Raghavi Kirouchenaradjou - 001826638

Assurance Insurance Group – A dynamic and responsive web application which allows transaction between doctor,patients and the insurance management!

Assurance Insurance Group

Web Tools Project SECTION 1

**Assurance Insurance Group - Project**

## Summary

**Web Application Name: Assurance Insurance Group**

Assurance Insurance Group is a Web application that allows patient, doctor and insurance management admin transactions. It makes life easier for booking an appointment with the Doctor, payment management and other transaction. **It sells traditional and consumer directed health care insurance plans and related services, such as medical**.

## Summary of the Functionalities

It provides the following feature: (More details about the functionality in future pages)

1. Registration – Patient, Doctor
2. Login – Patient, Doctor and Admin
3. Admin Login – Login via Web Security
4. Patient Dashboard
   1. Book an appointment with the Doctor
   2. Patient – Check status of the insurance claimed
   3. Patient – Enroll for the listed Insurance Plan
5. Doctor Dashboard
   1. Submit Patient Case Sheet
   2. See the history - Patients
6. Admin Dashboard
   1. Sanction Approval
   2. Create Insurance Plan
7. Log Off – Patient, Doctor

## Technologies Used

UI – Bootstrap,HTML,CSS

Script – Jquery,Java Script.

Used **Validators** for Registration and Insurance Plan Creation

**PDF Viewer** – Generate Report during Sanction Approval

**AJAX**– Book an Appointment with Doctor, Enroll for the Insurance Plan, Get More Details in Sanction Approval

**HQL**

**Web Security** – Admin Login

Exception Handling – User, Plan, Doctor.

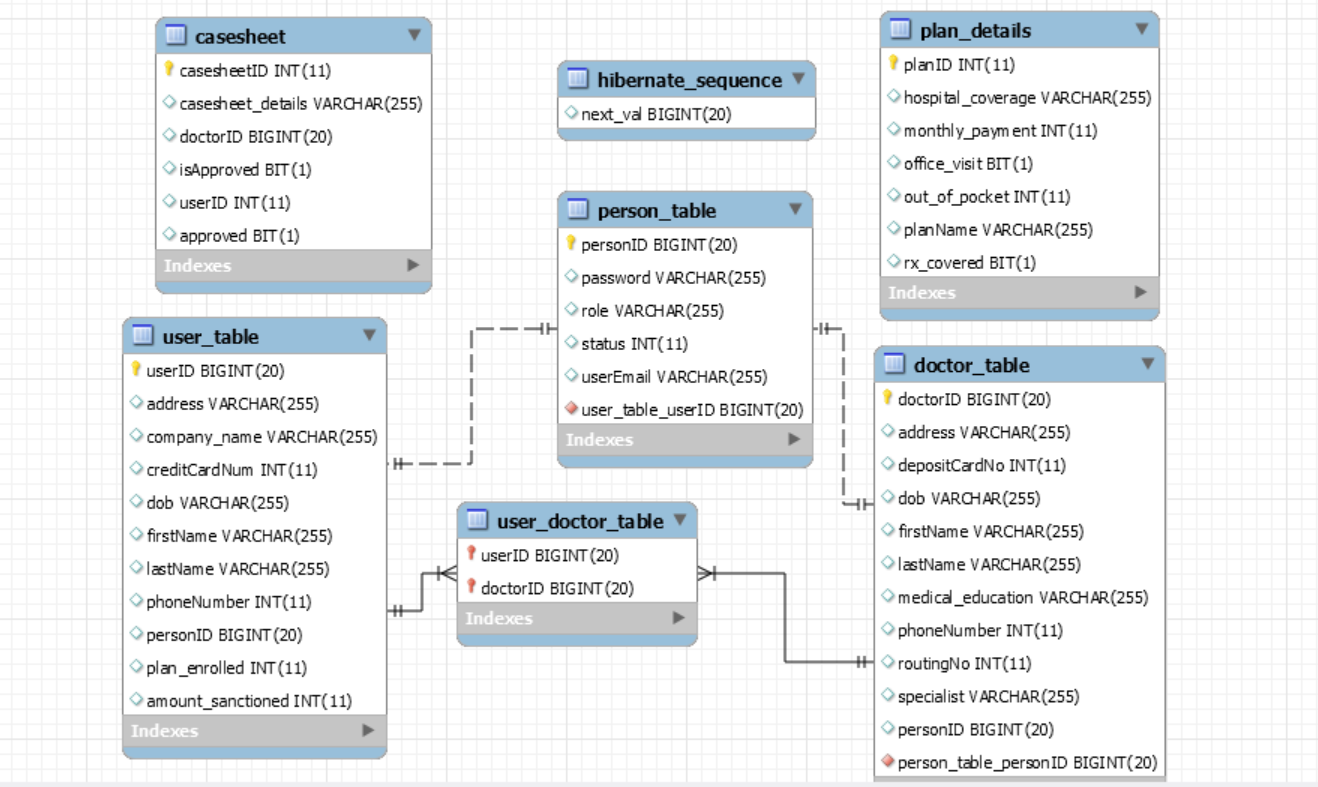
Other Securities – CAPTCHA

## Roles

Patient – Has one to one relationship with the Person and has ManyToMany Relationship with the Doctor

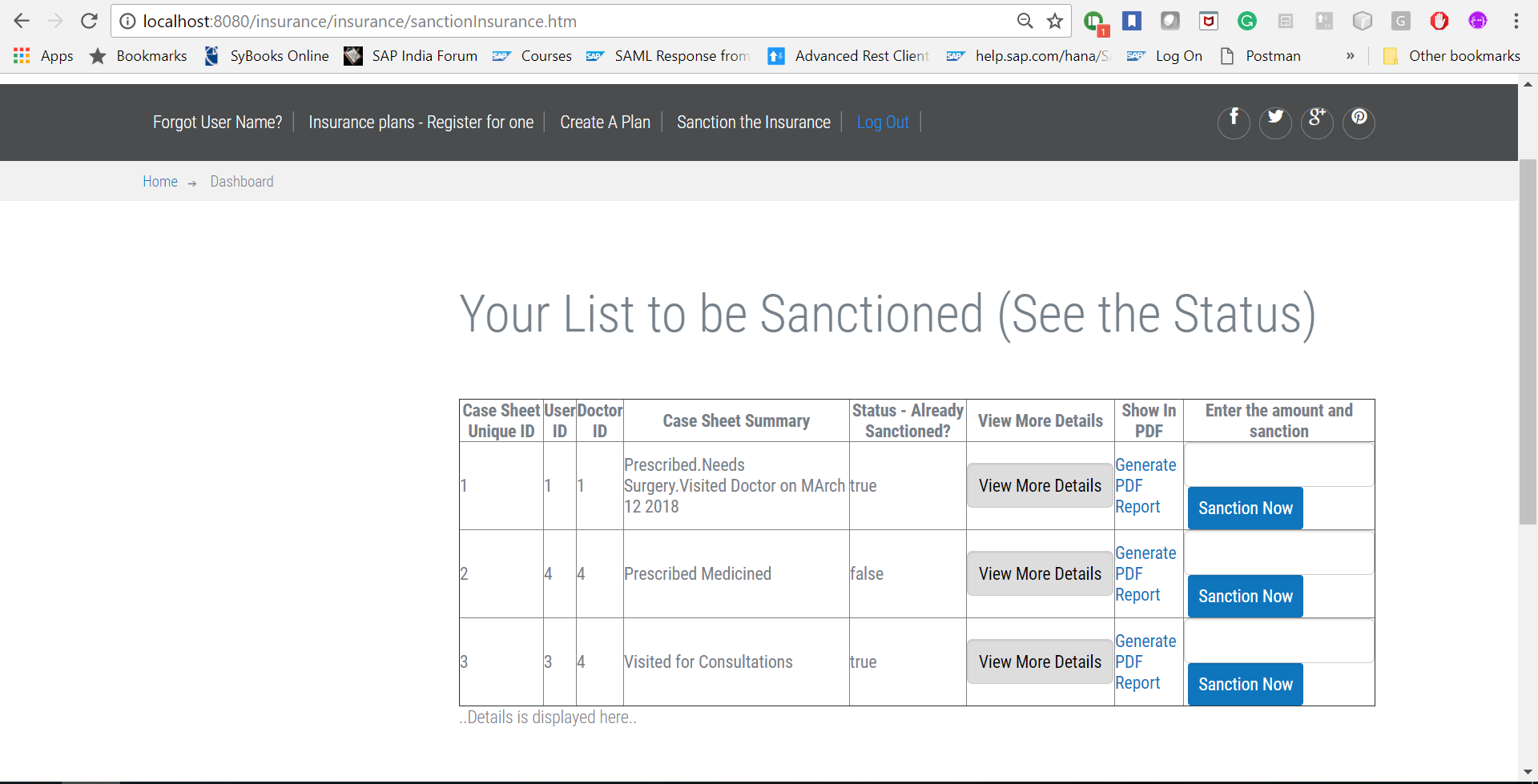
Doctor - Has one to one relationship with the User and has ManyToMany Relationship with the Doctor

Admin – Declarative Web Security – admin is the user role.



## Snapshots

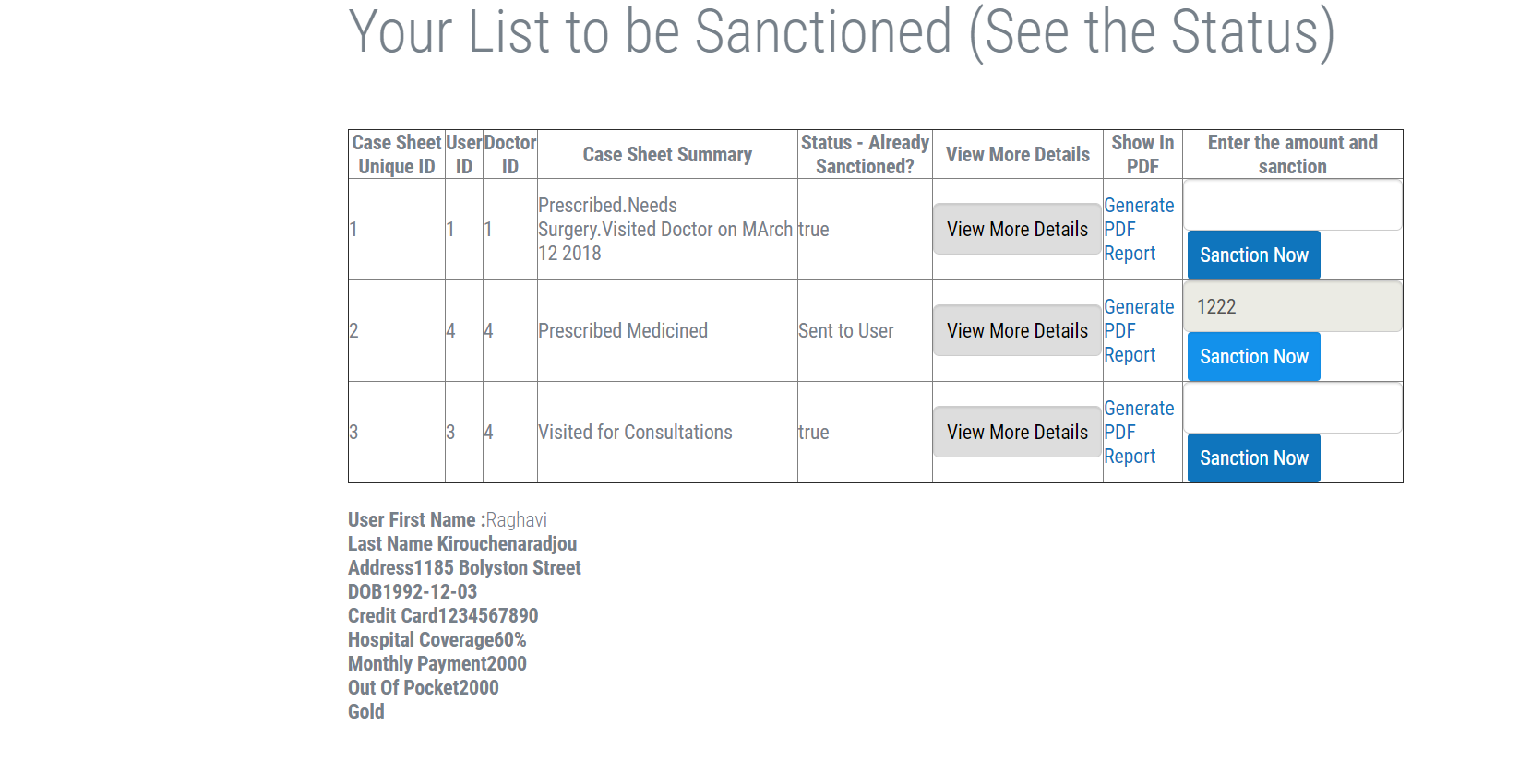
**Admin Panel – Sanction the Claim – AJAX with JQUERY and PDF Viewer**



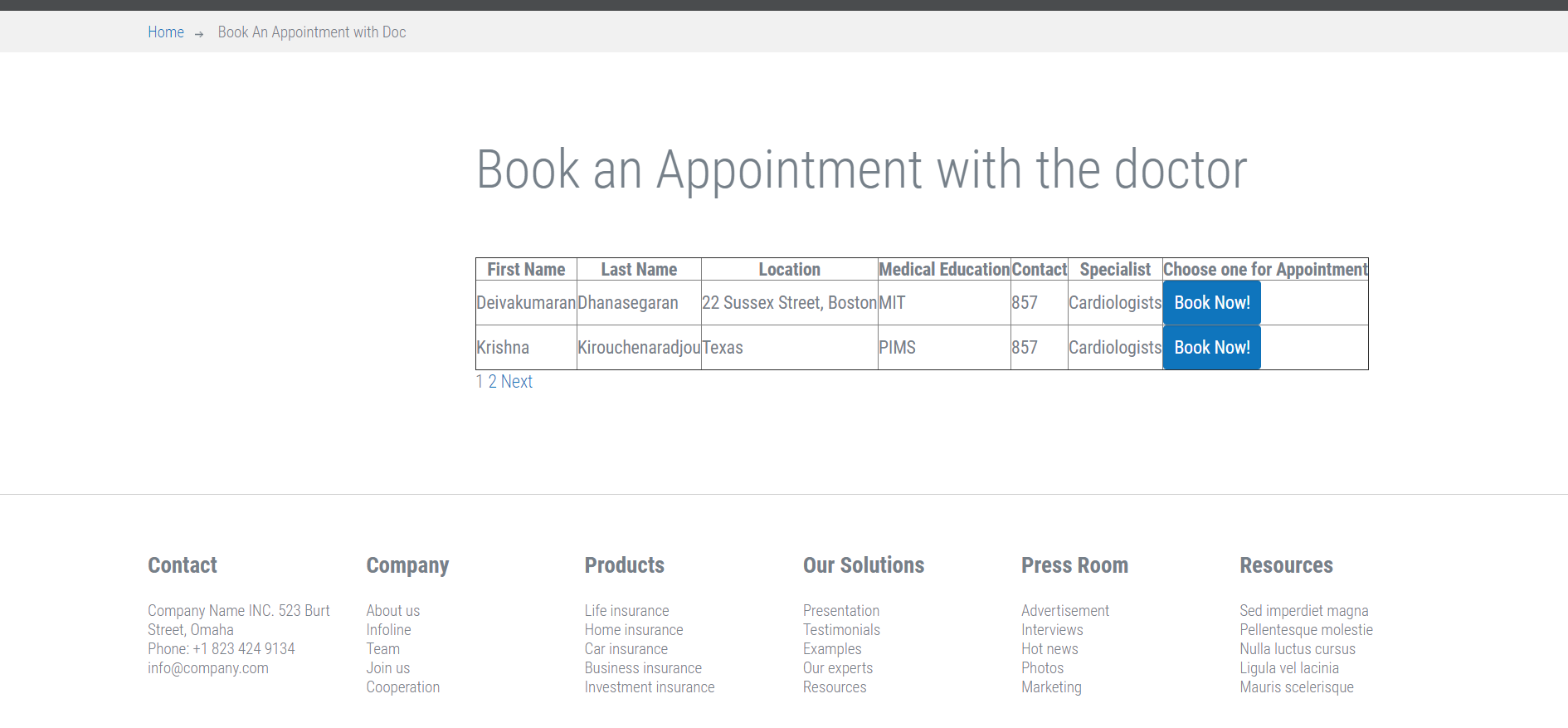
**View More Details is an AJAX Call:**



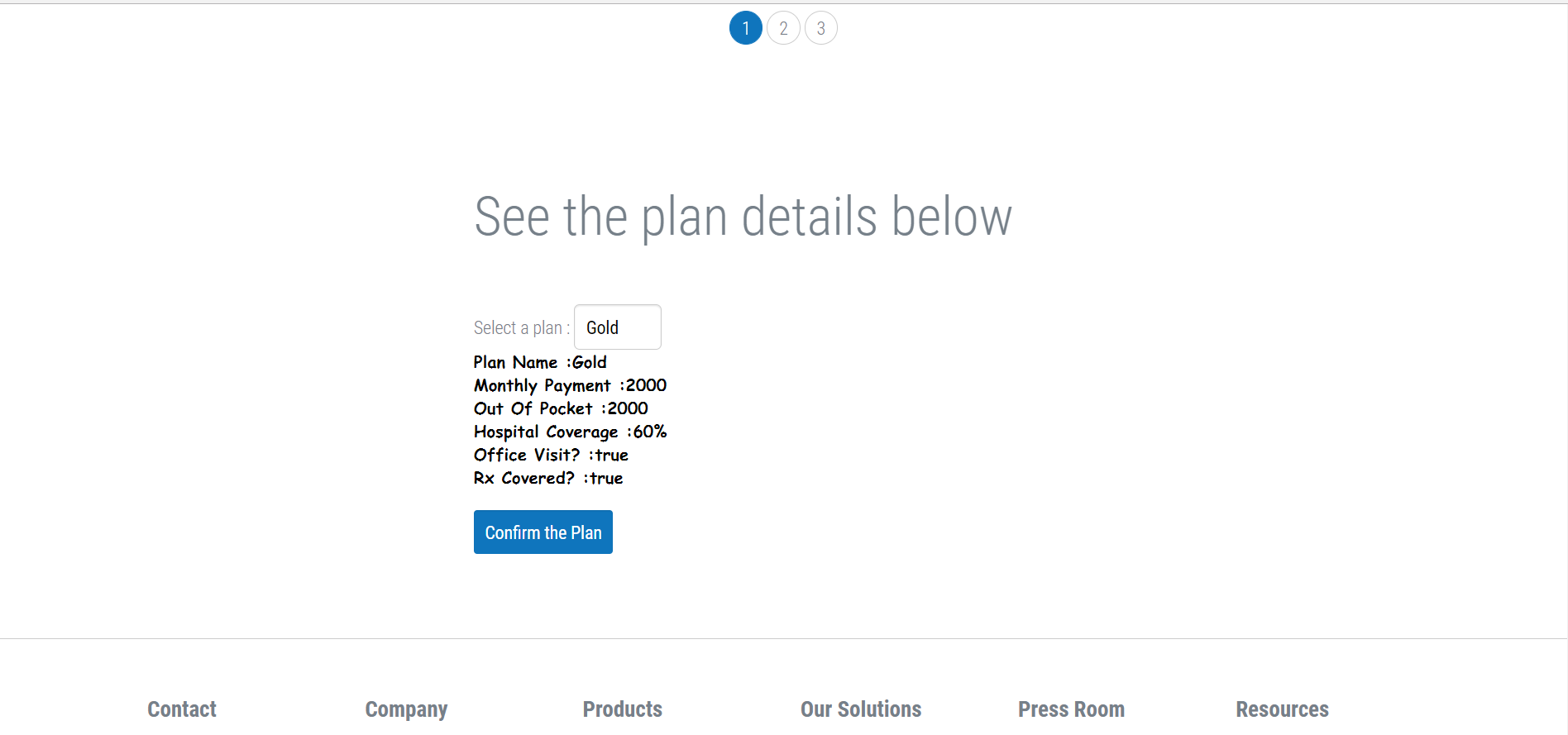
**Sanction Now is also An AJAX CALL which disables the button and updates the status.**



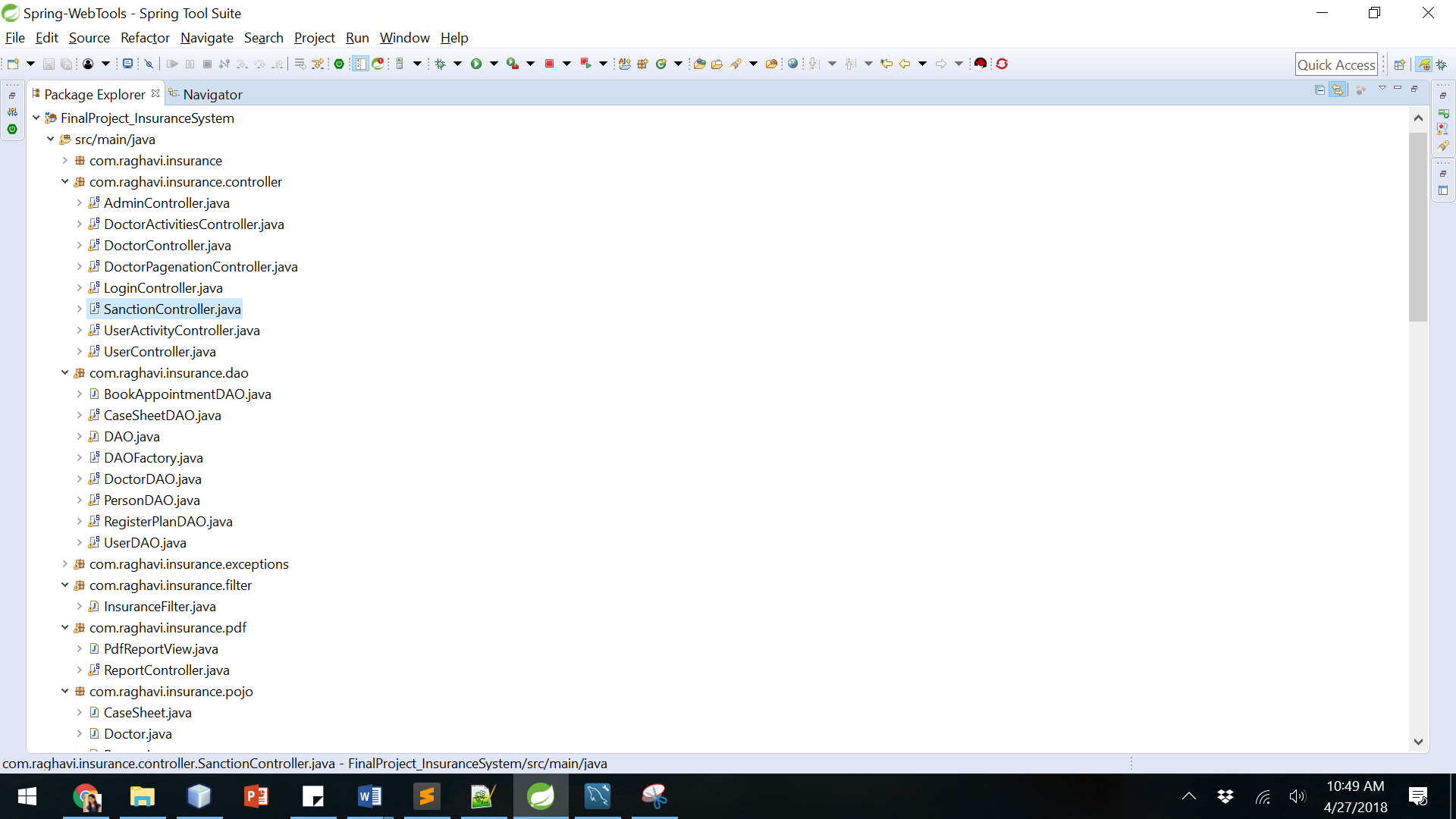
**Patient – Book an Appointment with Doctor – Pagination**

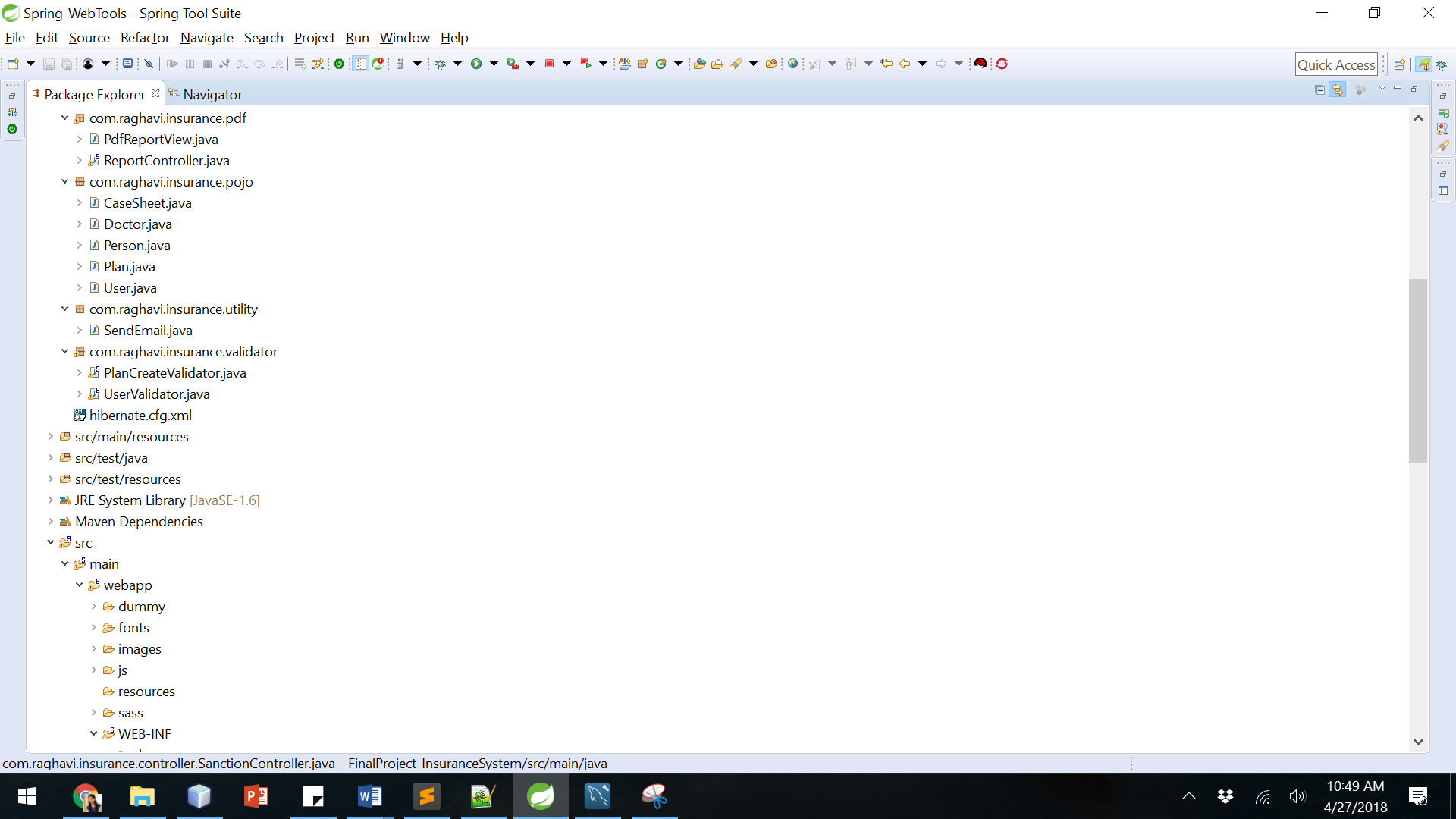


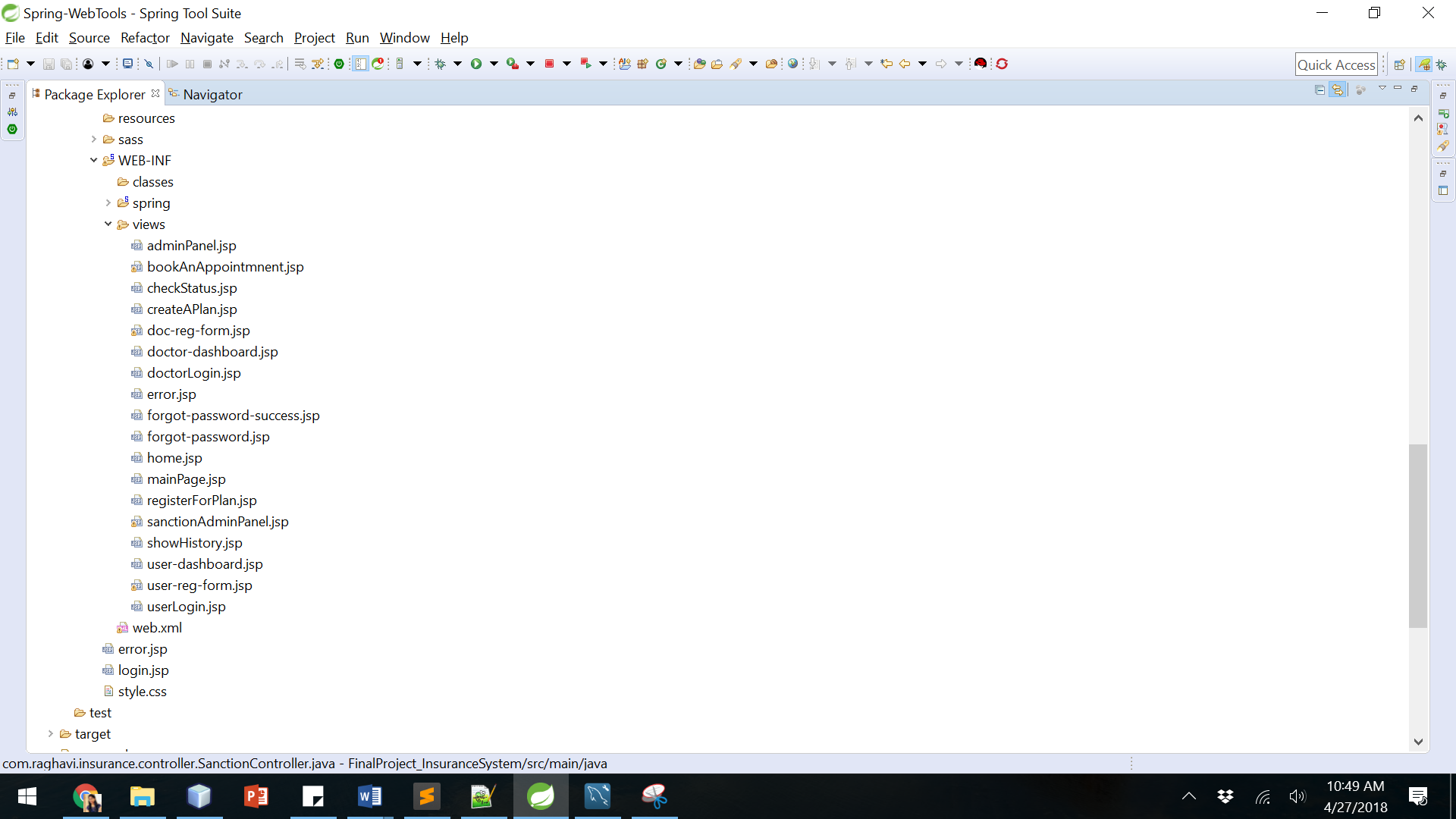
Patient – Confirm/Enroll Plan – AJAX to get the plan details



## Project Structure







## APPENDIX

1. CONTROLLER SOURCE CODE
   1. ADMIN
      1. ADMIN CONTROLLER------------------------------------------------------------------------ 9
      2. SANCTION CONTROLLER-------------------------------------------------------------------- 19
   2. DOCTOR
      1. DOCTOR ACTIVITIES CONTROLLER--------------------------------------------------------- 11
      2. DOCTOR CONTROLLER------------------------------------------------------------------------ 12
      3. DOCTOR PAGENATION CONTROLLER------------------------------------------------------16
   3. USER
      1. USER ACTIVITIES CONTROLLER---------------------------------------------------------------21
      2. USER CONTROLLER------------------------------------------------------------------------------23
   4. LOGIN CONTROLLER---------------------------------------------------------------------------------------27
2. POJO SOURCE CODE
   1. CASESHEET--------------------------------------------------------------------------------------------------28
   2. DOCTOR------------------------------------------------------------------------------------------------------30
   3. PERSON-------------------------------------------------------------------------------------------------------34
   4. PLAN-----------------------------------------------------------------------------------------------------------36
   5. USER-----------------------------------------------------------------------------------------------------------38

## Controller Source Code

AdminController

**package** com.raghavi.insurance.controller;

/\*\*

\* **@author** Raghavi Kirouchenaradjou

\*

\*/

@Controller

@RequestMapping("/insurance/\*")

**public** **class** AdminController {

@Autowired

@Qualifier("planCreateValidator")

PlanCreateValidator planCreateValidator;

@InitBinder

**private** **void** initBinder(WebDataBinder binder) {

binder.setValidator(planCreateValidator);

}

@Autowired

DAOFactory daoFactory;

@RequestMapping(value = "/insurance/createAPlan.htm", method = RequestMethod.***GET***)

**public** ModelAndView createAPlan(HttpServletRequest request) {

**return** **new** ModelAndView("createAPlan", "plan", **new** Plan());

}

@RequestMapping(value = "/insurance/sanctionInsurance.htm", method = RequestMethod.***GET***)

**public** ModelAndView sanctionInsurance(HttpServletRequest request) {

ModelAndView modelAndView = **new** ModelAndView("sanctionAdminPanel");

List<CaseSheet> caseSheets = **new** ArrayList<CaseSheet>();

**try**

{

CaseSheetDAO caseSheetDAO = daoFactory.createCaseSheetDAO();

caseSheets = caseSheetDAO.getCaseSheets();

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("type", "caseSheets");

map.put("cases", caseSheets);

modelAndView.addObject("map", map);

**return** modelAndView;

}

**catch**(Exception ex)

{

System.***out***.println(ex.getMessage()+"Error in Admin Controller - Sanction");

}

**return** modelAndView;

}

@RequestMapping(value = "/insurance/planCreate.htm", method = RequestMethod.***POST***)

**public** String handleCreateForm(HttpServletRequest request, @ModelAttribute("plan") Plan plan, ModelMap map,BindingResult result) {

RegisterPlanDAO registerPlanDAO = daoFactory.createPlanDAO();

planCreateValidator.validate(plan, result);

**if** (result.hasErrors()) {

**return** "error-page";

}

**try**

{

System.***out***.print("Create a Plan");

Plan u = registerPlanDAO.register(plan);

}

**catch** (PlanCreateExceptions e) {

System.***out***.println(e.getMessage());

**return** "error";

}

**return** "adminPanel";

}

}

### DoctorActivitiesController

/\*\*

\* **@author** Raghavi Kirouchenaradjou

\*

\*/

@Controller

@RequestMapping("/insurance/\*")

**public** **class** DoctorActivitiesController {

@Autowired

DAOFactory daoFactory;

@RequestMapping(value = "/insurance/getUserList.htm", method = RequestMethod.***POST***)

@ResponseBody

**public** String createAPlan(HttpServletRequest request) {

String userID = request.getParameter("userID");

HttpSession session = request.getSession();

String emailID = (String) session.getAttribute("username");

DoctorDAO doctor = daoFactory.createDoctorDAO();

CaseSheetDAO caseSheetDAO = daoFactory.createCaseSheetDAO();

String caseSheetDetails = request.getParameter("caseSheetID");

**try** {

Doctor doctorID = doctor.getDoctorID(emailID);

CaseSheet caseSheet = **new** CaseSheet(Integer.*parseInt*(userID), doctorID.getDoctorID(), caseSheetDetails,**false**);

**boolean** success = caseSheetDAO.createCaseSheet(caseSheet);

**if** (success)

**return** "success";

**else**

**return** "fail";

} **catch** (Exception e) {

e.printStackTrace();

}

**return** **null**;

}

}

### Doctor Controller

@Controller

@RequestMapping("/insurance/\*")

**public** **class** DoctorController {

@Autowired

DAOFactory daoFactory;

@RequestMapping(value = "/insurance/doclogin.htm", method = RequestMethod.***POST***)

**public** ModelAndView handleLoginForm(HttpServletRequest request, PersonDAO personDAO, ModelMap map) {

String username = request.getParameter("username");

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

**try** {

HttpSession session = request.getSession(**true**);

session.setAttribute("username", username);

Person u = personDAO.get(username, password);

**if** (u != **null** && u.getStatus() == 1) {

ModelAndView modelAndView = **new** ModelAndView("doctor-dashboard");

List<User> userList = **new** ArrayList<User>();

**try** {

Set<User> userListDetails = u.getDoctor().getUsers();

userList = **new** ArrayList<User>(userListDetails);

} **catch** (Exception e) {

e.printStackTrace();

}

Map<String, Object> map1 = **new** HashMap<String, Object>();

map1.put("type", "userList");

map1.put("users", userList);

modelAndView.addObject("map1", map1);

**return** modelAndView;

} **else** **if** (u != **null** && u.getStatus() == 0) {

map.addAttribute("errorMessage", "Please activate your account to login!");

**return** **new** ModelAndView("error-page");

} **else** {

map.addAttribute("errorMessage", "Invalid username/password!");

**return** **new** ModelAndView("error-page");

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** **null**;

}

@RequestMapping(value = "/insurance/docCreate.htm", method = RequestMethod.***GET***)

**public** String showCreateForm() {

**return** "doc-reg-form";

}

@RequestMapping(value = "/insurance/showHistory.htm", method = RequestMethod.***GET***)

**public** ModelAndView showHistory(HttpServletRequest request) {

List<CaseSheet> caseSheets = **new** ArrayList<CaseSheet>();

ModelAndView modelAndView = **new** ModelAndView("showHistory");

HttpSession session = request.getSession();

String emailID = (String) session.getAttribute("username");

**try** {

DoctorDAO doctorDAO = daoFactory.createDoctorDAO();

Doctor doc = doctorDAO.getDoctorID(emailID);

CaseSheetDAO caseSheetDAO = daoFactory.createCaseSheetDAO();

caseSheets = caseSheetDAO.getCaseSheetsByDoctor(doc.getDoctorID());

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("type", "caseSheets");

map.put("cases", caseSheets);

modelAndView.addObject("map", map);

**return** modelAndView;

} **catch** (Exception ex) {

System.***out***.println(ex);

}

**return** modelAndView;

}

@RequestMapping(value = "/insurance/docCreate.htm", method = RequestMethod.***POST***)

**public** String handleCreateForm(HttpServletRequest request, PersonDAO personDAO, ModelMap map) {

Captcha captcha = Captcha.*load*(request, "CaptchaObject");

String captchaCode = request.getParameter("captchaCode");

HttpSession session = request.getSession();

**if** (captcha.validate(captchaCode)) {

String useremail = request.getParameter("username");

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

String firstName = request.getParameter("firstName");

String lastName = request.getParameter("lastName");

Integer phoneNumber = Integer.*parseInt*(request.getParameter("phoneNumber"));

String specialist = request.getParameter("specialist");

String address = request.getParameter("address");

String medical\_education = request.getParameter("medical\_education");

Integer depositCardNo = Integer.*parseInt*(request.getParameter("depositCardNo"));

Integer routingNo = Integer.*parseInt*(request.getParameter("routingNo"));

Person person = **new** Person();

Doctor doctor = **new** Doctor(firstName, lastName, phoneNumber, specialist, address, medical\_education,

depositCardNo, routingNo, request.getParameter("dob"));

person.setUserEmail(useremail);

person.setPassword(password);

person.setRole("doctor");

person.setStatus(0);

doctor.setPerson(person);

person.setDoctor(doctor);

**try** {

Person u = personDAO.register(person);

Random rand = **new** Random();

**int** randomNum1 = rand.nextInt(5000000);

**int** randomNum2 = rand.nextInt(5000000);

**try** {

String str = "http://localhost:8080/insurance/insurance/validateemail.htm?email=" + useremail

+ "&key1=" + randomNum1 + "&key2=" + randomNum2;

session.setAttribute("key1", randomNum1);

session.setAttribute("key2", randomNum2);

sendEmail(useremail, "Click on this link to activate your account : " + str);

} **catch** (Exception e) {

System.***out***.println("Email cannot be sent");

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

} **else** {

map.addAttribute("errorMessage", "Invalid Captcha!");

**return** "doc-reg-form";

}

**return** "doctorLogin";

}

/\*\*

\* **@param** userEmail

\* **@param** string

\*/

**public** **void** sendEmail(String userEmail, String message) {

**try** {

Email email = **new** SimpleEmail();

email.setHostName("smtp.googlemail.com");

email.setSmtpPort(465);

email.setAuthenticator(**new** DefaultAuthenticator("contactapplication2018@gmail.com", "springmvc"));

email.setSSLOnConnect(**true**);

email.setFrom("no-reply@msis.neu.edu"); // This user email does not

// exist

email.setSubject("Web tools lab ");

email.setMsg(message); // Retrieve email from the DAO and send this

email.addTo(userEmail);

email.send();

} **catch** (EmailException e) {

System.***out***.println("Email cannot be sent");

}

}

@RequestMapping(value = "/insurance/validateemail.htm", method = RequestMethod.***GET***)

**public** String validateEmail(HttpServletRequest request, PersonDAO personDao, ModelMap map) {

// The user will be sent the following link when the use registers

// This is the format of the email

// http://hostname:8080/lab10/user/validateemail.htm?email=useremail&key1=<random\_number>&key2=<body

// of the email that when user registers>

HttpSession session = request.getSession();

String email = request.getParameter("email");

**int** key1 = Integer.*parseInt*(request.getParameter("key1"));

**int** key2 = Integer.*parseInt*(request.getParameter("key2"));

System.***out***.println(session.getAttribute("key1"));

System.***out***.println(session.getAttribute("key2"));

**if** ((Integer) (session.getAttribute("key1")) == key1 && ((Integer) session.getAttribute("key2")) == key2) {

**try** {

System.***out***.println("HI\_\_\_\_\_\_\_\_");

**boolean** updateStatus = personDao.updateUser(email);

**if** (updateStatus) {

**return** "doctorLogin";

} **else** {

**return** "error";

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

} **else** {

map.addAttribute("errorMessage", "Link expired , generate new link");

map.addAttribute("resendLink", **true**);

**return** "error";

}

**return** "doctorLogin";

}

@RequestMapping(value = "/insurance/forgotpassword.htm", method = RequestMethod.***GET***)

**public** String forgotPassword() {

**return** "forgot-password";

}

@RequestMapping(value = "/insurance/forgotpassword.htm", method = RequestMethod.***POST***)

**public** String handleForgotPasswordForm(HttpServletRequest request, PersonDAO personDAO) {

String useremail = request.getParameter("useremail");

Captcha captcha = Captcha.*load*(request, "CaptchaObject");

String captchaCode = request.getParameter("captchaCode");

**if** (captcha.validate(captchaCode)) {

Person user = personDAO.get(useremail);

sendEmail(useremail, "Your password is : " + user.getPassword());

**return** "forgot-password-success";

} **else** {

request.setAttribute("captchamsg", "Captcha not valid");

**return** "forgot-password";

}

}

}

### DoctorPagenationController

@Controller

@RequestMapping("/insurance/\*")

**public** **class** DoctorPagenationController {

@Autowired

DAOFactory daoFactory;

@RequestMapping(value = "/insurance/getDoctorList.htm", method = RequestMethod.***GET***)

**public** ModelAndView listOfUsers(@RequestParam(required = **false**) Integer page,

@RequestParam(required = **false**) String specilaist,HttpServletRequest request) {

ModelAndView modelAndView = **new** ModelAndView("bookAnAppointmnent");

List<Doctor> docorList = **new** ArrayList<Doctor>();

DoctorDAO doctorDAO = daoFactory.createDoctorDAO();

**try** {

**if**(request.getParameter("specialistSelection")==**null** || request.getParameter("specialistSelection").equals("All"))

{

docorList = doctorDAO.getDoclist();

}

**else**

{

docorList = doctorDAO.getDoclist(request.getParameter("specialistSelection"));

}

} **catch** (DoctorException ex) {

System.***out***.println("Exception while getting doc details" + ex);

}

PagedListHolder<Doctor> pagedListHolder = **new** PagedListHolder<Doctor>(docorList);

pagedListHolder.setPageSize(2);

modelAndView.addObject("maxPages", pagedListHolder.getPageCount());

**if** (page == **null** || page < 1 || page > pagedListHolder.getPageCount())

page = 1;

modelAndView.addObject("page", page);

**if** (page == **null** || page < 1 || page > pagedListHolder.getPageCount()) {

pagedListHolder.setPage(0);

modelAndView.addObject("users", pagedListHolder.getPageList());

} **else** **if** (page <= pagedListHolder.getPageCount()) {

pagedListHolder.setPage(page - 1);

modelAndView.addObject("users", pagedListHolder.getPageList());

}

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("type", "getDoc");

modelAndView.addObject("map", map);

**return** modelAndView;

}

@RequestMapping(value = "/insurance/getDoctorList.htm", method = RequestMethod.***POST***)

**public** ModelAndView bookAnAppointment(HttpServletRequest request) {

ModelAndView modelAndView = **new** ModelAndView("bookAnAppointmnent");

HttpSession session = request.getSession(**false**);

String emailID = (String) session.getAttribute("username");

**int** doctorID = Integer.*parseInt*(request.getParameter("docwho"));

**try** {

DoctorDAO doctorDAO = daoFactory.createDoctorDAO();

doctorDAO.bookAnAppointment(doctorID,emailID);

} **catch** (Exception e) {

}

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("type", "bookNow");

modelAndView.addObject("map", map);

**return** modelAndView;

}

}

### SanctionController

@Controller

@RequestMapping("/insurance/\*")

**public** **class** SanctionController {

@Autowired

DAOFactory daoFactory;

@RequestMapping(value = "/insurance/getMoreDetails.htm", method = RequestMethod.***POST***)

@ResponseBody

**public** String getPlanDetails(HttpServletRequest request) {

String result = "";

UserDAO userDAO = daoFactory.createUserDAO();

RegisterPlanDAO registerPlanDAO = daoFactory.createPlanDAO();

**int** userID = Integer.*parseInt*(request.getParameter("userID"));

**try** {

User user = userDAO.getUserByUserID(userID);

Plan planDetails = registerPlanDAO.getPlanDetails(user.getPlan\_enrolled());

result = "<br><strong>User First Name :</strong>"+user.getFirstName() + "<br><strong>Last Name </stroong>" + user.getLastName() + "<br><strong>Address</strong>" + user.getAddress() + "<br>DOB"

+ user.getDob() + "<br><strong>Credit Card</strong>" + user.getCreditCardNum() + "<br><strong>Hospital Coverage</strong>" + planDetails.getHospital\_coverage()

+ "<br><strong>Monthly Payment</strong>" + planDetails.getMonthly\_payment() + "<br><strong>Out Of Pocket</strong>" + planDetails.getOut\_of\_pocket() + "<br>"

+ planDetails.getPlanName();

**return** result;

} **catch** (PlanCreateExceptions e) {

System.***out***.println(e.getMessage());

}

**return** **null**;

}

@RequestMapping(value = "/insurance/sanctionInsuranceNow.htm", method = RequestMethod.***POST***)

@ResponseBody

**public** String santionInsurance(HttpServletRequest request) {

UserDAO userDAO = daoFactory.createUserDAO();

CaseSheetDAO caseSheetDAO = daoFactory.createCaseSheetDAO();

**int** userID = Integer.*parseInt*(request.getParameter("userID"));

**try** {

User user = userDAO.getUserByUserID(userID);

user.setAmount\_sanctioned(Integer.*parseInt*(request.getParameter("amount\_sanctioned")));

**boolean** yesApproved = userDAO.update(user);

**boolean** valuUpdated = caseSheetDAO.update(Integer.*parseInt*(request.getParameter("caseSheetID")));

String message ="Dear ".concat(user.getFirstName()).concat("/n")+"Hurray! You have been sanctioned the amount";

SendEmail.*sendEmail*(user.getPerson().getUserEmail(),message );

**if** (yesApproved && valuUpdated)

**return** "success";

**else**

**return** "failedToSanction";

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}

**return** **null**;

}

}

### UserActivityController

@Controller

@RequestMapping("/insurance/\*")

**public** **class** UserActivityController {

@Autowired

DAOFactory daoFactory;

@RequestMapping(value = "/insurance/selection.htm", method = RequestMethod.***POST***)

**public** ModelAndView activityDropDown(HttpServletRequest request) {

String action = request.getParameter("activityDropDown");

**if** (action.equals("bookAnAppointmnent")) {

**return** **new** ModelAndView("bookAnAppointmnent");

} **else** **if** (action.equals("registerForPlan")) {

**return** **new** ModelAndView("registerForPlan");

}

**else** **if** (action.equals("checkStatus")) {

HttpSession session = request.getSession(**false**);

String emailID = (String) session.getAttribute("username");

ModelAndView modelAndView = **new** ModelAndView("checkStatus");

**try**

{

UserDAO userDAO = **new** UserDAO();

User userByEmail = userDAO.getUserByEmailID(emailID);

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("type", "users");

map.put("user", userByEmail);

modelAndView.addObject("map", map);

**return** modelAndView;

}

**catch**(Exception ex)

{

System.***out***.println(ex.getMessage()+"Error in Check Ststus Controller - Sanction");

}

**return** modelAndView;

}

**return** **new** ModelAndView();

}

//Ajax to retrieve the plan Details

@RequestMapping(value = "/insurance/getPlanDetails.htm", method = RequestMethod.***POST***)

@ResponseBody

**public** String getPlanDetails(HttpServletRequest request) {

String result="";

RegisterPlanDAO registerPlanDAO = daoFactory.createPlanDAO();

**int** planID = Integer.*parseInt*(request.getParameter("planID"));

**try** {

Plan planDetails = registerPlanDAO.getPlanDetails(planID);

result = "<div class='planStyle'> Plan Name :"+planDetails.getPlanName() + "<br>" +

"<div class='planStyle'> Monthly Payment :"+ planDetails.getMonthly\_payment() + "<br>" +

"<div class='planStyle'> Out Of Pocket :"+ planDetails.getOut\_of\_pocket() + "<br>" +

"<div class='planStyle'> Hospital Coverage :"+ planDetails.getHospital\_coverage() + "<br>"+

"<div class='planStyle'> Office Visit? :"+ planDetails.isOffice\_visit() + "<br>" +

"<div class='planStyle'> Rx Covered? :"+ planDetails.isRx\_covered() + "<br>";

**return** result;

} **catch** (PlanCreateExceptions e) {

System.***out***.println(e.getMessage());

}

**return** **null**;

}

@RequestMapping(value = "/insurance/confirmPlan.htm", method = RequestMethod.***POST***)

**public** String confirmThePlan(HttpServletRequest request) {

HttpSession session = request.getSession(**false**);

String emailID = (String) session.getAttribute("username");

**int** planSelectionID = Integer.*parseInt*(request.getParameter("planSelection"));

RegisterPlanDAO updatePlan = daoFactory.createPlanDAO();

**try** {

**boolean** success = updatePlan.updatePlanEnrolled(emailID, planSelectionID);

**if**(success)

{

**return** "user-dashboard";

}

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}

**return** "error-page";

}

@RequestMapping(value = "/insurance/confirmPlan.htm", method = RequestMethod.***GET***)

**public** String getToUserDashboard(HttpServletRequest request) {

**return** "user-dashboard";

}

}

### UserController

@Controller

@RequestMapping("/insurance/\*")

**public** **class** UserController {

/\*@Autowired

@Qualifier("userValidator")

UserValidator validator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}\*/

@RequestMapping(value = "/insurance/userlogin.htm", method = RequestMethod.***POST***)

**public** String handleLoginForm(HttpServletRequest request, PersonDAO personDAO, ModelMap map) {

String username = request.getParameter("username");

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

**try** {

HttpSession session = request.getSession();

session.setAttribute("username", username);

Person u = personDAO.get(username, password);

**if** (u != **null** && u.getStatus() == 1) {

**return** "user-dashboard";

} **else** **if** (u != **null** && u.getStatus() == 0) {

map.addAttribute("errorMessage", "Please activate your account to login!");

**return** "error";

} **else** {

map.addAttribute("errorMessage", "Invalid username/password!");

**return** "error";

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** **null**;

}

@RequestMapping(value = "/insurance/userCreate.htm", method = RequestMethod.***GET***)

**public** ModelAndView showCreateForm() {

**return** **new** ModelAndView("user-reg-form", "user", **new** User());

}

@RequestMapping(value = "/insurance/userCreate.htm", method = RequestMethod.***POST***)

**public** String handleCreateForm(HttpServletRequest request, @ModelAttribute("user") User user,PersonDAO personDAO, ModelMap map,BindingResult result) {

/\*validator.validate(user, result);

if (result.hasErrors()) {

return "";

}\*/

Captcha captcha = Captcha.*load*(request, "CaptchaObject");

String captchaCode = request.getParameter("captchaCode");

HttpSession session = request.getSession();

**if** (captcha.validate(captchaCode)) {

System.***out***.println(user.getFirstName());

String firstName = request.getParameter("firstName");

String lastName = request.getParameter("lastName");

// phoneNumber = Integer.parseInt(request.getParameter("phoneNumber"));

String address = request.getParameter("address");

//Integer creditCardNum = Integer.parseInt(request.getParameter("creditCardNum"));

String company\_name = request.getParameter("company\_name");

Person person = **new** Person();

User userAccount = **new** User(firstName,lastName, user.getPhoneNumber(), address, user.getCreditCardNum(), request.getParameter("dob"), company\_name,0,0);

person.setUserEmail(user.getPerson().getUserEmail());

person.setPassword(user.getPerson().getPassword());

person.setRole("user");

person.setStatus(0);

userAccount.setPerson(person);

person.setUser(userAccount);

**try** {

Person u = personDAO.register(person);

Random rand = **new** Random();

**int** randomNum1 = rand.nextInt(5000000);

**int** randomNum2 = rand.nextInt(5000000);

**try** {

String str = "http://localhost:8080/insurance/insurance/validateuseremail.htm?email=" + user.getPerson().getUserEmail() + "&key1="

+ randomNum1 + "&key2=" + randomNum2;

session.setAttribute("key1", randomNum1);

session.setAttribute("key2", randomNum2);

String message ="Dear ".concat(userAccount.getFirstName()).concat("/n")+"Click on this link to activate your account : " + str;

sendEmail(user.getPerson().getUserEmail(),message );

} **catch** (Exception e) {

System.***out***.println("Email cannot be sent");

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

} **else** {

map.addAttribute("errorMessage", "Invalid Captcha!");

**return** "user-reg-form";

}

**return** "userLogin";

}

/\*\*

\* **@param** userEmail

\* **@param** string

\*/

**public** **void** sendEmail(String userEmail, String message) {

**try** {

Email email = **new** SimpleEmail();

email.setHostName("smtp.googlemail.com");

email.setSmtpPort(465);

email.setAuthenticator(**new** DefaultAuthenticator("contactapplication2018@gmail.com", "springmvc"));

email.setSSLOnConnect(**true**);

email.setFrom("no-reply@msis.neu.edu"); // This user email does not

// exist

email.setSubject("Insurance Company ");

email.setMsg(message); // Retrieve email from the DAO and send this

email.addTo(userEmail);

email.send();

} **catch** (EmailException e) {

System.***out***.println("Email cannot be sent");

}

}

@RequestMapping(value = "/insurance/validateuseremail.htm", method = RequestMethod.***GET***)

**public** String validateEmail(HttpServletRequest request, PersonDAO personDAO, ModelMap map) {

// The user will be sent the following link when the use registers

// This is the format of the email

// http://hostname:8080/lab10/user/validateemail.htm?email=useremail&key1=<random\_number>&key2=<body

// of the email that when user registers>

HttpSession session = request.getSession();

String email = request.getParameter("email");

**int** key1 = Integer.*parseInt*(request.getParameter("key1"));

**int** key2 = Integer.*parseInt*(request.getParameter("key2"));

System.***out***.println(session.getAttribute("key1"));

System.***out***.println(session.getAttribute("key2"));

**if** ((Integer) (session.getAttribute("key1")) == key1 && ((Integer) session.getAttribute("key2")) == key2) {

**try** {

System.***out***.println("HI\_\_\_\_\_\_\_\_");

**boolean** updateStatus = personDAO.updateUser(email);

**if** (updateStatus) {

**return** "userLogin";

} **else** {

**return** "error";

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

} **else** {

map.addAttribute("errorMessage", "Link expired , generate new link");

map.addAttribute("resendLink", **true**);

**return** "error";

}

**return** "userLogin";

}

}

### LoginController

@Controller

**public** **class** LoginController {

@RequestMapping(value = "/insurance/login.htm", method = RequestMethod.***GET***)

**public** String showLoginForm() {

**return** "mainPage";

}

@RequestMapping(value = "/insurance/admin.htm", method = RequestMethod.***GET***)

**public** String showAdminPanel() {

**return** "adminPanel";

}

@RequestMapping(value = "/insurance/userMainPage.htm", method = RequestMethod.***GET***)

**public** String showUserPanel() {

**return** "userLogin";

}

@RequestMapping(value = "/insurance/doctorMainPage.htm", method = RequestMethod.***GET***)

**public** String showDoctorPanel() {

**return** "doctorLogin";

}

@RequestMapping(value = "/insurance/logOff.htm", method = RequestMethod.***GET***)

**public** String showLoginFormAfterLogOff(HttpServletRequest request) {

HttpSession session = request.getSession(**false**);

session.invalidate();

**return** "mainPage";

}

}

POJO CLASSES

### CASESHEET

@Entity

@Table(name="casesheet")

**public** **class** CaseSheet {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "casesheetID")

**private** **int** casesheetID;

@Column(name="userID")

**private** **int** userID;

@Column(name="doctorID")

**private** **long** doctorID;

@Column(name="casesheet\_details")

**private** String casesheet\_details;

@Column(name="approved")

**private** **boolean** approved;

**public** **int** getCasesheetID() {

**return** casesheetID;

}

**public** **int** getUserID() {

**return** userID;

}

**public** **void** setUserID(**int** userID) {

**this**.userID = userID;

}

**public** **long** getDoctorID() {

**return** doctorID;

}

**public** **void** setDoctorID(**int** doctorID) {

**this**.doctorID = doctorID;

}

**public** String getCasesheet\_details() {

**return** casesheet\_details;

}

**public** **void** setCasesheet\_details(String casesheet\_details) {

**this**.casesheet\_details = casesheet\_details;

}

**public** **void** setDoctorID(**long** doctorID) {

**this**.doctorID = doctorID;

}

**public** **boolean** isApproved() {

**return** approved;

}

**public** **void** setApproved(**boolean** approved) {

**this**.approved = approved;

}

/\*\*

\* **@param** userID

\* **@param** doctorID

\* **@param** casesheet\_details

\*/

**public** CaseSheet(**int** userID, **long** doctorID, String casesheet\_details,**boolean** approved) {

**this**.userID = userID;

**this**.doctorID = doctorID;

**this**.casesheet\_details = casesheet\_details;

**this**.approved = approved;

}

**public** CaseSheet()

{

}

}

DOCTOR

/\*\*

\* **@author** Raghavi Kirouchenaradjou

\*

\*/

@Entity

@Table(name = "doctor\_table")

**public** **class** Doctor {

//@GenericGenerator(name = "generator", strategy = "foreign", parameters = @Parameter(name = "property", value = "person"))

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "doctorID")//, unique = true, nullable = false)

**private** **long** doctorID;

@Column(name = "firstName")

**private** String firstName;

@Column(name = "lastName")

**private** String lastName;

@Column(name = "phoneNumber")

**private** **int** phoneNumber;

@Column(name = "specialist")

**private** String specialist;

@Column(name = "address")

**private** String address;

@Column(name = "medical\_education")

**private** String medical\_education;

@Column(name = "depositCardNo")

**private** **int** depositCardNo;

@Column(name = "routingNo")

**private** **int** routingNo;

@Column(name = "dob")

**private** String dob;

@OneToOne

@JoinColumn (name = "personID")

**private** Person person;

@ManyToMany(mappedBy="doctors")

**private** Set<User> users = **new** HashSet<User>();

**public** Set<User> getUsers() {

**return** users;

}

**public** **void** setUsers(Set<User> users) {

**this**.users = users;

}

**public** Person getPerson() {

**return** person;

}

**public** **void** setPerson(Person person) {

**this**.person = person;

}

/\*\*

\* **@param** firstName

\* **@param** lastName

\* **@param** phoneNumber

\* **@param** specialist

\* **@param** address

\* **@param** medical\_education

\* **@param** depositCardNo

\* **@param** routingNo

\*/

**public** Doctor(String firstName, String lastName, **int** phoneNumber, String specialist, String address,

String medical\_education, **int** depositCardNo, **int** routingNo,String dob) {

**super**();

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.phoneNumber = phoneNumber;

**this**.specialist = specialist;

**this**.address = address;

**this**.medical\_education = medical\_education;

**this**.depositCardNo = depositCardNo;

**this**.routingNo = routingNo;

**this**.dob = dob;

}

**public** String getDob() {

**return** dob;

}

**public** **void** setDob(String dob) {

**this**.dob = dob;

}

**public** **int** getPhoneNumber() {

**return** phoneNumber;

}

**public** **void** setPhoneNumber(**int** phoneNumber) {

**this**.phoneNumber = phoneNumber;

}

**public** String getSpecialist() {

**return** specialist;

}

**public** **void** setSpecialist(String specialist) {

**this**.specialist = specialist;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** String getMedical\_education() {

**return** medical\_education;

}

**public** **void** setMedical\_education(String medical\_education) {

**this**.medical\_education = medical\_education;

}

**public** **int** getDepositCardNo() {

**return** depositCardNo;

}

**public** **void** setDepositCardNo(**int** depositCardNo) {

**this**.depositCardNo = depositCardNo;

}

**public** **int** getRoutingNo() {

**return** routingNo;

}

**public** **void** setRoutingNo(**int** routingNo) {

**this**.routingNo = routingNo;

}

**public** Doctor() {

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** **long** getDoctorID() {

**return** doctorID;

}

@Override

**public** String toString(){

**return** firstName.concat(" ").concat(lastName);

}

}

PERSON

/\*\*

\* **@author** Raghavi Kirouchenaradjou

\*

\*/

@Entity

@Table(name = "person\_table")

**public** **class** Person {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "personID", unique = **true**, nullable = **false**)

**private** **long** personID;

@Column(name = "userEmail")

**private** String userEmail;

@Column(name = "password")

**private** String password;

@Column(name = "status")

**private** **int** status;

@Column(name = "role")

**private** String role;

@OneToOne(fetch = FetchType.***LAZY***, mappedBy = "person", cascade = CascadeType.***ALL***)

@PrimaryKeyJoinColumn

**private** Doctor doctor;

@OneToOne(fetch = FetchType.***LAZY***, mappedBy = "person", cascade = CascadeType.***ALL***)

@PrimaryKeyJoinColumn

**private** User user;

**public** Person() {

}

**public** Doctor getDoctor() {

**return** doctor;

}

**public** **void** setDoctor(Doctor doctor) {

**this**.doctor = doctor;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** String getRole() {

**return** role;

}

**public** **void** setRole(String string) {

**this**.role = string;

}

**public** String getUserEmail() {

**return** userEmail;

}

**public** **void** setUserEmail(String userEmail) {

**this**.userEmail = userEmail;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** **int** getStatus() {

**return** status;

}

**public** **void** setStatus(**int** status) {

**this**.status = status;

}

}

PLAN

@Entity

@Table(name="plan\_details")

**public** **class** Plan {

@Id

@GeneratedValue(strategy = GenerationType.***SEQUENCE***)

@Column(name = "planID")

**private** **int** planID;

@Column(name="planName")

**private** String planName;

@Column(name="monthly\_payment")

**private** **int** monthly\_payment;

@Column(name="hospital\_coverage")

**private** String hospital\_coverage;

@Column(name="office\_visit")

**private** **boolean** office\_visit;

@Column(name="out\_of\_pocket")

**private** **int** out\_of\_pocket;

@Column(name="rx\_covered")

**private** **boolean** rx\_covered;

**public** String getPlanName() {

**return** planName;

}

**public** **void** setPlanName(String planName) {

**this**.planName = planName;

}

**public** **int** getMonthly\_payment() {

**return** monthly\_payment;

}

**public** **void** setMonthly\_payment(**int** monthly\_payment) {

**this**.monthly\_payment = monthly\_payment;

}

**public** String getHospital\_coverage() {

**return** hospital\_coverage;

}

**public** **void** setHospital\_coverage(String hospital\_coverage) {

**this**.hospital\_coverage = hospital\_coverage;

}

**public** **boolean** isOffice\_visit() {

**return** office\_visit;

}

**public** **void** setOffice\_visit(**boolean** office\_visit) {

**this**.office\_visit = office\_visit;

}

**public** **int** getOut\_of\_pocket() {

**return** out\_of\_pocket;

}

**public** **void** setOut\_of\_pocket(**int** out\_of\_pocket) {

**this**.out\_of\_pocket = out\_of\_pocket;

}

**public** **boolean** isRx\_covered() {

**return** rx\_covered;

}

**public** **void** setRx\_covered(**boolean** rx\_covered) {

**this**.rx\_covered = rx\_covered;

}

/\*\*

\* **@param** planName

\* **@param** monthly\_payment

\* **@param** hospital\_coverage

\* **@param** office\_visit

\* **@param** out\_of\_pocket

\* **@param** rx\_covered

\*/

**public** Plan(String planName, **int** monthly\_payment, String hospital\_coverage, **boolean** office\_visit, **int** out\_of\_pocket,

**boolean** rx\_covered) {

**this**.planName = planName;

**this**.monthly\_payment = monthly\_payment;

**this**.hospital\_coverage = hospital\_coverage;

**this**.office\_visit = office\_visit;

**this**.out\_of\_pocket = out\_of\_pocket;

**this**.rx\_covered = rx\_covered;

}

**public** Plan() {

}

}

USER

@Entity

@Table(name = "user\_table")

**public** **class** User {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "userID")

**private** **long** userID;

@Column(name = "firstName")

**private** String firstName;

@Column(name = "lastName")

**private** String lastName;

@Column(name = "phoneNumber")

**private** **int** phoneNumber;

@Column(name = "address")

**private** String address;

@Column(name = "creditCardNum")

**private** **int** creditCardNum;

@Column(name = "dob")

**private** String dob;

@Column(name="company\_name")

**private** String company\_name;

@Column(name="plan\_enrolled")

**private** **int** plan\_enrolled;

@Column(name="amount\_sanctioned")

**private** **int** amount\_sanctioned;

@OneToOne

@JoinColumn (name = "personID")

**private** Person person;

@ManyToMany

@JoinTable (

name="user\_doctor\_table",

joinColumns = {@JoinColumn(name="userID", nullable = **false**, updatable = **false**)},

inverseJoinColumns = {@JoinColumn(name="doctorID" )}

)

**private** Set<Doctor> doctors = **new** HashSet<Doctor>();

**public** Set<Doctor> getDoctors() {

**return** doctors;

}

**public** **void** setDoctors(Set<Doctor> doctors) {

**this**.doctors = doctors;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** Person getPerson() {

**return** person;

}

**public** **void** setPerson(Person person) {

**this**.person = person;

}

**public** **int** getPhoneNumber() {

**return** phoneNumber;

}

**public** **void** setPhoneNumber(**int** phoneNumber) {

**this**.phoneNumber = phoneNumber;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** **int** getCreditCardNum() {

**return** creditCardNum;

}

**public** **void** setCreditCardNum(**int** creditCardNum) {

**this**.creditCardNum = creditCardNum;

}

**public** String getDob() {

**return** dob;

}

**public** **void** setDob(String dob) {

**this**.dob = dob;

}

**public** String getCompany\_name() {

**return** company\_name;

}

**public** **void** setCompany\_name(String company\_name) {

**this**.company\_name = company\_name;

}

**public** **int** getPlan\_enrolled() {

**return** plan\_enrolled;

}

**public** **void** setPlan\_enrolled(**int** plan\_enrolled) {

**this**.plan\_enrolled = plan\_enrolled;

}

**public** **long** getUserID() {

**return** userID;

}

**public** **int** getAmount\_sanctioned() {

**return** amount\_sanctioned;

}

**public** **void** setAmount\_sanctioned(**int** amount\_sanctioned) {

**this**.amount\_sanctioned = amount\_sanctioned;

}

/\*\*

\* **@param** firstName

\* **@param** lastName

\* **@param** phoneNumber

\* **@param** address

\* **@param** creditCardNum

\* **@param** dob

\* **@param** company\_name

\*/

**public** User(String firstName, String lastName, **int** phoneNumber, String address, **int** creditCardNum, String dob,

String company\_name,**int** plan\_enrolled,**int** amount\_sanctioned) {

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.phoneNumber = phoneNumber;

**this**.address = address;

**this**.creditCardNum = creditCardNum;

**this**.dob = dob;

**this**.company\_name = company\_name;

**this**.plan\_enrolled = plan\_enrolled;

**this**.amount\_sanctioned = amount\_sanctioned;

}

**public** User()

{

}

@Override

**public** String toString(){

**return** firstName.concat(" ").concat(lastName);

}

}