

Northeastern University Mail Services

INFO – 6250 Web Development & Tools

Final Project | Palak Agarwal - 001090330

Summary

Mail and Package Services is the source for all incoming and outgoing mail. University Mail Services is the central mailing center for the University of Northeastern and functions as the liaison with the United States Postal Service by delivering and collecting first-class and bulk mailings. Mail Services uses a tracking system to scan in and track most packages for students. Students receive notification via the web portal of any packages available for pickup. Large packages received via the U.S. Mail (U.S.P.S.) are scanned in with a notification sent to students.

Identified Problems with the current system:

- Students must wait a lot in line to get their packages.
- Even though notification was received, it is not uncommon for students to visit a couple of times before their packages are found by staff.
- There is always a lack of space to keep all the packages and therefore intelligently receive, store, and distribute packages process is needed.
- Proper tracking of incoming and outgoing packages is needed.

User Personas and their functionalities

1. Facilities Coordinator
 - a. Create all users and roles and notify them by emailing their passwords.
 - b. View all users and their details and edit them if needed.
 - c. Monitor mailroom system analyzing the number of packages, aisles, and incoming/outgoing status
2. Mailroom Clerk
 - a. View incoming packages and make entry into the system as per student name
 - b. Assign aisles and shelf numbers where packages will be placed
 - c. Notify students of their packages ready for pickup by emailing randomly generated OTP.
3. Mail Associate
 - a. Verify the student details when a student comes to pick up their packages

- b. Deliver the package to students and update the system
- 4. Student
 - a. Students create their accounts by registering on the portal to view any incoming packages.
 - b. Receive notification emails to pick up their packages
 - c. Check package information using login details on the mailroom website
 - d. Pickup packages by showing their university id card and randomly generating OTP.

Technology Stack

Client-Side: HTML/JSP, CSS, Bootstrap, JavaScript, jQuery

Server Side: Spring MVC and Hibernate using annotations, Criteria Queries

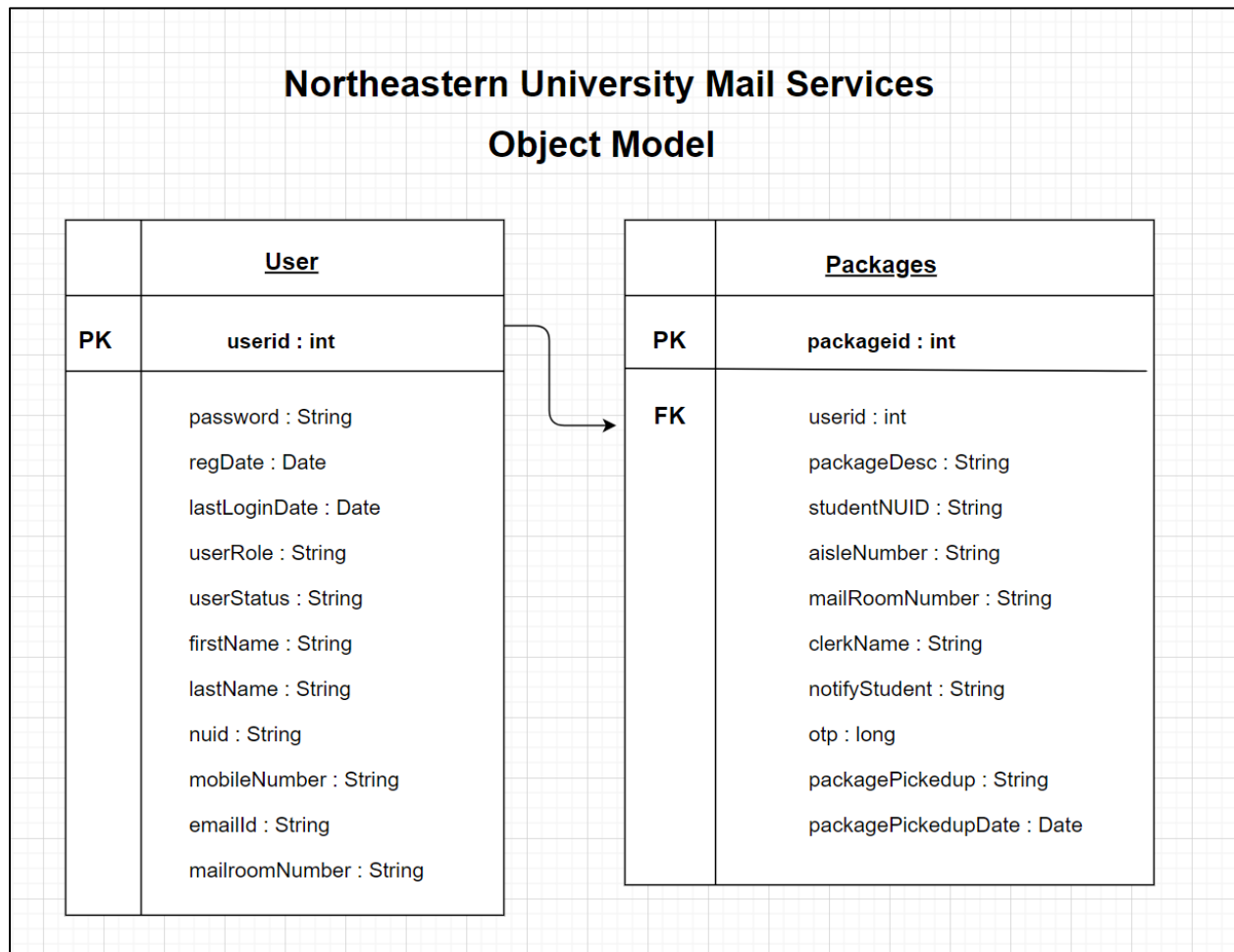
Database: MySQL

Server: Apache Tomcat

Course Material Concepts used:


- Hibernate
- Spring MVC Framework
- Criteria Queries
- JQuery
- Many-to-one mapping
- Custom Tags
- Log4j logging
- Sending Email
- Simple Form Controller
- Form Submission and JSPs
- Native SQL Queries
- HQL Query and Filtering
- JavaBeans
- Request Dispatcher
- Session Management
- Unidirectional Association with Join Tables
- Bootstrap and CSS for front-end


Object Model



Screenshots

Welcome Page

 Northeastern University Mail Services



University Mail Services

Welcome to Mail Services

Mail and Package Services is the source for all incoming and outgoing mail. University Mail Services is the central mailing center for the University of Northeastern and functions as the liaison with the United States Postal Service by both delivering and collecting first class and bulk mailings. Mail Services uses a tracking system to scan in and track most packages for students. Students receive notification via the web portal of any packages available for pick-up. Large packages received via the U.S. Mail (U.S.P.S.) are scanned in with a notification sent to students.

Shipping Services

Shipping Services Mail Services provides shipping services for all Northeastern faculty, staff and students! We sell postage stamps (singles, books, or rolls) and shipping supplies including a variety of boxes, packaging tape, bubble wrap and more - everything to meet your shipping needs. Ship your packages through us via the USPS, FedEx or UPS.

Come visit us

Speare Mailroom

Package Pick-up Hours

- Monday-Friday: 8AM - 8PM
- Saturday: 10AM - 4PM
- Sunday: CLOSED

Residential Mail

- 7 Speare Pl.
- 617-373-2529
- ResidentialMail@northeastern.edu

Columbus Mailroom

Package Pick-up Hours

- Monday-Friday: 8AM - 8PM
- Saturday: 10AM - 4PM
- Sunday: CLOSED

Residential Mail

- 716 Columbus Ave.
- 617-373-5178
- ResidentialMail@northeastern.edu

Mail & Package Delivery

Mail & Package Delivery Mail and/or packages are delivered daily by the USPS, FedEx, UPS and DHL as well as Amazon Direct. While your carrier may have sent you an e-mail stating your package has been delivered to Northeastern, Mail Services needs time to sort and process those packages for delivery to you. Package processing time may vary depending on incoming volume. However, we make every effort to have all packages processed and available for pick-up within 24 hours of receipt.

Northeastern University

[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)

Quick Links

[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)





Login to Portal

Login to access your packages

Login

Contact Us

ResidentialMail@northeastern.edu

Copyright Northeastern University © 2021. All rights reserved.

Login Page

Northeastern University Mail Services

Home

Explore Northeastern

Login to Portal

Login

Access the Northeastern University Mail Services Portal!

Enter Email id

Enter Password

Confirm

Register new account

Northeastern University

Quick Links

Emergency Information

Privacy Policy

Shipping Store

Login to Portal

Login to access your packages


Login

Contact Us

ResidentialMail@northeastern.edu

Copyright Northeastern University © 2021. All rights reserved.

Login Page Exception Handling

 Northeastern University Mail Services

Login Failed.! Please try Again.

Login

Access the Northeastern University Mail Services Portal!

Enter Email id

Enter Password

Confirm

[Register new account](#)

Northeastern University

Quick Links

Emergency Information

Privacy Policy

Shipping Store

Login to Portal

Login to access your packages


Login

Contact Us

ResidentialMail@northeastern.edu

Copyright Northeastern University © 2021. All rights reserved.

Facilities Coordinator Home Page

 Northeastern University Mail Services


Home

Create EmployeesMail Room CRMEdit Your Profile

Search

Logout

Facilities Coordinator Home Page



Northeastern University

Quick Links

[Emergency Information](#)[Privacy Policy](#)[Shipping Store](#)





Login to Portal

Login to access your packages

Login


Contact Us

ResidentialMail@northeastern.edu



Copyright Northeastern University © 2021. All rights reserved.

Register New Employee

 Northeastern University Mail Services

Home

Create EmployeesMail Room CRMEdit Your Profile

Search

Logout

Register New Employee

User Role

MailAssociate

First Name

Last Name

E-Mail Address

Mail Room Number

Password

Enter Password

Confirm Password

Confirm Password

Phone Number

Register

Northeastern University

Quick Links

[Emergency Information](#)[Privacy Policy](#)[Shipping Store](#)

Login to Portal

Login to access your packages

Login

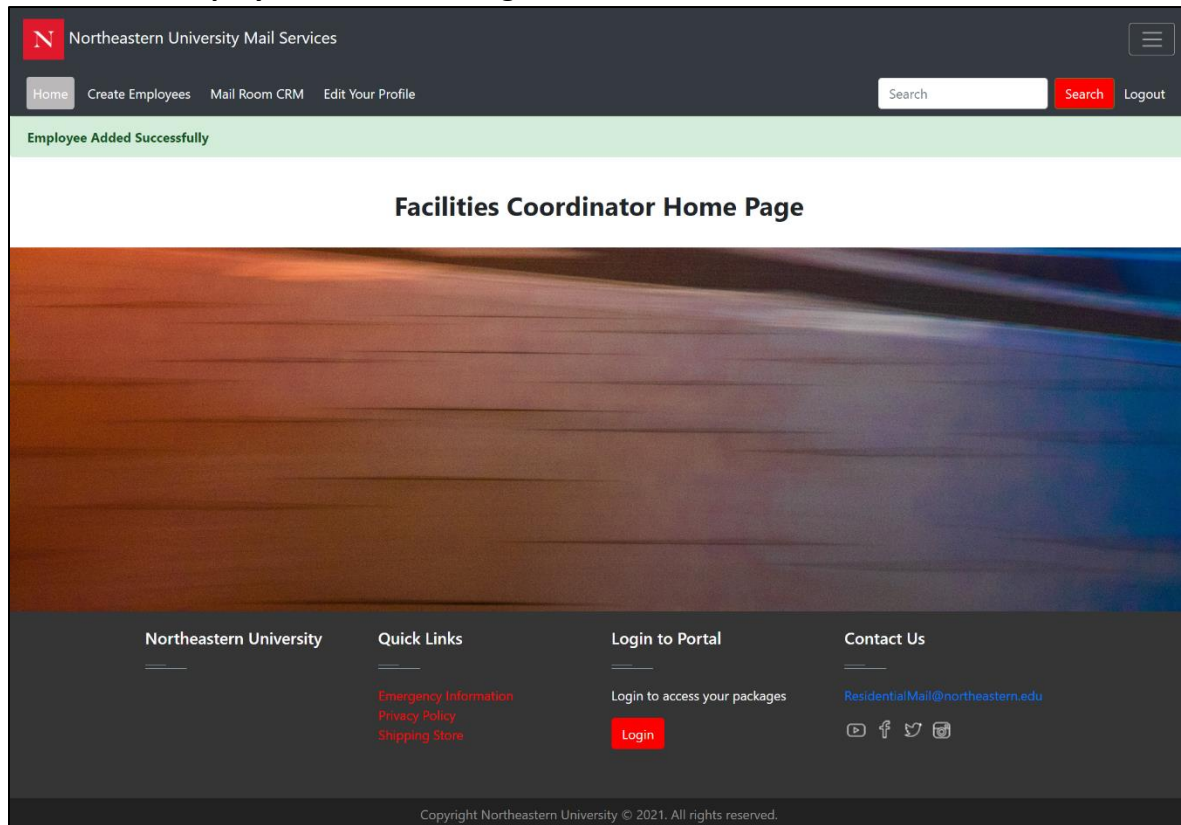
Contact Us

ResidentialMail@northeastern.edu



Copyright Northeastern University © 2021. All rights reserved.

Create New Employee Success Message




Facilities Coordinator Edit Profile

The screenshot shows the 'Edit Personal Details' form within the Northeastern University Mail Services portal. The form is titled 'Edit Personal Details' and contains the following fields and options:

- E-Mail Address: facilitiescoordinator@gmail.com
- First Name: Facilities
- Last Name: Coordinator
- Phone Number: 000000000
- Do you wish to change your password?: ☒
- New Password: Enter Password
- Confirm Password: Confirm Password
- Save Details (button)

User Account Updated Message

 Northeastern University Mail Services

[Home](#) [Create Employees](#) [Mail Room CRM](#) [Edit Your Profile](#)

[Search](#) [Logout](#)

User Account Updated Successfully

Facilities Coordinator Home Page



[Northeastern University](#)


[Quick Links](#)
[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)

[Login to Portal](#)
Login to access your packages
[Login](#)

[Contact Us](#)
ResidentialMail@northeastern.edu
   

Copyright Northeastern University © 2021. All rights reserved.

Mail Room Clerk Home Page

 Northeastern University Mail Services

[Home](#) [Add New Packages](#) [Edit Your Profile](#)

[Search](#) [Logout](#)

Mail Room Clerk Home Page



[Northeastern University](#)


[Quick Links](#)
[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)

[Login to Portal](#)
Login to access your packages
[Login](#)

[Contact Us](#)
ResidentialMail@northeastern.edu
   

Copyright Northeastern University © 2021. All rights reserved.

Add New Packages

 Northeastern University Mail Services

Home

Add New Packages

Edit Your Profile

Search

Logout

Add New Packages

Package Description

Aisle Number

Mail Room Number

NEU ID

Add Package Details

Northeastern University

Quick Links

Emergency Information

Privacy Policy

Shipping Store





Login to Portal

Login to access your packages

Login


Contact Us

ResidentialMail@northeastern.edu



Copyright Northeastern University © 2021. All rights reserved.

Student Registration Page

 Northeastern University Mail Services

Register

First Name

Last Name

E-Mail Address

Password

Confirm Password

Phone Number

NEU ID

Register

Northeastern University

Quick Links

Emergency Information

Privacy Policy

Shipping Store





Login to Portal

Login to access your packages

Login


Contact Us

ResidentialMail@northeastern.edu



Copyright Northeastern University © 2021. All rights reserved.

Student Home Page

 Northeastern University Mail Services

Home

Edit Profile

Search

Search

Logout

Mail Room CRM

Search...

#	Package Description ↕	Mail Room Number ↕	OTP
1	Amazon Package #0022	1	7474

Northeastern University

Quick Links





[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)

Login to Portal

Login to access your packages


Login

Contact Us

ResidentialMail@northeastern.edu
   

Copyright Northeastern University © 2021. All rights reserved.

Student Edit Profile

 Northeastern University Mail Services

Home

Edit Profile

Search

Search

Logout

Edit Personal Details

E-Mail Address

palakagarwalo88@gmail.com

First Name

Palak

Last Name

Agarwal

Phone Number

6092737093

Do you wish to change your password?

☒

New Password


Enter Password

Confirm Password

Confirm Password

Save Details

Mail Associate Home Page

 Northeastern University Mail Services

Home Edit Your Profile

Search Search Logout

Mail Associate Home Page

Find Packages

Enter OTP





View Ready to Deliver Packages

Northeastern University

Quick Links
[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)


Login to Portal
Login to access your packages

Login

Contact Us
ResidentialMail@northeastern.edu
   

Copyright Northeastern University © 2021. All rights reserved.

Student Package View Page

 Northeastern University Mail Services

Home

Search Search Logout

Mail Room CRM

Search...





#	Package Description ↕	Student NEU Id ↕	Mail Room Number ↕	Aisle Number ↕	Package Delivered
1	Amazon Package #0022	001090330	1	83	<div>Deliver Package</div>

Northeastern University

Quick Links
[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)


Login to Portal
Login to access your packages

Login

Contact Us
ResidentialMail@northeastern.edu
   

Copyright Northeastern University © 2021. All rights reserved.

Package Delivered Success Message

 Northeastern University Mail Services

[Home](#) [Edit Your Profile](#)

[Search](#) [Logout](#)

Package Details Updated Successfully

Mail Associate Home Page

Find Packages

[View Ready to Deliver Packages](#)

Northeastern University


Quick Links
[Emergency Information](#)
[Privacy Policy](#)
[Shipping Store](#)

Login to Portal
Login to access your packages
[Login](#)

Contact Us
ResidentialMail@northeastern.edu
[YouTube](#) [Facebook](#) [Twitter](#) [Instagram](#)

Copyright Northeastern University © 2021. All rights reserved.

Mail Room CRM

 Northeastern University Mail Services

[Home](#) [Create Employees](#) [Mail Room CRM](#) [Edit Your Profile](#)

[Search](#) [Logout](#)

Mail Room CRM

#	Package Description ↕	Student NEU Id ↕	Mail Room Number ↕	Aisle Number ↕	Clerk Name ↕	Was Student Notified ↕	Package Picked Up	Package Picked Date
1	Amazon Package #0022	001090330	1	B3	MailRoom	Yes	Yes	2021-12-17

Controller Code

1. Home Controller

```
package com.mvc.universitymailservices;

import javax.servlet.http.HttpSession;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.mvc.dao.UserDAO;
import com.mvc.pojo.user;

/**
 * Handles requests for the application home page.
 */
@Controller
public class HomeController {

    private PasswordEncoder passwordEncoder = new BCryptPasswordEncoder(10);
    private final Logger logger = LoggerFactory.getLogger(HomeController.class);

    // HomePage
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public ModelAndView goHome(UserDAO userdao) {

        user user = userdao.getUserByEmail("facilitiescoordinator@gmail.com");
        if(user==null)
        {
            String password = "Password";
            String hashedPassword = passwordEncoder.encode(password);
            userdao.addAdminUser(hashedPassword);
        }
    }
}
```

```

        ModelAndView model = new ModelAndView("home");
        return model;
    }

    // HomePage
    @RequestMapping(value = "/home", method = RequestMethod.GET)
    public ModelAndView getHomePage(HttpSession session) {
        user user = (user) session.getAttribute("user");
        if (user != null) {
            ModelAndView model = new ModelAndView("home");
            model.addObject("user", user);
            return model;
        } else {
            ModelAndView model = new ModelAndView("redirect:/");
            return model;
        }
    }

    // GET Login Page
    @RequestMapping(value = "/login", method = RequestMethod.GET)
    public ModelAndView getLogin(ModelMap model) {
        user user = new user(); // FormBackingObject
        model.addAttribute("user", user);
        ModelAndView mv = new ModelAndView("login");
        return mv;
    }

    // GET Sign Up Page Request
    @RequestMapping(value = "/register", method = RequestMethod.GET)
    public ModelAndView getRegister(ModelMap model, @ModelAttribute("user")
user user) {
        model.addAttribute("user", user);
        ModelAndView mv = new ModelAndView("register");
        return mv;
    }
}

```

2. User Controller

```

package com.mvc.universitymailservices;

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

```

```

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.HtmlEmail;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.support.SessionStatus;
import org.springframework.web.servlet.ModelAndView;

import com.mvc.dao.UserDAO;
import com.mvc.pojo.user;

@Controller
public class UserController {

    private final Logger logger = LoggerFactory.getLogger(UserController.class);
    private static BCryptPasswordEncoder passwordEncoder = new
BCryptPasswordEncoder();

    // Login POST Request
    @RequestMapping(value = "/submitLogin", method = RequestMethod.POST)
    public ModelAndView loginProcess(HttpServletRequest request, UserDAO
userdao, HttpSession session, ModelMap model,
        SessionStatus status) {

        String emailId = request.getParameter("emailId");
        String password = request.getParameter("password");

        user user = userdao.getUserByEmail(emailId);
        if (user == null) {

            String errormsg = "Please Enter Valid Email Address";
            ModelAndView mv = new ModelAndView("login", "errormsg",
errormsg);

            return mv;
        } else {
            String role = user.getUserRole();
            String userstatus = user.getUserStatus();

```



```

        Boolean checkPassword = passwordEncoder.matches(password,
user.getPassword());

        if (checkPassword) {

            if (role.equals("FacilitiesCoordinator")) {

                if (userstatus.equals("Active")) {

                    userDao.updateLogin(user);
                    session.setAttribute("facilitiesCoordinator",
user);

                    ModelAndView mv = new
ModelAndView("redirect:/facilitiesCoordinatorHome");
                    return mv;

                } else {

                    String errormsg = "Account is InActive.
Contact ResidentialMail@northeastern.edu";
                    ModelAndView mv = new
ModelAndView("login", "errormsg", errormsg);
                    return mv;

                }

            } else if (role.equals("MailAssociate")) {

                if (userstatus.equals("Active")) {

                    userDao.updateLogin(user);
                    session.setAttribute("mailAssociate", user);

                    ModelAndView mv = new
ModelAndView("redirect:/mailAssociateHome");
                    return mv;

                } else {

                    String errormsg = "Account is InActive.
Contact ResidentialMail@northeastern.edu";

```

```

        ModelAndView mv = new
ModelAndView("login", "errmsg", errmsg);
        return mv;
    }

    } else if (role.equals("Student")) {

        if (userstatus.equals("Active")) {

            userdao.updateLogin(user);
            session.setAttribute("student", user);
            ModelAndView mv = new
ModelAndView("redirect:/studentHome");
            return mv;

        } else {

            String errmsg = "Account is InActive.
Contact ResidentialMail@northeastern.edu";
            ModelAndView mv = new
ModelAndView("login", "errmsg", errmsg);
            return mv;

        }

    } else {

        if (userstatus.equals("Active")) {

            userdao.updateLogin(user);
            session.setAttribute("mailroomClerk",
user);

            ModelAndView mv = new
ModelAndView("redirect:/mailroomClerkHome");
            return mv;

        } else {

            String errmsg = "Account is InActive.
Contact ResidentialMail@northeastern.edu";
            ModelAndView mv = new
ModelAndView("login", "errmsg", errmsg);

```

```

        return mv;
    }
}

} else {

    String errmsg = "Login Failed.! Please try Again.";
    ModelAndView mv = new ModelAndView("login",
"errmsg", errmsg);

    return mv;

}

}

}

// Register POST Request
@RequestMapping(value = "/register", method = RequestMethod.POST)
public ModelAndView registrationProcess(HttpSession session,
HttpServletRequest request, UserDao userDao,
BindingResult result, ModelMap model, SessionStatus status) {

    user user = (user) session.getAttribute("user");

    String password = request.getParameter("password");
    String emailId = request.getParameter("email_address");
    String firstName = request.getParameter("firstName");
    String lastName = request.getParameter("lastName");
    String mobileNumber = request.getParameter("phonenummer");
    String nuid = request.getParameter("neuid");

    String regex = "^[\\w-_.+]*[\\w-_.]\\@([\\w]+\\.)+[\\w]+[\\w]$$";
    boolean validEmail = emailId.matches(regex);

    if (!validEmail) {
        result.rejectValue("password", "user.emailId.invalid", "Please
enter valid Email Id!");
    }

    if (result.hasErrors()) {

        ModelAndView m = new ModelAndView("register");
        return m;
    }
}

```

```

        boolean bEmailExist = userDao.doUserExists(emailId);

        if (!bEmailExist) {
            status.setComplete();

            String errormsg = "User Already Exists.!!";
            ModelAndView m = new ModelAndView("register", "errormsg",
errormsg);
            return m;
        } else {

            String userrole = "Student";
            String hashedPassword = passwordEncoder.encode(password);

            userDao.addUser(emailId, hashedPassword, userrole, firstName,
lastName, mobileNumber, null, nuid);
            status.setComplete();

            String msg = "User Account Created Successfully";

            Email email = new HtmlEmail();

            email.setHostName("smtp.gmail.com");
            email.setSmtpPort(465);
            email.setAuthenticator(new
DefaultAuthenticator("residentialmailneu@gmail.com", "Password@123"));
            email.setSSLOnConnect(true);

            try {

                email.setFrom("residentialmailneu@gmail.com");
                email.setSubject("NeuMailServices :: Your Account has
been Created");
                email.setMsg(" <h2>Account Details </h2>" + "<br>" +
"<h3>Email ID: " + emailId + "</h3>"
                                + "<h3>Password: " + password + "</h3>" +
"<span style:'color:red;'>");
                email.addTo(emailId);
                email.send();

            } catch (EmailException e) {

                e.printStackTrace();
            }
        }
    }
}

```

```

        }

        ModelAndView m = new ModelAndView("login", "msg", msg);
        return m;
    }

}

// Logout
@RequestMapping("/userLogout")
public ModelAndView userLogout(HttpSession session) {
    session.invalidate();
    ModelAndView m = new ModelAndView("redirect:/");
    return m;
}

}

```

3. Facilities Coordinator Controller

```

package com.mvc.universitymailservices;

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

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

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.HtmlEmail;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
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.bind.support.SessionStatus;
import org.springframework.web.servlet.ModelAndView;

```

```

import com.mvc.dao.FacilitiesCoordinatorDAO;
import com.mvc.dao.UserDAO;
import com.mvc.pojo.FacilitiesCoordinator;
import com.mvc.pojo.user;

/**
 * Handles requests for the application home page.
 */
@Controller
public class FacilitiesCoordinatorController {

    private static BCryptPasswordEncoder passwordEncoder = new
BCryptPasswordEncoder();
    private final Logger logger =
LoggerFactory.getLogger(FacilitiesCoordinatorController.class);

    // FacilitiesCoordinator Dashboard
    @RequestMapping(value = "/facilitiesCoordinatorHome", method =
RequestMethod.GET)
    @ResponseBody
    public ModelAndView getFacilitiesCoordinatorHomePage(HttpSession session,
Map<String, Object> model,
FacilitiesCoordinatorDAO fCdao) {

        user user = (user) session.getAttribute("facilitiesCoordinator");
        FacilitiesCoordinator facilitiesCoordinator =
fCdao.getUserById(user.getUserid());

        ModelAndView m = new ModelAndView("facilitiesCoordinatorHome");
        model.put("facilitiesCoordinator", facilitiesCoordinator);
        model.put("user", user);
        return m;
    }

    // Add User GET Request
    @RequestMapping(value = "/facilitiesCoordinatorHome/createEmployees",
method = RequestMethod.GET)
    public ModelAndView viewAddUser(HttpSession session, Map<String, Object>
model, FacilitiesCoordinatorDAO fCdao) {

        user user = (user) session.getAttribute("facilitiesCoordinator");
        FacilitiesCoordinator facilitiesCoordinator =
fCdao.getUserById(user.getUserid());

```

```

        List<String> userrole = new ArrayList<String>();
        userrole.add("MailAssociate");
        userrole.add("MailRoomClerk");

        model.put("userrole", userrole);
        model.put("facilitiesCoordinator", facilitiesCoordinator);
        model.put("user", user);
        ModelAndView m = new ModelAndView("createEmployees");

        return m;
    }

    @RequestMapping(value = "/facilitiesCoordinatorHome/createEmployees",
        method = RequestMethod.POST)
    public ModelAndView AddUser(HttpSession session, HttpServletRequest
        request, UserDao userDao,
        BindingResult result, ModelMap model, SessionStatus status) {

        String password = request.getParameter("password");
        String emailId = request.getParameter("email_address");
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
        String phonenumber = request.getParameter("onenumber");
        String mailroomnumber = request.getParameter("mailroomnumber");
        user user = (user) session.getAttribute("facilitiesCoordinator");

        String regex = "^[\\w-_.+]*[\\w-_.]@[([\\w]+\\.)+[\\w]+[\\w]$";
        boolean validEmail = emailId.matches(regex);

        if (!validEmail) {
            result.rejectValue("password", "user.emailId.invalid", "Please
enter valid Email Id!");
        }

        if (result.hasErrors()) {
            String errormsg = "Please enter valid Email Id!";
            List<String> userrole = new ArrayList<String>();
            userrole.add("MailAssociate");
            userrole.add("MailRoomClerk");
            model.put("userrole", userrole);
            ModelAndView m = new ModelAndView("createEmployees",
"errormsg", errormsg);
            return m;
        }
    }

```



```

    }

    String userrole = request.getParameter("userrole");
    String hashedPassword = passwordEncoder.encode(password);
    boolean bEmailExist = userDao.doUserExists(emailId);

    if (!bEmailExist) {

        status.setComplete();

        String errormsg = "Employee Already Exists.!!";

        List<user> list = userDao.getAllUsers();
        model.put("users", list);

        List<String> userRole = new ArrayList<String>();
        userRole.add("MailAssociate");
        userRole.add("MailRoomClerk");
        model.put("userrole", userRole);

        ModelAndView m = new ModelAndView("createEmployees",
"errormsg", errormsg);
        return m;

    } else {

        userDao.addUser(emailId, hashedPassword, userrole, firstName,
lastName, phonenumber, mailroomnumber,null);

        status.setComplete();

        String msg = "Employee Added Successfully";

        List<user> list = userDao.getAllUsers();
        model.put("users", list);

        Email email = new HtmlEmail();

        email.setHostName("smtp.gmail.com");
        email.setSmtpPort(587);
        email.setAuthenticator(new
DefaultAuthenticator("residentialmailneu@gmail.com", "Password@123"));
        email.setSSLonConnect(true);

```

```

        try {

            email.setFrom("residentialmailneu@gmail.com");
            email.setSubject("NeuMailServices :: Your Account has
been Created");

            email.setMsg(" <h2>Account Details </h2>" + "<br>" +
"<h3>Email ID: " + emailId + "</h3>"
                        + "<h3>Password: " + password + "</h3>" +
"<span style:'color:red;'>");

            email.addTo(emailId);
            email.send();

        } catch (EmailException e) {

            e.printStackTrace();

        }

        ModelAndView m = new
ModelAndView("facilitiesCoordinatorHome", "msg", msg);
        return m;
    }

    // GET Edit Profile
    @RequestMapping(value =
"facilitiesCoordinatorHome/facilitiesCoordinatorEditProfile", method =
RequestMethod.GET, headers = "Accept=application/json")
    public ModelAndView getUserHistory(HttpSession session, Map<String, Object>
model, FacilitiesCoordinatorDAO fCdao) {

        user user = (user) session.getAttribute("facilitiesCoordinator");
        //FacilitiesCoordinator facilitiesCoordinator =
fCdao.getUserById(user.getUserid());

        model.put("facilitiesCoordinator", user);
        model.put("user", user);
        ModelAndView m = new
ModelAndView("facilitiesCoordinatorEditProfile");
        return m;
    }

    // Update User Profile

```

```

        @RequestMapping(value =
"facilitiesCoordinatorHome/facilitiesCoordinatorEditProfile", method =
RequestMethod.POST)
        public ModelAndView updateUser(HttpSession session, HttpServletRequest
request, ModelMap model,
                SessionStatus status, FacilitiesCoordinatorDAO fCdao) {

                String password = request.getParameter("password");
                String firstName = request.getParameter("firstName");
                String lastName = request.getParameter("lastName");
                String phonenumber = request.getParameter("onenumber");

                user user = (user) session.getAttribute("facilitiesCoordinator");

                //FacilitiesCoordinator facilitiesCoordinator =
fCdao.getUserById(u.getUserid());

                if (password != null) {
                        String hashedPassword = passwordEncoder.encode(password);
                        fCdao.updateDetails(hashedPassword, firstName, lastName,
phonenumber, user);
                        status.setComplete();
                } else {
                        fCdao.updateDetails(firstName, lastName, phonenumber, user);
                        status.setComplete();
                }

                String msg = "User Account Updated Successfully";
                ModelAndView m = new ModelAndView("facilitiesCoordinatorHome",
"msg", msg);
                return m;
        }
}

```

4. Mailroom Associate Controller

```

package com.mvc.universitymailservices;

import java.util.List;
import java.util.Map;

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

```

```

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.PathVariable;
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.bind.support.SessionStatus;
import org.springframework.web.servlet.ModelAndView;

import com.mvc.dao.MailAssociateDAO;
import com.mvc.dao.PackagesDAO;
import com.mvc.pojo.MailAssociate;
import com.mvc.pojo.packages;
import com.mvc.pojo.user;

/**
 * Handles requests for the application home page.
 */
@Controller
public class MailroomAssociateController {

    private static BCryptPasswordEncoder passwordEncoder = new
BCryptPasswordEncoder();
    private final Logger logger =
LoggerFactory.getLogger(MailroomAssociateController.class);

    // FacilitiesCoordinator Dashboard
    @RequestMapping(value = "/mailAssociateHome", method =
RequestMethod.GET)
    @ResponseBody
    public ModelAndView getFacilitiesCoordinatorHomePage(HttpSession session,
Map<String, Object> model,
MailAssociateDAO maildao) {

        user user = (user) session.getAttribute("mailAssociate");
        MailAssociate mailAssociate = maildao.getUserById(user.getUserid());

        ModelAndView m = new ModelAndView("mailAssociateHome");
        model.put("mailAssociate", mailAssociate);
        model.put("user", user);
        return m;
    }
}

```

```

    }

    @RequestMapping(value = "/mailAssociateHome", method =
RequestMethod.POST)
    public ModelAndView viewStudentPackages(HttpServletRequest request,
Map<String, Object> model,
        PackagesDAO packagesdao, MailAssociateDAO maildao,
HttpSession session) {

        user user = (user) session.getAttribute("mailAssociate");
        String otp = request.getParameter("otp");
        List<packages> packages = PackagesDAO.getAllPackagesbyOtp(otp);

        ModelAndView m = new ModelAndView("studentPackagesView",
"packages", packages);
        model.put("mailAssociate", user);
        model.put("otp", otp);
        return m;
    }

    @RequestMapping(value = "/DeliveredStudentPackages/{packageId}/{otp}",
method = RequestMethod.GET)
    public ModelAndView attendOpdPatient(@PathVariable int packageId,
@PathVariable String otp, HttpSession session,
        HttpServletRequest request, Map<String, Object> model,
PackagesDAO packagesdao, MailAssociateDAO maildao) {

        // int packageId = Integer.parseInt(request.getParameter("packageId"));
        user user = (user) session.getAttribute("mailAssociate");
        MailAssociate mailAssociate = maildao.getUserById(user.getUserid());

        packagesdao.updatePackageDelivery(packageId);
        String msg = "Package Details Updated Successfully";

        ModelAndView m = new ModelAndView("mailAssociateHome");
        model.put("msg", msg);
        model.put("mailAssociate", mailAssociate);
        model.put("user", user);
        return m;
    }

    // GET Edit Profile
    @RequestMapping(value = "mailAssociateHome/mailAssociateEditProfile",
method = RequestMethod.GET, headers = "Accept=application/json")

```

```

        public ModelAndView getUserHistory(HttpSession session, Map<String, Object>
model, MailAssociateDAO maildao) {

            user user = (user) session.getAttribute("mailAssociate");
            //MailAssociate mailAssociate = maildao.getUserById(user.getUserid());

            ModelAndView m = new ModelAndView("mailAssociateEditProfile");
            model.put("mailAssociate", user);
            model.put("user", user);
            return m;
        }

        // Update User Profile
        @RequestMapping(value = "mailAssociateHome/mailAssociateEditProfile",
method = RequestMethod.POST)
        public ModelAndView updateUser(HttpSession session,HttpServletRequest
request, ModelMap model, MailAssociateDAO maildao,
            SessionStatus status) {

            String password = request.getParameter("password");
            String firstName = request.getParameter("firstName");
            String lastName = request.getParameter("lastName");
            String phonenumber = request.getParameter("onenumber");

            user user = (user) session.getAttribute("mailAssociate");

            //MailAssociate mailAssociate = maildao.getUserById(u.getUserid());

            if (password != null) {
                String hashedPassword = passwordEncoder.encode(password);
                maildao.updateDetails(hashedPassword, firstName, lastName,
phonenumber, user);
                status.setComplete();
            } else {
                maildao.updateDetails(firstName, lastName, phonenumber, user);
                status.setComplete();
            }

            String msg = "User Account Updated Successfully";
            ModelAndView m = new ModelAndView("mailAssociateHome", "msg",
msg);
            return m;
        }
    }
}

```

5. Mailroom Clerk Controller

```
package com.mvc.universitymailservices;

import java.util.Map;
import java.util.Random;

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

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.HtmlEmail;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
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.bind.support.SessionStatus;
import org.springframework.web.servlet.ModelAndView;

import com.mvc.dao.MailClerkDAO;
import com.mvc.dao.PackagesDAO;
import com.mvc.dao.UserDAO;
import com.mvc.pojo.MailroomClerk;
import com.mvc.pojo.user;

/**
 * Handles requests for the application home page.
 */
@Controller
public class MailroomClerkController {

    private static BCryptPasswordEncoder passwordEncoder = new
    BCryptPasswordEncoder();
    private final Logger logger =
    LoggerFactory.getLogger(MailroomClerkController.class);

    // FacilitiesCoordinator Dashboard
```



```

        @RequestMapping(value = "/mailroomClerkHome", method =
RequestMethod.GET)
        @ResponseBody
        public ModelAndView getFacilitiesCoordinatorHomePage(HttpSession session,
Map<String, Object> model,
MailClerkDAO mailclerkdao) {

            user user = (user) session.getAttribute("mailroomClerk");
            MailroomClerk mailroomClerk =
mailclerkdao.getUserById(user.getUserid());

            ModelAndView m = new ModelAndView("mailroomClerkHome");
            model.put("mailroomClerk", mailroomClerk);
            model.put("user", user);
            return m;
        }

        // Add User GET Request
        @RequestMapping(value = "/mailroomClerkHome/addNewPackages", method =
RequestMethod.GET)
        public ModelAndView viewAddUser(HttpSession session, Map<String, Object>
model, MailClerkDAO mailclerkdao) {

            user user = (user) session.getAttribute("mailroomClerk");
            // MailroomClerk mailroomClerk =
mailclerkdao.getUserById(user.getUserid());

            model.put("mailroomClerk", user);
            model.put("user", user);

            ModelAndView m = new ModelAndView("addNewPackages");
            return m;
        }

        @RequestMapping(value = "/mailroomClerkHome/addNewPackages", method =
RequestMethod.POST)
        public ModelAndView AddPackage(HttpSession session, HttpServletRequest
request, UserDao userDao,
PackagesDAO packagesdao, BindingResult result, ModelMap
model, SessionStatus status,
MailClerkDAO mailclerkdao) {

            String packageDesc = request.getParameter("packageDesc");
            String aisleNumber = request.getParameter("aisleNumber");

```

```

String mailRoomNumber = request.getParameter("mailRoomNumber");
String nuid = request.getParameter("neuid");

user user = (user) session.getAttribute("mailroomClerk");

String notifyStudent = "Yes";
String clerkName = "";

Random randomGenerator = new Random();
long otp = randomGenerator.nextInt(10000);

if ((user.getFirstName()) == null) {
    clerkName = "Mail Associate";
} else {
    clerkName = user.getFirstName();
}

packagesdao.addPackage(packageDesc, aisleNumber, mailRoomNumber,
clerkName, nuid, notifyStudent, otp, user);
mailclerkdao.updatepackageUserId(nuid);

status.setComplete();

String msg = "Package Added Successfully";

if (notifyStudent.equals("Yes")) {
    Email email = new HtmlEmail();

    email.setHostName("smtp.gmail.com");
    email.setSmtpPort(465);
    email.setAuthenticator(new
DefaultAuthenticator("residentialmailneu@gmail.com", "Password@123"));
    email.setSSLOnConnect(true);

    try {

        email.setFrom("residentialmailneu@gmail.com");
        email.setSubject("NeuMailServices :: Your Package is ready
for pickup");
        email.setMsg(" <h2>Account Details </h2>" + "<br>" +
"<h3>Your OTP to pickup packages are: " + otp
+ "</h3>" + "<span style:'color:red;'>");
        email.addTo("residentialmailneu@gmail.com");
        email.send();
    }
}

```

```

        } catch (EmailException e) {

            e.printStackTrace();
        }
    }

    ModelAndView m = new ModelAndView("mailroomClerkHome", "msg",
msg);
    return m;
}

// GET Edit Profile
@RequestMapping(value = "mailroomClerkHome/mailClerkEditProfile", method
= RequestMethod.GET, headers = "Accept=application/json")
    public ModelAndView getUserHistory(HttpSession session, Map<String, Object>
model, MailClerkDAO mailclerkdao) {

        user user = (user) session.getAttribute("mailroomClerk");
        MailroomClerk mailroomClerk =
mailclerkdao.getUserById(user.getUserid());

        model.put("mailroomClerk", mailroomClerk);
        model.put("user", user);
        ModelAndView m = new ModelAndView("mailClerkEditProfile");
        return m;
    }

// Update User Profile
@RequestMapping(value = "mailroomClerkHome/mailClerkEditProfile", method
= RequestMethod.POST)
    public ModelAndView updateUser(HttpServletRequest request, ModelMap
model, SessionStatus status,
        MailClerkDAO mailclerkdao, HttpSession session) {

        String password = request.getParameter("password");
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
        String phonenumber = request.getParameter("phonenumber");

        user user = (user) session.getAttribute("mailroomClerk");

        // MailroomClerk mailroomClerk =
mailclerkdao.getUserById(u.getUserid());

```

```

        if (password != null) {
            String hashedPassword = passwordEncoder.encode(password);
            mailclerkdao.updateDetails(hashedPassword, firstName,
lastName, phonenumber, user);
            status.setComplete();
        } else {
            mailclerkdao.updateDetails(firstName, lastName, phonenumber,
user);
            status.setComplete();
        }

        String msg = "User Account Updated Successfully";
        ModelAndView m = new ModelAndView("mailroomClerkHome", "msg",
msg);
        return m;
    }
}

```

6. Package Controller

```

package com.mvc.universitymailservices;

import java.util.List;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.mvc.dao.PackagesDAO;
import com.mvc.pojo.packages;
import com.mvc.pojo.user;

@Controller
public class PackageController {

```

```

        private final Logger logger = LoggerFactory.getLogger(PackageController.class);

        // Add User GET Request
        @RequestMapping(value = "/facilitiesCoordinatorHome/mailroom", method =
RequestMethod.GET)
        public ModelAndView viewMailroom(HttpServletRequest request, Map<String,
Object> model, PackagesDAO packagesdao, @ModelAttribute("user") user user) {

            List<packages> packages = PackagesDAO.getAllPackages();

            ModelAndView m = new ModelAndView("mailroomView", "packages",
packages);
            return m;
        }
    }
}

```

7. Student Controller

```

package com.mvc.universitymailservices;

import java.util.List;
import java.util.Map;

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

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
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.bind.support.SessionStatus;
import org.springframework.web.servlet.ModelAndView;

import com.mvc.dao.PackagesDAO;
import com.mvc.dao.StudentDAO;
import com.mvc.pojo.Student;
import com.mvc.pojo.packages;
import com.mvc.pojo.user;

```

```

/**
 * Handles requests for the application home page.
 */
@Controller
public class StudentController {

    private static BCryptPasswordEncoder passwordEncoder = new
BCryptPasswordEncoder();
    private final Logger logger = LoggerFactory.getLogger(StudentController.class);

    // FacilitiesCoordinator Dashboard
    @RequestMapping(value = "/studentHome", method = RequestMethod.GET)
    @ResponseBody
    public ModelAndView getStudentHomePage(HttpSession session, Map<String,
Object> model,
                                   StudentDAO studentdao, PackagesDAO packagesdao) {

        user user = (user) session.getAttribute("student");
        model.put("student", user);
        Student student = studentdao.getStudentById(user.getUserid());
        List<packages> packages =
PackagesDAO.getAllPackagesbyUserid(user.getUserid());

        ModelAndView m = new ModelAndView("studentHome");
        model.put("packages", packages);
        model.put("user", user);
        return m;
    }

    // GET Edit Profile
    @RequestMapping(value = "/studentHome/studentEditProfile", method =
RequestMethod.GET, headers = "Accept=application/json")
    public ModelAndView getUserHistory(HttpSession session, HttpServletRequest
request, ModelMap model,
                                   StudentDAO studentdao) {

        user user = (user) session.getAttribute("student");
        // Student student = studentdao.getStudentById(user.getUserid());

        model.put("student", user);
        model.put("user", user);
        ModelAndView m = new ModelAndView("studentEditProfile");
        return m;
    }
}

```

```

// Update User Profile
@RequestMapping(value = "/studentHome/studentEditProfile", method =
RequestMethod.POST)
public ModelAndView updateUser(HttpServletRequest request, ModelMap
model, StudentDAO studentdao,
        SessionStatus status, HttpSession session) {

    String password = request.getParameter("password");
    String firstName = request.getParameter("firstName");
    String lastName = request.getParameter("lastName");
    String phonenumber = request.getParameter("phonenumber");

    user user = (user) session.getAttribute("student");

    // Student student = studentdao.getStudentById(u.getUserid());

    if (password != null) {
        String hashedPassword = passwordEncoder.encode(password);
        studentdao.updateStudentDetails(hashedPassword, firstName,
lastName, phonenumber, user);
        status.setComplete();
    } else {
        studentdao.updateStudentDetails(firstName, lastName,
phonenumber, user);
        status.setComplete();
    }

    String msg = "User Account Updated Successfully";
    ModelAndView m = new ModelAndView("studentHome", "msg", msg);
    return m;
}
}

```