T809 Final Documentation

Table of Contents

Description

iTrust is a web-based medical records system that enables patients and health care providers (HCP) to keep records of their medical data and operational data. The scope of this project focuses on the obstetric health care module and requires the team to implement four pre-determined use cases as well as a custom use case. A summary of the objectives for each use case can be found in the following table:

Use Case	Objective
UC93	An HCP specialized in obstetric care shall be able to initialize and establish a registered patient’s obstetric status. He/she shall be able to document the patient’s obstetric record.

UC94	An HCP specialized in obstetric care shall be able to schedule, document, and edit an obstetric office visit.
UC95	Any HCP shall be able to generate a comprehensive obstetric record of an obstetric patient in case of emergency.
UC96	Any HCP shall be able to view an obstetric patient’s obstetric record during a childbirth hospital visit. The HCP shall be able to document the delivery and the newborn shall be initialized as a new patient in the system.
UC Custom	Any HCP shall be able to post on an internal bulletin board as well as comment on each other’s bulletin post. Any HCP shall be able to edit or delete their own posts or comments.

Architecture and Design

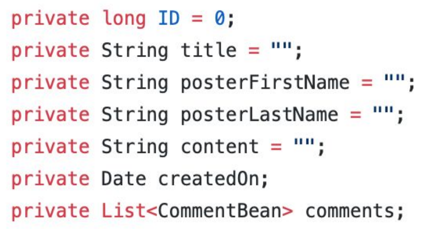
Bean

A bean is a representation of a table in the database. Its main function is to store data (as a tuple) regarding the records in the database table. In order to achieve data hiding, all variables are set to private and there are specific getters and setters for data retrieval and assignment respectively.

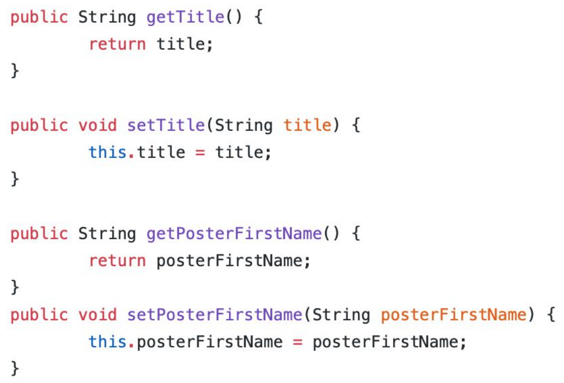
Using our Custom Use Case as an example, the newly created **Bulletin** table consists of the following attributes: Title, PosterFirstName, PosterLastName, Content, and CreatedOn. Thus, in our **BulletinBoardBean**, we have private variables of the same name (and data type) to store such data in the Java backend.

Beans are be used by the Loaders and DAOs to store the data for their accessing database (e.g. getComment() in **BulletinBoardDAO** saves a **CommentBean** for return).

The following code snippet demonstrates the template of bean implementation in iTrust.



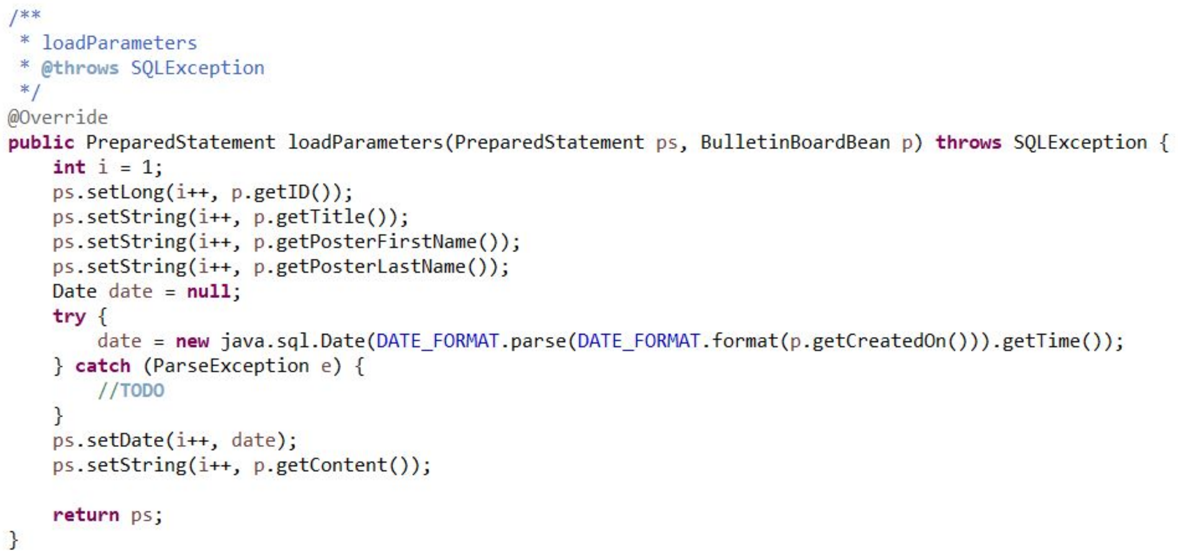
```
private long ID = 0;
private String title = "";
private String posterFirstName = "";
private String posterLastName = "";
private String content = "";
private Date createdOn;
private List<CommentBean> comments;
```



```
public String getTitle() {
        return title;
}

public void setTitle(String title) {
        this.title = title;
}

public String getPosterFirstName() {
        return posterFirstName;
}
public void setPosterFirstName(String posterFirstName) {
        this.posterFirstName = posterFirstName;
}
```

Loader

A Loader class loads information to and from a Bean. It is used mainly by the DAO, both to prepare statements for use in database queries and to load in resulting information from the queries themselves. For example, the **loadParameters** function in **BulletinBoardLoader.java** takes in a **BulletinBoardBean**, and loads in all of the necessary data for bulletin board creation into a PreparedStatement.



```
/**
 * loadParameters
 * @throws SQLException
 */
@Override
public PreparedStatement loadParameters(PreparedStatement ps, BulletinBoardBean p) throws SQLException {
    int i = 1;
    ps.setLong(i++, p.getID());
    ps.setString(i++, p.getTitle());
    ps.setString(i++, p.getPosterFirstName());
    ps.setString(i++, p.getPosterLastName());
    Date date = null;
    try {
        date = new java.sql.Date(DATE_FORMAT.parse(DATE_FORMAT.format(p.getCreatedOn())).getTime());
    } catch (ParseException e) {
        //TODO
    }
    ps.setDate(i++, date);
    ps.setString(i++, p.getContent());

    return ps;
}
```

The parameters must be set in the exact order required by the DAO.

Form

The Form class stores input fields sent by requests. It is a bean that only stores parameters temporarily for later processing, and it does not have a related **DAO**.

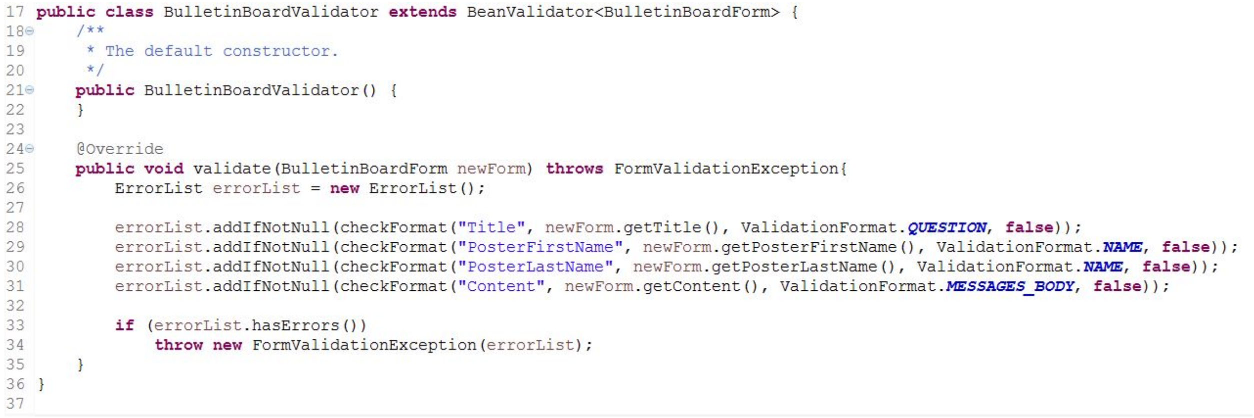
We use it to parse request parameters in jsp files, and inside jsp files, create **action** objects. **Action** objects have methods that take request parameters as variables and create a new form object to store the values. The form object will later be validated by **validators**.

A form example:

```
public class CommentBean implements Serializable, Comparable<CommentBean> {

        private long ID = 0;
        private long bulletinBoardID = 0;
        private String posterFirstName = "";
        private String posterLastName = "";
        private String text = "";
        private Date createdOn;

        public long getID() {
                return ID;
        }
        public void setID(long ID) {
                this.ID = ID;
        }

        public long getBulletinBoardID() {
                return bulletinBoardID;
        }
        public void setBulletinBoardID(long ID) {
                this.bulletinBoardID = ID;
        }

        public String getPosterFirstName() {
                return posterFirstName;
        }
        public void setPosterFirstName(String posterFirstName) {
                this.posterFirstName = posterFirstName;
        }

        .
        .
        .
}
```

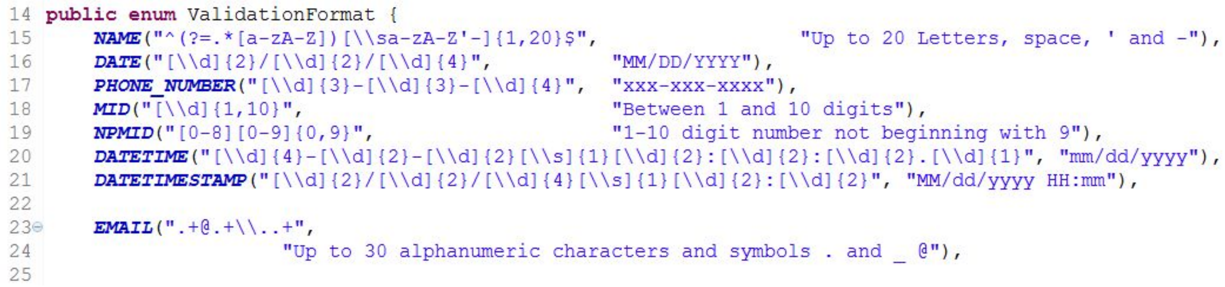
Validator

The validator class is used to check that input is in the correct format. It is used for the form class to validate user input. The validator consists of two files. The first is the class-specific Validator shown in Figure 1 where you create and errorlist and run through each input you want to validate. The second is the ValidationFormat shown in Figure 2 that contains all of the regular expressions used to check input in the class-specific validator class.



```
public class BulletinBoardValidator extends BeanValidator<BulletinBoardForm> {
    /**
     * The default constructor.
     */
    public BulletinBoardValidator() {
    }

    @Override
    public void validate(BulletinBoardForm newForm) throws FormValidationException{
        ErrorList errorList = new ErrorList();

        errorList.addIfNotNull(checkFormat("Title", newForm.getTitle(), ValidationFormat.QUESTION, false));
        errorList.addIfNotNull(checkFormat("PosterFirstName", newForm.getPosterFirstName(), ValidationFormat.NAME, false));
        errorList.addIfNotNull(checkFormat("PosterLastName", newForm.getPosterLastName(), ValidationFormat.NAME, false));
        errorList.addIfNotNull(checkFormat("Content", newForm.getContent(), ValidationFormat.MESSAGES_BODY, false));

        if (errorList.hasErrors())
            throw new FormValidationException(errorList);
    }
}
```

Figure 1: Class-specific validator



```
public enum ValidationFormat {
    NAME("^(?=.*[a-zA-Z])[\\sa-zA-Z'-]{1,20}$",                "Up to 20 Letters, space, ' and -"),
    DATE("[\\d]{2}/[\\d]{2}/[\\d]{4}",          "MM/DD/YYYY"),
    PHONE_NUMBER("[\\d]{3}-[\\d]{3}-[\\d]{4}",  "xxx-xxx-xxxx"),
    MID("[\\d]{1,10}",                          "Between 1 and 10 digits"),
    NPMID("[0-8][0-9]{0,9}",                    "1-10 digit number not beginning with 9"),
    DATETIME("[\\d]{4}-[\\d]{2}-[\\d]{2}[\\s]{1}[\\d]{2}:[\\d]{2}:[\\d]{2}.[\\d]{1}", "mm/dd/yyyy"),
    DATETIMESTAMP("[\\d]{2}/[\\d]{2}/[\\d]{4}[\\s]{1}[\\d]{2}:[\\d]{2}", "MM/dd/yyyy HH:mm"),

    EMAIL(".+@.+\\..+",
                    "Up to 30 alphanumeric characters and symbols . and _ @"),

```

Figure 2: Regular expressions

DAO

“DAO” stands for Data Access Object. It is a class that contains methods to access data directly in the database (mysql in this project). For example, calling **addComment** in **BulletinBoardDAO**, with a **CommentBean** object as the parameter, will add a new comment into the database. The idea of implementation of a DAO is to hard code a piece of sql query in a function, and use a **loader** (described in another section) to parse the target object to get the desire parameters for the query, and then execute the query. DAOs are used in many different kinds of files in iTrust project, but they will be mainly used in **action** files.

The following code is a very typical way to write functions in DAO files. The **loader** we use here is a **CommentLoader**, and the results of the loader will be plugged into the question marks in the query.

```
public long addComment(CommentBean newComment) throws DBException {
        try (Connection conn = factory.getConnection();
                PreparedStatement stmt = commentLoader.loadParameters(conn.prepareStatement(
                        "INSERT INTO comments (BulletinBoardID, PosterFirstName, PosterLastName, "
                        + "Text, CreatedOn) VALUES (?, ?, ?, ?, ?)"),
                        newComment)) {
                stmt.executeUpdate();
                return DBUtil.getLastInsert(conn);
        } catch (SQLException e) {
                System.out.println(e.getMessage());
                throw new DBException(e);
        }
}
```

You have to add a function into the **DAOFactory.java** for your newly created DAO.

```
public BulletinBoardDAO getBulletinBoardDAO() {
        return new BulletinBoardDAO(this);
}
```

Action

An “Action” is a Controller object that communicates data back and forth between the View objects (JSPs) and the Model objects (Beans, DAOs, and others). For instance, JSPs can get a list of **BulletinBoardBean** with all of the data in the database stored inside by calling **getAllBulletinBoards** in **ViewBulletinBoardAction**. JSPs can also call **addBulletinBoard** in **AddBulletinBoardAction** and provide a **BulletinBoardBean** as the parameter to add a new bulletin board post.

The following snippets of code demonstrate the implementation of action objects.

For **getAllBulletinBoards** in **ViewBulletinBoardAction,** we simply call **getAllBulletinBoards** in **BulletinBoardDAO** and return the list we get.

```
public List<BulletinBoardBean> getAllBulletinBoards() throws ITrustException {
        return bulletinBoardDAO.getAllBulletinBoards();
    }
```

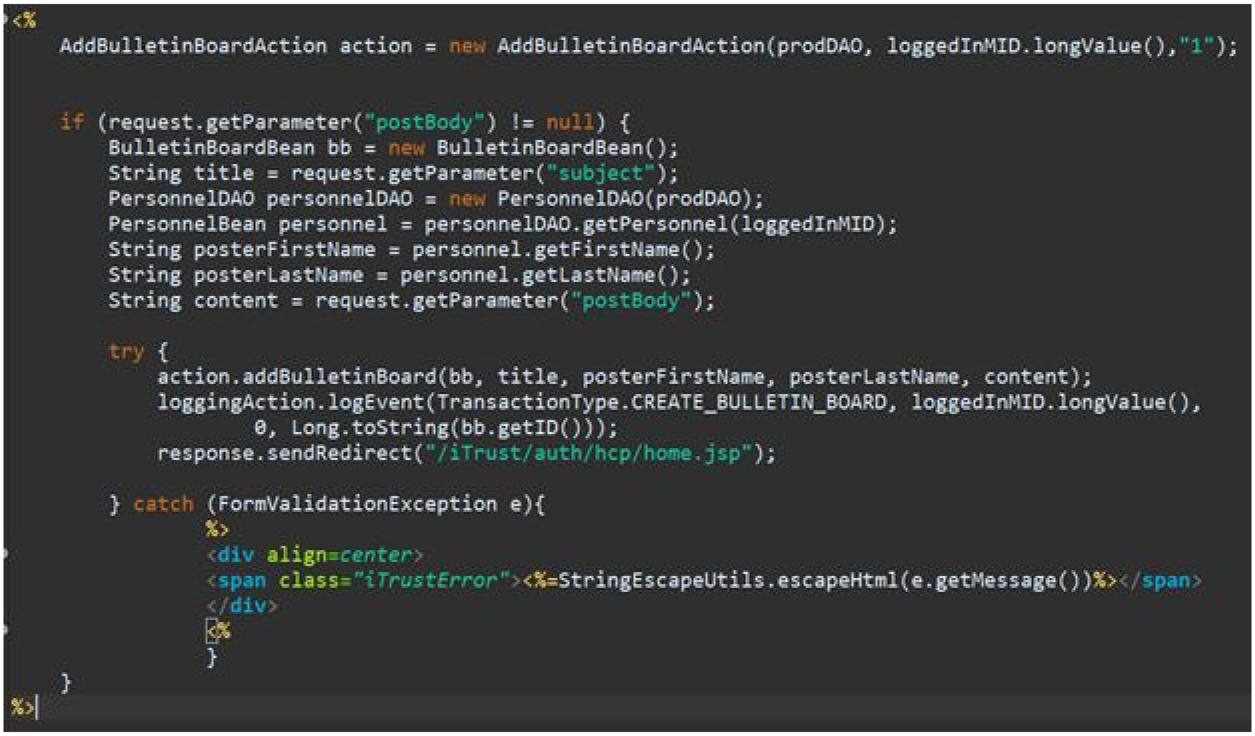
When implementing **addBulletinBoard** in **AddBulletinBoardAction**, we first store input data in a form and validate it by a validator, then we call **addBulletinBoard** function in **BulletinBoardDAO** and give a **BulletinBoardBean** object as the parameter to add a new bulletin board post into the database.

```
public long addBulletinBoard(BulletinBoardBean newBulletinBoard, String title, String
osterFirstName, String posterLastName, String content) throws ITrustException,
FormValidationException {
        SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy HH:mm");
        BulletinBoardForm form = new BulletinBoardForm(title, posterFirstName,
posterLastName, content);
        bValidator.validate(form);

        newBulletinBoard.setTitle(title);
        newBulletinBoard.setPosterFirstName(posterFirstName);
        newBulletinBoard.setPosterLastName(posterLastName);
        Date CreatedOn = new Date();
        newBulletinBoard.setCreatedOn(CreatedOn);
        newBulletinBoard.setContent(content);

        return bulletinBoardDAO.addBulletinBoard(newBulletinBoard);
    }
```

JSP

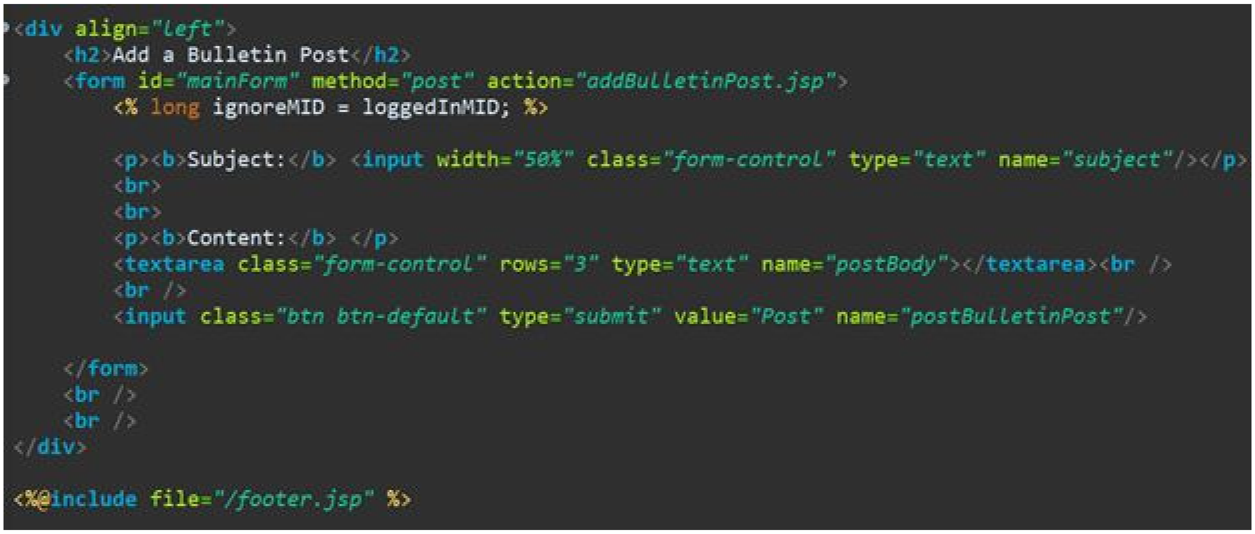
A JavaServer Page (JSP) is an HTML-based file that creates web pages with a servlet container in a compatible web server, which is the Apache Web Server in this project. A JSP can be perceived as a high-level abstraction of Java servlets from an architectural perspective, and it is usually used as the view component to communicate with JavaBeans and Java servlets in an MVC design.

JSPs are employed in this project to deliver static web markup content, and they also creates **action** objects to load potential user input into the corresponding JavaBean object. The following snippet of code demonstrates how **addBulletinPost.jsp** creates an **AddBulletinBoardAction** object upon request, and how it feeds the user input into a **BulletinBoardBean** by reading the content of each web element.



```
<%
    AddBulletinBoardAction action = new AddBulletinBoardAction(prodDAO, loggedInMID.longValue(),"1");


    if (request.getParameter("postBody") != null) {
        BulletinBoardBean bb = new BulletinBoardBean();
        String title = request.getParameter("subject");
        PersonnelDAO personnelDAO = new PersonnelDAO(prodDAO);
        PersonnelBean personnel = personnelDAO.getPersonnel(loggedInMID);
        String posterFirstName = personnel.getFirstName();
        String posterLastName = personnel.getLastName();
        String content = request.getParameter("postBody");

        try {
            action.addBulletinBoard(bb, title, posterFirstName, posterLastName, content);
            loggingAction.logEvent(TransactionType.CREATE_BULLETIN_BOARD, loggedInMID.longValue(),
                    0, Long.toString(bb.getID()));
            response.sendRedirect("/iTrust/auth/hcp/home.jsp");

        } catch (FormValidationException e){
                %>
                <div align=center>
                <span class="iTrustError"><%=StringEscapeUtils.escapeHtml(e.getMessage())%></span>
                </div>
                <%
                }
    }
%>
```

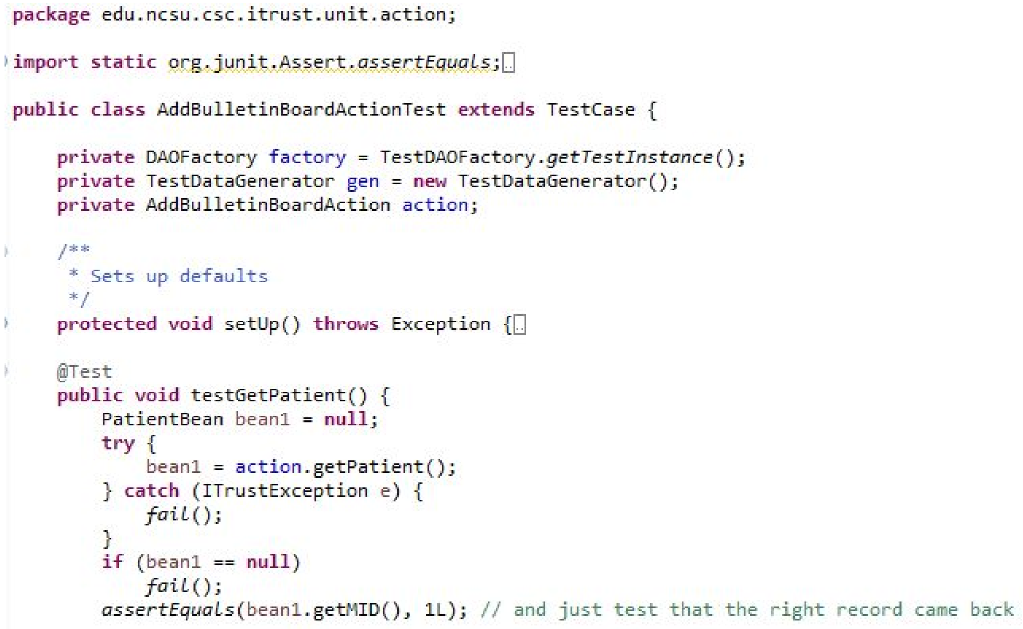
The following snippet of code presents the HTML lines in **addBulletinPost.jsp,** which specifies the markup of the UI for adding a new bulletin post.



```
<div align="left">
    <h2>Add a Bulletin Post</h2>
    <form id="mainForm" method="post" action="addBulletinPost.jsp">
        <% long ignoreMID = loggedInMID; %>

        <p><b>Subject:</b> <input width="50%" class="form-control" type="text" name="subject"/></p>
        <br>
        <br>
        <p><b>Content:</b> </p>
        <textarea class="form-control" rows="3" type="text" name="postBody"></textarea><br />
        <br />
        <input class="btn btn-default" type="submit" value="Post" name="postBulletinPost"/>

    </form>
    <br />
    <br />
</div>

<%@include file="/footer.jsp" %>
```

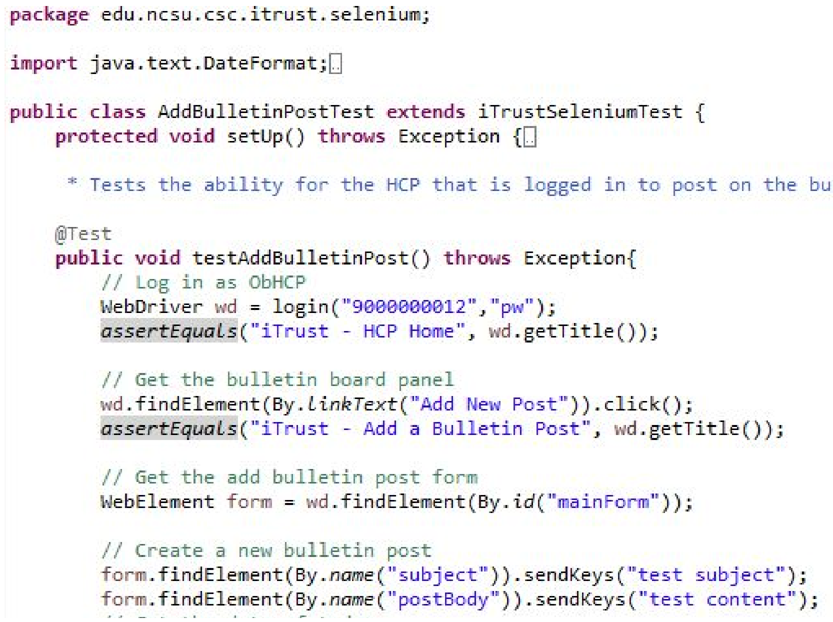
Tests

Test classes are used to check if the code produces the correct behavior. There are two types of tests in this project: unit tests and UI tests. See the Appendix for how to run these tests. Each unit test will test one java class file for at least 80% code coverage.



```
package edu.ncsu.csc.itrust.unit.action;

import static org.junit.Assert.assertEquals;□

public class AddBulletinBoardActionTest extends TestCase {

    private DAOFactory factory = TestDAOFactory.getTestInstance();
    private TestDataGenerator gen = new TestDataGenerator();
    private AddBulletinBoardAction action;

    /**
     * Sets up defaults
     */
    protected void setUp() throws Exception {□

    @Test
    public void testGetPatient() {
        PatientBean bean1 = null;
        try {
            bean1 = action.getPatient();
        } catch (ITrustException e) {
            fail();
        }
        if (bean1 == null)
            fail();
        assertEquals(bean1.getMID(), 1L); // and just test that the right record came back
```

To test the UI, we used Selenium tests to simulate a user's behavior.



```
package edu.ncsu.csc.itrust.selenium;

import java.text.DateFormat;□

public class AddBulletinPostTest extends iTrustSeleniumTest {
    protected void setUp() throws Exception {□

     * Tests the ability for the HCP that is logged in to post on the bulletin board□

    @Test
    public void testAddBulletinPost() throws Exception{
        // Log in as ObHCP
        WebDriver wd = login("9000000012","pw");
        assertEquals("iTrust - HCP Home", wd.getTitle());

        // Get the bulletin board panel
        wd.findElement(By.linkText("Add New Post")).click();
        assertEquals("iTrust - Add a Bulletin Post", wd.getTitle());

        // Get the add bulletin post form
        WebElement form = wd.findElement(By.id("mainForm"));

        // Create a new bulletin post
        form.findElement(By.name("subject")).sendKeys("test subject");
        form.findElement(By.name("postBody")).sendKeys("test content");
```

Appendix

Special Note: UC 94 - Ultrasound Image

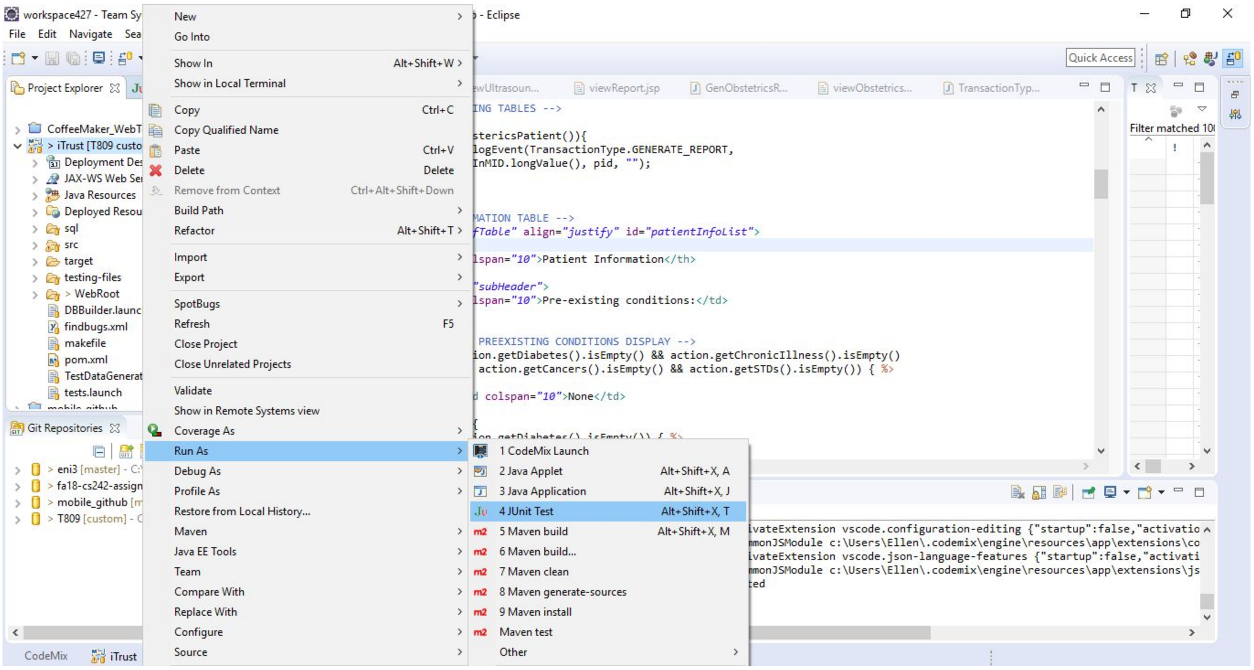
We implemented the sub-flow by storing image files as a BLOB object in the database (datatype being a LONGBLOG to store large, high-quality files). The image would first be converted to a byte array, then be stored to the database just like how a regular String would. For display, the DAO simply retrieves the byte array (again like any other regular datatype) then the array would be converted to an image depending on its content-type. We implemented a DisplayServlet to display the ultrasound image (in essence transferring the image from the backend to the frontend).
Please note in order for MySQL to store large byte array (file), the MySQL server configuration needs to be modified by setting `max_allowed_packet = 809M`.

How to Run Tests

Open iTrust in Eclipse. In order to find test coverage, EclEmma must be installed.

Run All Tests

To run all tests, right click on the iTrust directory in the Project Explorer, hover over the Run As tab, and click on the JUnit Test option as shown in Figure 3.



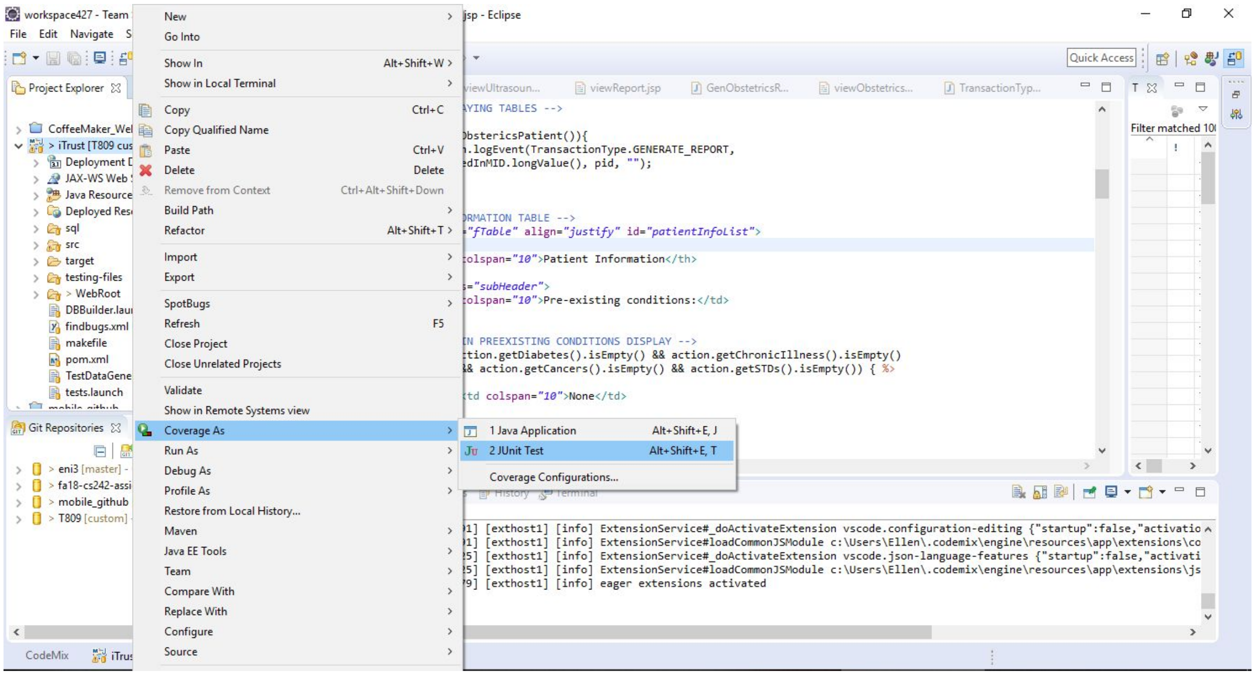
```
ING TABLES -->

stericsPatient()){
logEvent(TransactionType.GENERATE_REPORT,
InMID.longValue(), pid, "");



MATION TABLE -->
fTable" align="justify" id="patientInfoList">

lspan="10">Patient Information</th>

"subHeader">
lspan="10">Pre-existing conditions:</td>


 PREEXISTING CONDITIONS DISPLAY -->
ion.getDiabetes().isEmpty() && action.getChronicIllness().isEmpty()
 action.getCancers().isEmpty() && action.getSTDs().isEmpty()) { %>

d colspan="10">None</td>

{
ion.getDiabetes().isEmpty()) { %>
```

```
ivateExtension vscode.configuration-editing {"startup":false,"activatio
mmonJSModule c:\Users\Ellen\.codemix\engine\resources\app\extensions\co
ivateExtension vscode.json-language-features {"startup":false,"activati
mmonJSModule c:\Users\Ellen\.codemix\engine\resources\app\extensions\js
ted
```

Figure 3: Run all tests

Find Coverage for All Tests

If EclEmma is installed, the option Coverage As will be available in the menu when you right click the iTrust directory in the Project Explorer. As shown in Figure 4, click on the JUnit Test option.



```
AYING TABLES -->

ObstericsPatient()){
n.logEvent(TransactionType.GENERATE_REPORT,
edInMID.longValue(), pid, "");



ORMATION TABLE -->
="fTable" align="justify" id="patientInfoList">

colspan="10">Patient Information</th>

s="subHeader">
colspan="10">Pre-existing conditions:</td>


IN PREEXISTING CONDITIONS DISPLAY -->
ction.getDiabetes().isEmpty() && action.getChronicIllness().isEmpty()
&& action.getCancers().isEmpty() && action.getSTDs().isEmpty()) { %>

<td colspan="10">None</td>
```

```
01] [exthost1] [info] ExtensionService#_doActivateExtension vscode.configuration-editing {"startup":false,"activatio
01] [exthost1] [info] ExtensionService#loadCommonJSModule c:\Users\Ellen\.codemix\engine\resources\app\extensions\co
25] [exthost1] [info] ExtensionService#_doActivateExtension vscode.json-language-features {"startup":false,"activati
25] [exthost1] [info] ExtensionService#loadCommonJSModule c:\Users\Ellen\.codemix\engine\resources\app\extensions\js
79] [exthost1] [info] eager extensions activated
```

Figure 4: Coverage of all tests

Run Specific Unit Tests or Coverage

The path is iTrust/src/test/java/edu/ncsu/csc/itrust/unit/

Alternatively, navigate to the unit folder in the Project Explorer as shown in Figure 6.

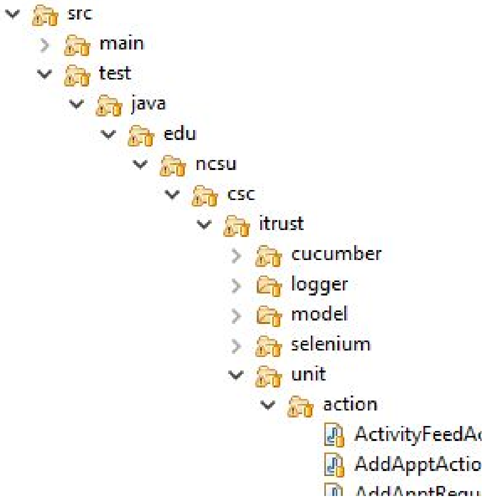
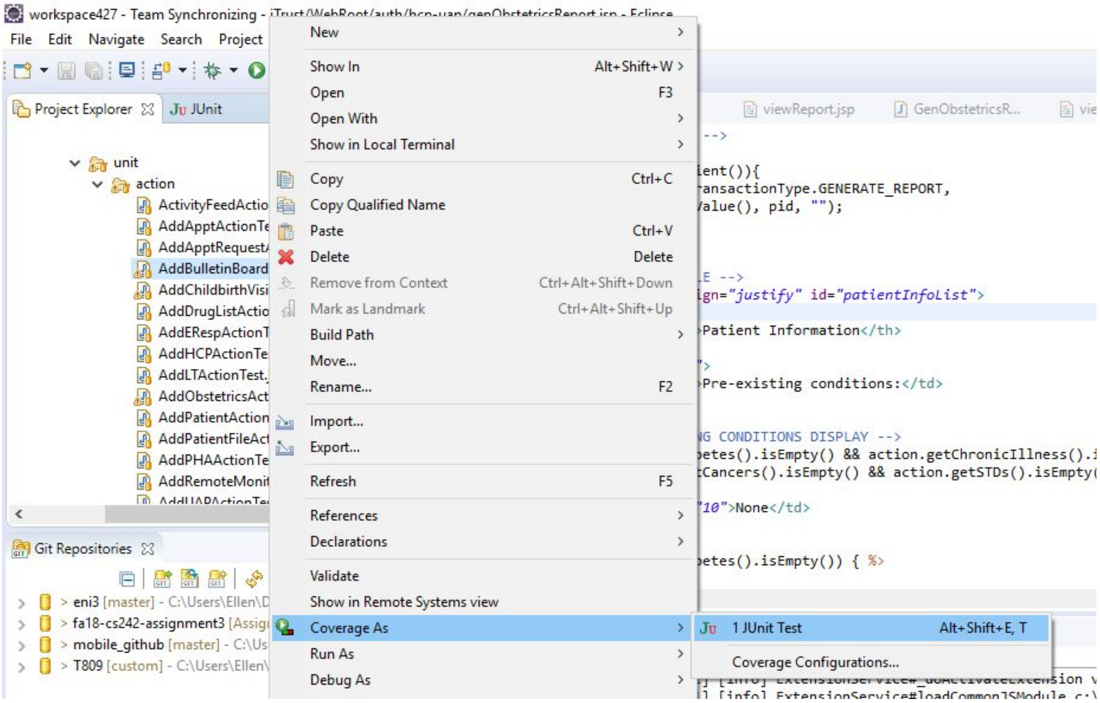


Figure 6: unit tests folder

As shown in Figure 7, right click on the test you want to run and select either Run As or Coverage As and then JUnit Test. This example uses Coverage As.



```
-->
ient()){
ransactionType.GENERATE_REPORT,
/alue(), pid, "");

E -->
ign="justify" id="patientInfoList">
Patient Information</th>

">
Pre-existing conditions:</td>

G CONDITIONS DISPLAY -->
etes().isEmpty() && action.getChronicIllness().i
Cancers().isEmpty() && action.getSTDs().isEmpty(

"10">None</td>

etes().isEmpty()) { %>
```

Figure 7: coverage as specific unit test

Run Specific Selenium Tests

The path is iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/
Alternatively, navigate to the selenium folder as shown in Figure 8.

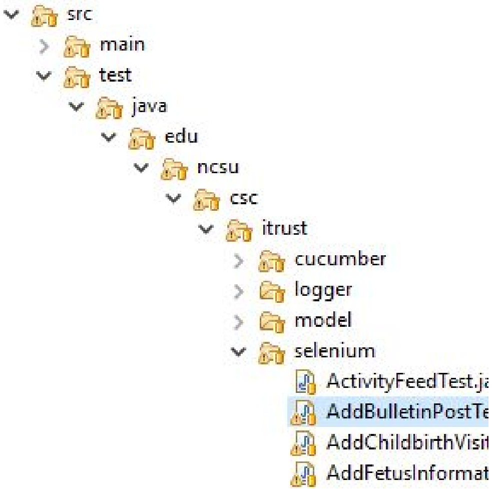
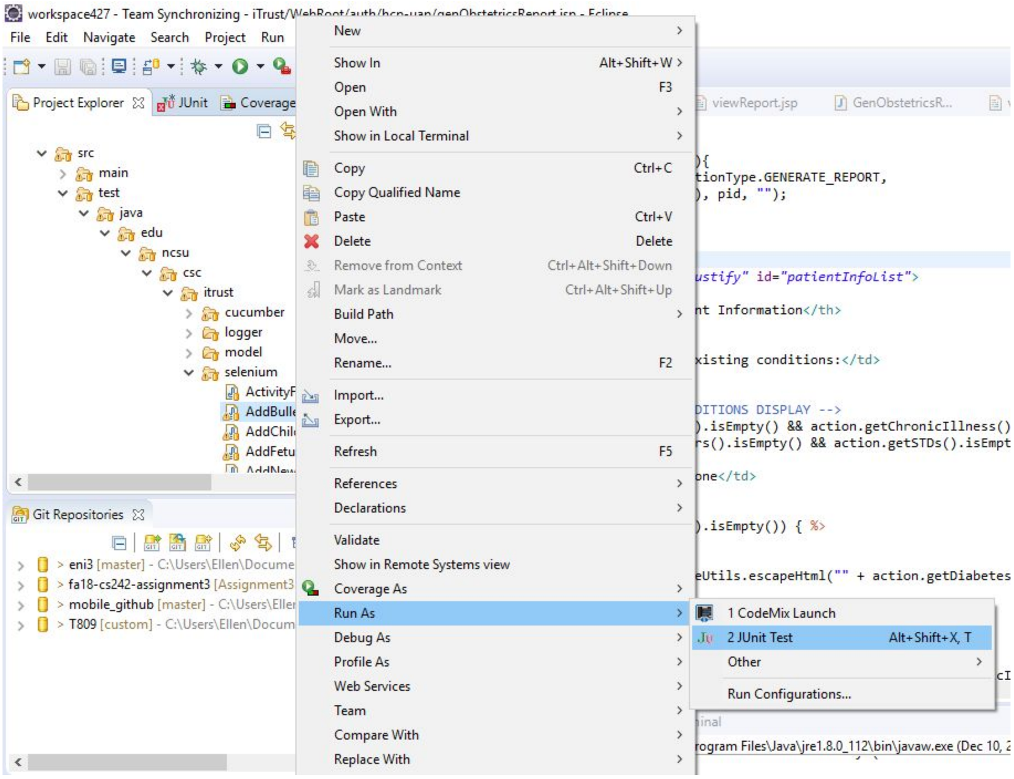


Figure 8: Selenium tests folder

As shown in Figure 9, right click on the test you want to run and select Run As and JUnit test.



```
){
tionType.GENERATE_REPORT,
), pid, "");

ustify" id="patientInfoList">

nt Information</th>

xisting conditions:</td>

DITIONS DISPLAY -->
).isEmpty() && action.getChronicIllness()
rs().isEmpty() && action.getSTDs().isEmpt

one</td>

).isEmpty()) { %>

eUtils.escapeHtml("" + action.getDiabetes

cI
inal
rogram Files\Java\jre1.8.0_112\bin\javaw.exe (Dec 10, 2
```

Figure 9: Run specific Selenium tests

Files Modified and Added

A iTrust/WebRoot/auth/hcp-uap/addChildbirthVisit.jsp
A iTrust/WebRoot/auth/hcp-uap/addFetusInformation.jsp
A iTrust/WebRoot/auth/hcp-uap/addNewborns.jsp
A iTrust/WebRoot/auth/hcp-uap/addObstetricsRecord.jsp
A iTrust/WebRoot/auth/hcp-uap/addObstetricsVisit.jsp
A iTrust/WebRoot/auth/hcp-uap/addUltrasoundRecord.jsp
A iTrust/WebRoot/auth/hcp-uap/editChildbirthVisit.jsp
A iTrust/WebRoot/auth/hcp-uap/editNewborn.jsp
A iTrust/WebRoot/auth/hcp-uap/editObstetricsVisit.jsp
A iTrust/WebRoot/auth/hcp-uap/genObstetricsReport.jsp
A iTrust/WebRoot/auth/hcp-uap/viewChildbirthVisit.jsp
A iTrust/WebRoot/auth/hcp-uap/viewObstetrics.jsp
A iTrust/WebRoot/auth/hcp-uap/viewObstetricsRecords.jsp
A iTrust/WebRoot/auth/hcp-uap/viewObstetricsVisit.jsp
A iTrust/WebRoot/auth/hcp-uap/viewUltrasound.jsp
A iTrust/WebRoot/auth/hcp-uap/viewUltrasoundRecords.jsp
A iTrust/WebRoot/auth/hcp/BulletinPost.jsp
A iTrust/WebRoot/auth/hcp/addBulletinPost.jsp
A iTrust/WebRoot/auth/hcp/editBulletinPost.jsp
M iTrust/WebRoot/auth/hcp/home.jsp
M iTrust/WebRoot/auth/hcp/menu.xhtml
A iTrust/WebRoot/auth/hcp/viewBulletinPost.jsp

```
M    iTrust/WebRoot/auth/hcp/viewMyAppts.jsp
M    iTrust/WebRoot/auth/patient/viewMyAppts.jsp
M    iTrust/sql/createTables.sql
M    iTrust/sql/data/deleteFromAllTables.sql
A    iTrust/sql/data/hcpOb.sql
M    iTrust/sql/data/patient1.sql
A    iTrust/sql/data/uc93.sql
A    iTrust/sql/data/uc94.sql
A    iTrust/sql/data/uc95.sql
A    iTrust/sql/data/uc96.sql
A    iTrust/sql/data/ucCustom.sql
M    iTrust/sql/dropTables.sql
A    iTrust/src/main/edu/ncsu/csc/itrust/action/AddBulletinBoardAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/AddChildbirthVisitAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/AddObstetricsAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/AddObstetricsVisitAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/AddUltrasoundAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/DisplayServlet.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/EditBulletinBoardAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/EditChildbirthVisitAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/EditObstetricsVisitAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/GenObstetricsReportAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/GetNextVisitAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/ViewBulletinBoardAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/ViewChildbirthVisitAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/ViewObstetricsAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/ViewObstetricsVisitAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/action/ViewUltrasoundAction.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/BulletinBoardBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/ChildbirthVisitBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/CommentBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/DeliveryRecordBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/FetusBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/ObstetricsBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/ObstetricsVisitBean.java
M    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/PatientBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/PregnancyBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/UltrasoundBean.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/BulletinBoardForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/ChildbirthVisitForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/CommentForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/DeliveryRecordForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/FetusForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/ObstetricsForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/ObstetricsVisitForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/PregnancyForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/forms/UltrasoundForm.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/BulletinBoardLoader.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/ChildbirthVisitLoader.java
A    iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/CommentLoader.java
```

```
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/DeliveryRecordLoader.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/FetusLoader.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/ObstetricsLoader.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/ObstetricsVisitLoader.java
M     iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/PatientLoader.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/PregnancyLoader.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/beans/loaders/UltrasoundLoader.java
M     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/DAOFactory.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/mysql/BulletinBoardDAO.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/mysql/ChildbirthVisitDAO.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/mysql/ObstetricsDAO.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/mysql/ObstetricsVisitDAO.java
M     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/mysql/PatientDAO.java
M     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/mysql/TransactionDAO.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/dao/mysql/UltrasoundDAO.java
M     iTrust/src/main/edu/ncsu/csc/itrust/model/old/enums/TransactionType.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/BulletinBoardValidator.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/ChildbirthVisitValidator.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/CommentValidator.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/DeliveryRecordValidator.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/FetusValidator.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/ObstetricsValidator.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/ObstetricsVisitValidator.java
A     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/UltrasoundValidator.java
M     iTrust/src/main/edu/ncsu/csc/itrust/model/old/validate/ValidationFormat.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/model/bulletinboard/BulletinBoardFormTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/AddBulletinPostTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/AddChildbirthVisitTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/AddFetusInformationTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/AddNewbornsTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/AddObstetricRecordTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/AddObstetricVisitTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/AddUltrasoundRecordTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/EditBulletinPostTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/EditChildbirthVisitTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/EditNewbornTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/EditObstetricVisitTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/GenObstetricsReportTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/ViewBulletinPostTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/ViewChildbirthVisitTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/ViewObstetricRecordTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/ViewObstetricVisitTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/ViewUltrasoundRecordsTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/selenium/ViewUltrasoundTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/AddBulletinBoardActionTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/AddChildbirthVisitActionTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/AddObstetricsActionTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/AddUltrasoundActionTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/EditBulletinBoardActionTest.java
A     iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/EditChildbirthVisitActionTest.java
```

```
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/GenObstetricsReportActionTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/GetNextVisitActionTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/ViewBulletinBoardActionTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/ViewChildbirthVisitActionTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/ViewObstetricsActionTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/ViewObstetricsVisitActionTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/action/ViewUltrasoundActionTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/BulletinBoardBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/BulletinBoardLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/ChildbirthVisitBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/ChildbirthVisitLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/CommentBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/CommentLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/DeliveryRecordBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/DeliveryRecordLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/FetusBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/FetusLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/ObstetricsBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/ObstetricsLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/ObstetricsVisitBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/ObstetricsVisitLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/PregnancyBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/PregnancyLoaderTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/bean/UltrasoundBeanTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/dao/bulletinboard/BulletinBoardDAOTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/dao/childbirth/ChildbirthVisitDAOTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/dao/obstetrics/ObstetricsDAOTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/dao/obstetrics/ObstetricsVisitDAOTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/dao/ultrasound/UltrasoundDAOTest.java
M	 iTrust/src/test/java/edu/ncsu/csc/itrust/unit/datagenerators/TestDataGenerator.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/childbirth/ChildbirthVisitFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/comment/CommentFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/delivery/DeliveryRecordFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/fetus/FetusFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/obstetrics/ObstetricsFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/obstetrics/ObstetricsVisitFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/pregnancy/PregnancyFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/model/ultrasound/UltrasoundFormTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/BulletinBoardValidatorTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/ChildbirthVisitValidatorTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/CommentValidatorTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/DeliveryRecordValidatorTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/FetusValidatorTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/ObstetricsValidatorTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/ObstetricsVisitValidatorTest.java
A	iTrust/src/test/java/edu/ncsu/csc/itrust/unit/validate/bean/UltrasoundValidatorTest.java
```