

# Project Report

## Login Module

### *Description:*

login.jsp:

- This page displays a username/password field for the user to enter. When users hit the submit button, login.jsp stores the username and password. It then connects to the database and executes sql (1) to get the true password for the corresponding username. If the user has entered the correct password, then they get redirected to menu.jsp. Otherwise, an error message appears saying they have entered the incorrect user details.

logout.jsp:

- Handles the logout functionality. It removes the current session attributes.

modifyAccount.jsp:

- This page is selected from the main menu and allows the user to modify their personal information and/or the password. The JSP page gets the user data from the current session user's username by executing sql statements (2.1) and (2.2). A form is displayed for the user containing this information, any of which they can change. Upon hitting the submit button, the JSP redirects to ModifyAccountCheck.java.

ModifyAccountCheck.java:

- This servlet checks over the information changes and performs updates accordingly. An error message is returned if the user enters an already existing email (unique constraint) or if they incorrectly enter their old password. When the updates are done, the servlet redirects to the main menu.

### *SQL Statements:*

- (1) select password, class, person\_id from users where user\_name = "" + username + ""
- (2.1) select \* from persons where person\_id in (select person\_id from users where user\_name = "" + username + "")
- (2.2) select password from users where user\_name = "" + username + ""
- (3.1) update persons set first\_name = "" + fname + "" where person\_id = " + curr\_id
- (3.2) update persons set last\_name = "" + lname + "" where person\_id = " + curr\_id
- (3.3) update persons set address = "" + address + "" where person\_id = " + curr\_id
- (3.4a) select count(\*) from persons where person\_id != " + curr\_id + " and email = "" + email + ""
- (3.4b) update persons set email = "" + email + "" where person\_id = " + curr\_id
- (3.5) update persons set phone = "" + phone + "" where person\_id = " + curr\_id
- (3.6a) select password from users where person\_id = " + curr\_id

- (3.6b) update users set password = "" + new\_password + "" where person\_id = "" + curr\_id

### ***Reuse Statements:***

login.jsp:

- Modified from: <http://webdocs.cs.ualberta.ca/~yuan/servlets/login.jsp>

## **User Management Module**

### ***Description:***

userManagementMenu.jsp:

- This page displays two forms, both sets of radio buttons with a submit button. The two forms differentiate between each other based on their actions. The first set of buttons is used to select between existing users, persons or family doctors for which the administrator wishes to search for and update. The second set of radio buttons is used to select and create a new user, person, or family doctor. If the search button is pressed, the page is redirected to searchUserManagement.jsp and if the create button is pressed, the page is directed to createEntry.jsp. If the user's session indicates that they are not an administrator, an access denied error message is displayed and the user is sent back to the main menu.

searchUserManagement.jsp:

- The page will display different options depending on the radio button selection made in userManagementMenu.jsp. If no selection is made, it displays the search form from userManagementMenu.jsp and continues to do so until the user makes a selection. All forms displayed will direct the user to the updateEntry.jsp page. If users is selected in userManagementMenu.jsp, a username field is displayed to search for the user by username. If persons is selected, a person ID field is displayed and if family doctors is selected, a doctor id and person id field are displayed. If the user's session indicates that they are not an administrator, an access denied error message is displayed and the user is sent back to the main menu.

updateEntry.jsp:

- Depending on which search method was selected in userManagementMenu.jsp, updateEntry.jsp will display different fields. If the admin had chosen to search by user and entered in a valid username, the page will display the username being updated as well as text fields filled with the current information for that user. This uses sql statement (4.1) to get the information related to that username. If the admin had chosen to search by persons and entered in a valid person ID, the page will display the person ID being updated as well as text fields filled with the current information for that person. This uses sql statement (4.2) to get the information related to that person ID. If the admin had chosen to search by family doctors and had entered in valid person and doctor IDs, the page will display the person ID being updated as well as text fields filled with the doctor ID of that person. This uses sql statement (4.3) to get the information related to that

person ID and doctor ID. If incorrect information was given by the searchUserManagement.jsp, the page will display an error and return to the userManagementMenu.jsp page. If the user's session indicates that they are not an administrator, an access denied error message is displayed and the user is sent back to the main menu.

updateSubmit.jsp:

- This page takes the information entered from updateEntry.jsp and uses it to update the data associated to the key given in searchUserManagement.jsp. A message is displayed saying whether or not the update was successful. If the information being updated is for the user table, sql statement (5.1) is used. If the information being updated is for the persons table, sql statement (5.2) is used and if the information being updated is for the family doctor table, sql statement (5.3) is used. If the user's session indicates that they are not an administrator, an access denied error message is displayed and the user is sent back to the main menu.

createEntry.jsp:

- Depending on which search method was selected in userManagementMenu.jsp, create Entry.jsp will display different fields. If the admin had chosen to create a user, the page will display text fields asking for relevant information to create the user. If the admin had chosen to create a person, the page will display text fields asking for relevant information to create the person. If the admin had chosen create a family doctor entry, the page will display text fields asking for relevant information to create the family doctor entry. If the user's session indicates that they are not an administrator, an access denied error message is displayed and the user is sent back to the main menu.

createSubmit.jsp:

- This page takes the information entered from createEntry.jsp and uses it to create an entry in the table selected by userManagementMenu.jsp. A message is displayed saying whether or not the create was successful. If the information being created is for the user table, sql statement (6.1) is used. If the information being created is for the persons table, sql statement (7) is used to determine the current highest person ID. The obtained person ID is incremented to create the new ID and the new data entry is created with sql statement (6.2). If the information being created is for the family doctor table, sql statement (6.3) is used. If the user's session indicates that they are not an administrator, an access denied error message is displayed and the user is sent back to the main menu.

### ***SQL Statements:***

- (4.1) select password, class, person\_id from users where user\_name = "" + username + ""
- (4.2) select first\_name, last\_name, address, email, phone from persons where person\_id = "" + personID + ""
- (4.3) select \* from family\_doctor where doctor\_id = "" + docID + "" AND patient\_id = "" + personID + ""
- (5.1) UPDATE users SET password="" + password + "", class="" + classStr + "", person\_id="" + personID + "" WHERE user\_name="" + username + ""

- (5.2) UPDATE persons SET first\_name="" + firstName + "", last\_name="" + lastName + "", address="" + address + "", email="" + email + "", phone="" + phone + "" WHERE person\_id="" + personID + ""
- (5.3) UPDATE family\_doctor SET doctor\_id="" + docID + "" WHERE patient\_id="" + personID
- (6.1) INSERT INTO users VALUES("" + username + "", "" + password + "", "" + classStr + "", "" + personID + "", to\_date("" + dateStr + "", 'MM/DD/YYYY'))
- (6.2) INSERT INTO persons VALUES("" + personID + "", "" + firstName + "", "" + lastName + "", "" + address + "", "" + email + "", "" + phone + "")
- (6.3) INSERT INTO family\_doctor VALUES("" + docID + "", "" + personID + "")
- (7) select MAX(person\_id) from persons

## Report Generating Module

### *Description:*

report.jsp:

- This module is only accessible to those with an administrator account. The option will not appear if the stated requirement is not met. This gives the user the ability to generate a list of patients that match the parameters given by the user. Text fields where the diagnosis, and a date range are entered and the variables are passed to reportResult.jsp as iDiag, iDate1, and iDate2 respectively. When the submit button is pressed the page is redirected to reportResult.jsp. If any of the text fields are left empty, an error message will appear prompting the user to fill in all fields.

reportResult.jsp:

- The file takes in variables iDiag, iDate1, and iDate2 to be used in printing and SQL statements. The database is connected to and SQL statement (8) is ran to get the rows that match the specifications of the retrieved variables. The row is printed out into a HTML table for the user to view.

### *SQL Statements:*

- (8) select p.last\_name, p.first\_name, p.address, p.phone, r.test\_date from persons p, radiology\_record r where p.person\_id = r.patient\_id AND LOWER("" + iDiag + "") = r.diagnosis AND r.test\_date between to\_date("" + iDate1 + "", 'DD/MM/YYYY') AND to\_date("" + iDate2 + "", 'DD/MM/YYYY') ORDER BY p.last\_name

### *Reuse Statements:*

reportResult.jsp:

- Modified from:  
[https://eclass.srv.ualberta.ca/pluginfile.php/1336718/mod\\_page/content/91/Lab-IntroductionToHTML.pdf](https://eclass.srv.ualberta.ca/pluginfile.php/1336718/mod_page/content/91/Lab-IntroductionToHTML.pdf)

## Uploading Module

### *Description:*

uploadImage.jsp:

- uploadImage.jsp displays a form for a radiologist to enter a radiology record id and to browse their local file system for an image to upload. The radiologist must enter a valid record id along with an image to upload. Otherwise, an error message is displayed.

UploadImage.java

- This servlet handles the image uploading functionalities. It acquires the image from uploadImage.jsp and uploads the image into the database for the corresponding record id.
- void doPost(HttpServletRequest request, HttpServletResponse response): This method is called when uploadImage.jsp redirects to UploadImage.java. It parses the form data and uses the shrink() function to create different sized images for the thumbnail, regular, and full sizes. It executes sql statement (9) to insert empty blobs into the database and then uses sql statement (10) to write the images to the blob objects.
- BufferedImage shrink(BufferedImage image, int n): Shrinks the provided image by a shrinking factor of n. It returns the shrunk image.
- boolean isValidID(int record\_id): Checks to see if a valid record id has been entered by the radiologist. It uses sql statement (11) to see if the entered id exists in the database. If it exists, the function returns true, else false.

### *SQL Statements:*

- (9) INSERT INTO pacs\_images VALUES(" + record\_id + "," + image\_id + ", empty\_blob(), empty\_blob(), empty\_blob())
- (10) SELECT \* FROM pacs\_images WHERE image\_id = " + image\_id + " FOR UPDATE
- (11) select count(\*) from radiology\_record where record\_id = " + record\_id

### *Reuse Statements:*

UploadImage.java:

- Modified from: <http://luscar.cs.ualberta.ca:8080/yuan/UploadImage.java>

## Search Module

### *Description:*

searchStart.jsp:

- searchStart.jsp gives the user the ability to enter sorting preferences as well as enter in keyword(s), and a time period to search the database of records. Text fields where the keyword(s), and/or a date range are entered are set in variables passed to searchResult.jsp as

searchKeyword, searchTime1, and searchTime2 respectively. The sorting preference is set into the variable searchtype as either recentFirst, recentLast or relevance. When the submit button is pressed the page is redirected to searchResult.jsp. An error message is displayed if the user does not enter in the information correctly.

searchResult.jsp:

- searchResult.jsp takes in variables searchKeyword, searchData1, searchData2 and searchtype to be used in SQL statement. Depending on the value in searchtype, a corresponding SQL statement will be set for the order. The SQL statement is then dynamically prepared depending on the information that is given and the variables entered giving SQL statement (12). The keywords are also split in case of multiple keywords and a proportional number of SQL statement sets will be prepared to be used. A HTML table is made, the database is connected to where the SQL statement will be used giving a result to be put into the table. All the radiology record information, the patients first and last name, as well as any related images are displayed in the result of the table.

viewImage.jsp:

- This page displays the regular sized image of the thumbnail that was clicked in the search results. When the regular sized image is clicked, the image zooms into its full sized image and vice versa. It uses GetOnePic.java to display the image on the page.

GetOnePic.java:

- This servlet obtains the image\_id from the query string and executes one of (13.1, 13.2, 13.3) depending on the size of the image specified in the query string. It retrieves the imagebyte from the result set and writes it to the page.

### ***SQL Statements:***

- (12a) select r.\*, p.first\_name, p.last\_name
- (12a.1) 6\*score("+ Integer.toString(matchNum+1) +")+6\*score("+ Integer.toString(matchNum+2) +")+3\*score("+ Integer.toString(matchNum+3) +")+score("+ Integer.toString(matchNum+4) +") as rank
- (12b) from radiology\_record r FULL JOIN persons p ON r.patient\_id = p.person\_id where
- (12b.1) r.patient\_id = "" + userID + "" AND
- (12b.2) r.doctor\_id = "" + userID + "" AND
- (12b.3) r.radiologist\_id = "" + userID + "" AND
- (12c) contains(r.diagnosis, "" + wordList[i] + "", "+ Integer.toString(countNum+3) +") > 0 OR contains(r.description, "" + wordList[i] + "", "+ Integer.toString(countNum+4) +") > 0 OR contains(p.first\_name, "" + wordList[i] + "", "+ Integer.toString(countNum+1) +") > 0 OR contains(p.last\_name, "" + wordList[i] + "", "+ Integer.toString(countNum+2) +") > 0
- (12c.1) ORDER BY r.test\_date DESC
- (12c.2) ORDER BY r.test\_date ASC
- (12c.3) ORDER BY rank desc
- (13.1) select thumbnail from pacs\_images where image\_id="" + image\_id.substring(9)

- (13.2) select regular\_size from pacs\_images where image\_id=" + image\_id.substring(7)
- (13.3) select full\_size from pacs\_images where image\_id=" + image\_id

### ***Reuse Statements:***

searchResult.jsp:

- Modified from: <http://webdocs.cs.ualberta.ca/~yuan/servlets/inverted/myIndex.sql>
- Modified from: <http://webdocs.cs.ualberta.ca/~yuan/servlets/inverted/item.sql>
- Modified from: <http://webdocs.cs.ualberta.ca/~yuan/servlets/inverted/drjobdml.sql>

GetOnePic.java:

- Modified from: <http://luscar.cs.ualberta.ca:8080/yuan/GetOnePic.java>

## **Data Analysis Module**

### ***Description:***

analysisStart.jsp:

- analysisStart.jsp can only be accessed by an administrator. Once accessed analysisStart.jsp allows the user to fetch results via a "cube". The user checks off at least one of the three check boxes to select what tables are to be displayed. There are options of patient ID, test type, and time period, which are stored in the variables displayType1, displayType2 and displayType3 respectively. If the user wishes to display the time period, another option must be chosen below the check boxes, which indicates if the week, month or year is to be displayed. When the submit button is pressed, the page is redirected to analysisResult.jsp. If no options are selected or if time period is selected but no type of date, then an error message appears.

analysisResult.jsp:

- analysisResult.jsp displays the data selected from analysisStart.jsp options into a HTML table. displayType1 to displayType3 are retrieved from analysisStart.jsp and are processed through different conditions. Depending on which of the options are selected, a different sql statement is added dynamically to the final sql string to create the statement seen in (14). The database is then connected to and the tables are displayed for the user. The database is then closed.

### ***SQL Statements:***

- (14a) select
- (14a.1) p.person\_id
- (14a.2) r.test\_type
- (14a.3.1) trunc(r.test\_date, 'IW') as test\_date
- (14a.3.2) trunc(r.test\_date, 'MM') as test\_date
- (14a.3.3) trunc(r.test\_date, 'Y') as test\_date

- (14b) count(i.record\_id) as image\_count from persons p, radiology\_record r, pacs\_images i where p.person\_id = r.patient\_id AND r.record\_id = i.record\_id group by CUBE(
- (14b.1) p.person\_id
- (14b.2) r.test\_type
- (14b.3) test\_date
- (14c) )

(Any combination of steps 14a.1-3 and 14b.1-3 can be used)

***Reuse Statements:***

Adapted data cube sql code from -

[http://docs.oracle.com/cd/B19306\\_01/server.102/b14223/agggreg.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14223/agggreg.htm)