Login Module

The Login Module is the main access point that a user will use to access our web application radiology database system. This module contains a static HTML file that contains the form that ask the user for their authentication information. Once a user has inputted their username and credentials(password), a connection is created using the ConnectionMaker Module and the connection and a request will be sent to the SQLController Module containing the information that the user has inputted (username, password). The SQLController Module will handle the database request sending the following query, “SELECT password FROM users WHERE user_name = input_user_name”. The SQLController Module will handle the database response and return a boolean for the Login Module to determine if to allow access to the user. If the returning object is True, the Login Module will create a session for the user, saving information that will be useful for other modules, and redirects the user to the Home Menu Module.

Home Menu Module

Once the Login Module has allowed access to the user, the Home Menu Module is the user’s access interface for all other modules. Based on the class of the user that has just logged in, it will determine the static HTML file to use that will contain the links to all other major functional modules. An administrator would be able to access all other modules (except for the create radiology record module), being able to update, search, analyze and ask for reports on all information in the database. A radiologist would be limited to searching users that the radiologist has personally entered and the module to insert and update radiology records. A doctor would be limited to searching users that are their patients or have been at some point. A patient will only have access to their own records. All users will have access to their own Account Management Module that allows users to update the following personal information: name, username, password, phone number, and email.

User Management Module

This module is only accessible to Administrators. This module allows the Admin access to all user accounts and allows the Admin to create new accounts for the radiology system. If an Admin would like to update an account, they are prompted for a person ID number. Once an appropriate ID number is given, the module calls the SQLMaker and connectionmaker to access the database to retrieve any information that is associated to that particular ID number. This information is retrieved by the query, “SELECT * FROM users u,persons p,family_doctor d WHERE p.person_id = u.person_id AND p.person_id = d.patient_id AND u.user_name=<input_user_name>'". Once the information related to that user is retrieved, a form will appear that will have all the corresponding fields filled in. The admin can then change any of the fields shown. Note that an admin can change his/her own personal information using this module.

Report Generating Module

This module is only accessible to Administrators. This module allows the Admin to generate a report based on a specified diagnosis within a certain time frame. If an Admin would like to generate a report, they are prompted for the specified diagnosis and the time frame in which patients have been first tested with that diagnosis. The query “SELECT * FROM (SELECT p.first_name,p.last_name,p.address,p.phone,r.test_date,ROW_NUMBER() OVER (PARTITION BY p.person_id ORDER BY r.test_date) AS rownumber FROM persons p,radiology_record r WHERE p.person_id=r.patient_id AND r.diagnosis=<diagnosis> AND r.test_date BETWEEN TO_DATE(<start_date>','yyyy-mm-dd') AND TO_DATE(<end_date>,'yyyy-mm-dd')) WHERE rownumber=1". Note that this query will only give the date of the first record with that diagnosis for each patient. Once both are given, a table with the patient’s name, address, phone number, and day the test was perform will be displayed.

Radiology Record Management Module

This module is only accessible to Radiologists. This module will first do a check to make sure the user requesting access has a session that is translated as a user that is logged in as a radiologist. This module allows radiologists to create new radiology records and upload images associated with that radiology record. This module contains a static HTML file that contains a form with the fields required to create a radiology record. Once all information has been entered, the HTML file will send it to an HTTPservlet that translates the request to an SQL statement, creating a connection to the DBMS. A javascript file will ask for the jpg files that are to be associated with the radiology record. To create the radiologist record, this module will count the amount of radiology records existing in the database and increment the count for a new Record ID number. This module also similarly assigns image ID numbers through this method. Once all information has been collected and parsed, this module will execute the following SQL Statement "INSERT INTO radiology_record VALUES (<Input from user>)" and commit the new radiology record. After the record has been pushed to the database, this module can associate images to that record by inserting to the relational table "pacs_images" with the following SQL statement: "INSERT INTO pacs_images VALUES image_id, record_id ….<values>". This module is extended to another interface that allows users to input the radiology record ID and update it with more images if required, through the statements previously mentioned.

Search Module

This module is accessible to all users; however, depending on the type of user the search output is filtered accordingly (User information restrictions were mentioned in Home Menu Module). This module consists of a static HTML file that contains a form that allows the user to input a number of keywords separated by a comma, a date to return data from, a date to return data to, and the chosen method of ranking output data. Once a request is given by the user, the Search Module connects to the database

throught the ConnectionMaker Module and sends the connection and input data to SQLController Module. The SQLController dynamically creates an SQL query for the input given. The rankings that are allowed to chosen are the most recent radiology records, the earliest radiology records, and through the following function: Rank(record_id) = 6*frequency(patient_name) + 3*frequency(diagnosis) + frequency(description).

The SQLController sequentially goes through the request sent determining which fields have been given by the user as a request. If keywords are given, SQLController will create a statement that will request for all radiology records with all the keywords by first splitting the keywords into a list, and then iteratively going through each item and adding it to the function. To get the overall ranking the following SQL statement is used in this module: SELECT <all record information> , 6*(SCORE(1)+Score(2)+(Score(3).....Score(n))+3*(SCORE(1)+Score(2)+(Score(3)..... Score(n))+(SCORE(1)+Score(2)+(Score(3).....Score(n)) as rank FROM radiology_record r, persons p, pacs_images m WHERE p.person_id = r.patient_id userID AND <date restriction included> AND (contains(first_name,'<term>',1)>0, contains(last_name,'<term>',2)>0, contains(diagnosis,'<term>',3)>0, contains(description,'<term>',4)>0, ….contains(description,"<term n>', n)>0... ORDER BY <Rank Type>". The result set that is ranked is returned to the Search Module which also grabs the images associated with the records and outputs the records in a table for the user to view.

Data Analysis Module

This module is only accessible to Administrators. This module allows the Admin to generate an OLAP report for data analysis. If the admin wants to generate an OLAP report, they can do so by selecting certain options they want to include in the report. The time period option allows three options: Weekly, Monthly, or Yearly. When the admin generates the report, a sql string and data cube is generated based on the

options the admin has chosen. An example of a generated sql statement would be: “SELECT first_name, last_name, TRUNC(test_date, 'YEAR') AS time, COUNT(image_id) FROM analysis_view GROUP BY CUBE (first_name, last_name, time)” to get number of images of each patient for each year. The analysis_view is a view created with the joining of three tables: persons,radiology_record, and pacs_images. After getting the result set, all the values with null will be parsed and excluded from the table displayed. The view can be refreshed by recreating the analysis_view.

connmaker.java

This java file establishes a connection for all the modules to connect to oracle. You must input your own oracle username and password in this file and then compile it.

SQLController.java

This java file is one of the main parsers for any difficult or repeated SQL statements. Each module calls this module and queries information that will be returned.