

1. REST/AJAX (**Kamanashis**)

Controller: AppointmentController

Method: findDoctorBySpecialization

Function: This method accepts a Specialization from a Path Variable (spec ID) and returns a Map containing Doctor objects found after querying the DB of doctors with the supplied specialization. The jQuery code can be found in /webapp/resources/js/main.js with the function name "changeSpecialization".

Testing Procedure:

1. Click Login button on Home page
2. Login as: Email = toanqc@gmail.com Password = toanqc
3. On Patient dashboard, click "Schedule an appointment" link
4. Click on Specialty select-box (i.e. Dentist, Ent), the corresponding doctors should appear on the Doctor select-box.

2. Validation

Controller: AppointmentController (**Toan Quach**)

Method: showRegistrationAppointment - *RequestMethod.POST*

Function: Save new appointment from patient with validations declared in the Appointment domain class in *java.mum.waa.fd.app.domain*

Testing Procedure:

1. Click Login button on Home page
2. Login as: Email = toanqc@gmail.com Password = toanqc
3. On Patient dashboard, click "Schedule an appointment" link
4. Submit the form. You should see validation messages.

Controller: AdminController (**Mark Pit**)

Method: saveDoctor

Function: Save new doctor information from admin with validations declared in the Doctor domain class in *java.mum.waa.fd.app.domain*

Testing Procedure:

1. Click Login button on Home page
2. Login as: Email = admin@gmail.com Password = admin
3. On dashboard, click "Add Doctor" link
4. Submit the form. You should see validation messages.

3. Tiles (**Mark Pit**)

Please check tilesViewResolver and tilesConfigurer beans in webapp/WEB-INF/config/servlet-context.xml

Tiles definitions file is in /WEB-INF/tiles/definitions/tile-definition.xml

Template files are in /WEB-INF/tiles/template

Testing Procedure:

1. Please check the JSP view files in /WEB-INF/views, the file only contains the body section.
2. Load any page and view the source. You can see that the HTML generated contains the header and footer as defined in the definitions.xml for each view.

4. Internationalization (**Long Ma**)

The configuration is in /WEB-INF/config/servlet-context.xml under bean localeResolver.

Message properties files are in /main/resources/messages.properties and

/main/resources/messages_zh_CN.properties

Testing Procedure:

1. Go to the Home page
2. Click on the language/internationalization links (EN | CN) on the top right corner. This link is also available in the admin, doctor and patient

dashboards.

5. JPA - Hibernate

The JPA configurations can be found in /WEB-INF/config/application-context.xml

All domain classes are annotated for persistence in *java.mum.waa.fd.app.domain*

Controller: AppointmentController (**Toan Quach**)

Method: showRegistrationAppointment - *RequestMethod.POST*

Function: Save new appointment from patient with validations declared in the Appointment domain class in *java.mum.waa.fd.app.domain*

Testing Procedure:

1. Click Login button on Home page
2. Login as: Email = toanqc@gmail.com Password = toanqc
3. On Patient dashboard, click "Schedule an appointment" link
4. Complete the form then Submit. This will save the information into the database using the CRUD repository (AppointmentRepository) in *java.mum.waa.fd.app.repository*.

Controller: PatientController (**Toan Quach**)

Method: showPatientHome

Function: Retrieve and shows the patient appointments

Testing Procedure:

1. Click Login button on Home page
2. Login as: Email = toanqc@gmail.com Password = toanqc
3. On the dashboard you can see available appointments. It calls the methods *getUpcomingAppointment* and *getOverdueAppointment* in the *PatientServiceImpl (java.mum.waa.fd.app.service.impl)*.

Controller: AdminController (**Mark Pit**)

Method: saveDoctor

Function: Save new doctor information from admin with validations declared in the Doctor domain class in *java.mum.waa.fd.app.domain*

Testing Procedure:

1. Click Login button on Home page
2. Login as: Email = admin@gmail.com Password = admin
3. On dashboard, click "Add Doctor" link
4. Complete the form then Submit. This will save the information into the database using the CRUD repository (DoctorRepository) in *java.mum.waa.fd.app.repository*.

6. Security (**Toan Quach**)

Configurations are in /WEB-INF/config/security-context.xml

Controller: LoginController

Methods: showLogin - loads the login form

loginSuccess - redirects to corresponding user

home page by Role after successful login

loginFailed - loads the login form with fail message

logout - loads home page after successful logout

Testing Procedure:

1. Go to the home page then click on Login button
2. Enter an random/invalid Email and Password then submit. You should see an incorrect email/password message.

3. Enter a valid email and password:

Admin user:

email = admin@gmail.com

password = admin

Doctor:

email = doctor@gmail.com

password = doctor

Patient:

email = toanqc@gmail.com

password = toanqc

You will be redirected to the corresponding dashboard.

4. Click "logout" on the top right corner under the "Welcome" message to log-out of the system.