    SportsacademyDAO spacDAO;
    // Signin - Get

    @RequestMapping(value = "/user/login", method = RequestMethod.GET)
    protected ModelAndView signInUse(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();

        if (session.getAttribute("ltype") == null) {
            return new ModelAndView("redirect-Page");
        } else {

            String ltype = (String) session.getAttribute("ltype");
            String ladmin = "Admin";
            String enduser = "User";
            String servicep = "ServiceProvider";

            if (ltype.equals(ladmin)) {
//              Person user = userDao.get(request.getParameter("email"), request.getParameter("password"),
//                      request.getParameter("loginType"));
                Person user = (Person) session.getAttribute("cuser");
                if (user != null) {
                    return new ModelAndView("admin", "user", user);
                } else {
                    session.setAttribute("errorMessage", "UserName/Password does not exist. Login Again");
                    return new ModelAndView("userNotFound");
                }

            } else if (ltype.equals(enduser)) {
                List<Sports> spList = sportsDAO.list();
                return new ModelAndView("userPage", "spList", spList);
            } else if (ltype.equals(servicep)) {
                Person user = (Person) session.getAttribute("spuser");
                if (user == null) {
                    session.setAttribute("errorMessage", "UserName/Password does not exist. Login Again");
                    return new ModelAndView("userNotFound");
                } else {
                    return new ModelAndView("serviceproviderPage", "user", user);
                }
            } else {
                session.setAttribute("errorMessage", "UserName/Password does not exist. Login Again");
                return new ModelAndView("userNotFound");
            }
        }
    }

    // signin - Post
```




```
    // Signin - Post
    @RequestMapping(value = "/user/login", method = RequestMethod.POST)
    protected ModelAndView signInUser(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        try {

            Booking bk = bookingDAO.get();
            if (bk == null) {
                System.out.println("Booking");
            }
            SportsAcademy spac = spacDAO.get();
            if (spac == null) {
                System.out.println("Booking");
            }

            String loginType = request.getParameter("loginType");
            if (loginType.equalsIgnoreCase("Admin")) {
                Person user = userDao.get(request.getParameter("email"), request.getParameter("password"),
                        request.getParameter("loginType"));

                if (user == null) {
                    System.out.println("UserName/Password does not exist");
                    session.setAttribute("errorMessage", "UserName/Password does not exist");
                    return new ModelAndView("userNotFound");
                }
                session.setAttribute("cuser", user);
                session.setAttribute("ltype", request.getParameter("loginType"));
                return new ModelAndView("admin", "user", user);

            } else if (loginType.equalsIgnoreCase("User")) {

                List<Sports> spList = sportsDAO.list();
                Enduser user = euserDao.get(request.getParameter("email"), request.getParameter("password"),
                        loginType);

                if (user == null) {
                    System.out.println("UserName/Password does not exist");
                    session.setAttribute("errorMessage", "UserName/Password does not exist");
                    return new ModelAndView("userNotFound");
                }
                session.setAttribute("ltype", request.getParameter("loginType"));
                session.setAttribute("Enduser", user);
                return new ModelAndView("userPage", "spList", spList);
            } else {

                Serviceprovider user = svcprDAO.get(request.getParameter("email"), request.getParameter("password"),
                        request.getParameter("loginType"));

                if (user == null) {
                    System.out.println("UserName/Password does not exist");
                    session.setAttribute("errorMessage", "UserName/Password does not exist");
                    return new ModelAndView("userNotFound");
                }
                Academy acad = academyDAO.getAcademy(user.getAcadId());
```




```
                Academy acad = academyDAO.getAcademy(user.getAcadId());
                if (acad == null) {
                    session.setAttribute("errorMessage", "Issue in Sigin");
                    return new ModelAndView("userNotFound");
                }

                List<Sports> spList = sportsDAO.list();
                session.setAttribute("spList", spList);
                session.setAttribute("Academyobj", acad);
                session.setAttribute("spuser", user);
                session.setAttribute("ltype", request.getParameter("loginType"));
                return new ModelAndView("serviceproviderPage", "user", user);
            }

        } catch (UserException e) {
            System.out.println("Exception: " + e.getMessage());
            session.setAttribute("errorMessage", "error while login");
            return new ModelAndView("userNotFound");

        }
    }

    // User SignUp - Get
    @RequestMapping(value = "/user/signup", method = RequestMethod.GET)
    protected String signUpUse(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        try {
            Enduser user = euserDao.signup(request.getParameter("firstName"), request.getParameter("lastName"),
                    request.getParameter("age"), request.getParameter("emailId"), request.getParameter("contactNo"),
                    request.getParameter("userName"), request.getParameter("password"),
                    request.getParameter("loginType"));
            if (user == null) {
                System.out.println("Issue While Creating User at this point .Try again Later");
                session.setAttribute("errorMessage", "Issue While Creating User at this point .Try again Later");
                return "error";
            }
            session.setAttribute("cuser", user);
            return "home";

        } catch (UserException e) {
            System.out.println("Exception: " + e.getMessage());
            session.setAttribute("errorMessage", "error while SignUp");
            return "error";
        }

    }

    // Academy check
    @RequestMapping(value = "/user/signup", method = RequestMethod.GET, produces = MediaType.TEXT_PLAIN_VALUE)
    protected @ResponseBody String usercheck(HttpServletRequest request) throws Exception {
        Person user =  userDao.getnew(request.getParameter("aid1"));
        if (user != null) {
            return "Email Already Registered";
        } else {
            return "";
        }
    }



    // User signup - Post
    @RequestMapping(value = "/user/signup", method = RequestMethod.POST)
    protected ModelAndView signUpUser(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        try {
            Person puser =  userDao.getnew(request.getParameter("emailId"));
            if (puser==null){
            Enduser user = euserDao.signup(request.getParameter("firstName"), request.getParameter("lastName"),
                    request.getParameter("age"), request.getParameter("emailId"), request.getParameter("contactNo"),
                    request.getParameter("userName"), request.getParameter("password"),
                    request.getParameter("loginType"));
            if (user == null) {
                System.out.println("Issue While Creating User at this point .Try again Later");
                session.setAttribute("errorMessage", "Issue While Creating User at this point .Try again Later");
                return  new ModelAndView("error");
            }
            List<Sports> spList = sportsDAO.list();
            session.setAttribute("ltype", request.getParameter("loginType"));
            session.setAttribute("Enduser", user);
            return new ModelAndView("userPage", "spList", spList);
            }
            else {
                session.setAttribute("errorMessage", "Email Already Registered");
                return  new ModelAndView("error");

            }

        } catch (UserException e) {
            System.out.println("Exception: " + e.getMessage());
            session.setAttribute("errorMessage", "error while SignUp");
            return  new ModelAndView("error");
        }

    }

}
```