GIORG Online Query Platform Report

Chen Ju

7th May 2007

Table of Content:

Chapter 1:

What is the project about?

The GRIOG is a cross-hospitals study on the gastro and intestine cancer lead by NUS team. There are 6 hospitals involved in the project. They are: National University Hospital, SGH, John Hopkins Hospital, TTSH, NCC and Alexandra Hospital.
This project aim to manage all their patients' information in a way that cross-hospital/department, web-based and secure queries can be implemented easily.
Previously, the data was stored in document files, and the NUS staffs need to go to different hospitals to get their data and key them into MS excel file for further analysis. During the process, error could be easily introduced. More importantly, because different hospitals were using different scheme to record patient information, it was almost impossible to retrieve the "join" data cross-hospitals. These are the main reason why a new system must come into play.

What are the Challenges?

There are a few challenges need to be solved in building this system.

Problems	Difficulty
Handling special symbols.	
Varity types of queries need to be considered	There are many types of queries would be asked in the system. And we cannot directly write the SQL language to the server (must use GUI). Most commonly ask queries must be analyzed and optimized before hand.
Standardization of the information schema across hospitals.	As mention, different hospital use different schema to record their patients' information. This is not only reflected in the attributes name, and document type, but also in the format of the data values. For instance, some dataset use string data type to record the patient birthday; and there are many cases that the height attribute has character comment as its value. This string type value would make the comparison query impossible.
Flexibility for future update	Attributes for this study might change. Some new attributes may be introduced; while some old attributes may become unnecessary. All the function to do this modification should be implemented in the system.
Easy data bulk uploading	The clients currently store all their data in either MS Access file or Excel file. However, because of the standardization problem, almost every dataset has its

	own ways of recording. (Fortunately, most of the attribute names in used are the same, only the arrangement of these attributes are different.) Either we manually arrange the attributes before uploading or we write a program to do this automatically.
Direct access to the tree structure data record	The clients want to directly access some part of the data under one patient, like their histopathology record. Because the database is hierarchical, the change to histopathology record might cause the foreign key constrain problem (when they want to delete).

Chapter 2:

User requirement:

The clients' requirement could be divided into three parts:

- Administration function
- Data entry function
- Data retrieval function

1. Administration function

Administration includes functions like creating new doctors account, deactivate an existing doctor account, and modify certain attributes in patients' information, etc. Detail specification is listed below:

A. Creating new doctor account by the database administrator. A doctor account could be able to send a query to the database administrator to retrieve data he/she specify; or send a request to the database administrator to report some error he/she found in the data. Lastly, he/she can also update their particulars, like password.

B. Creating new site manager account by the database administrator. A site manager is playing the role of data entry in each hospital. Beside sending request to database administrator and modifying their particulars, he/she can also add, delete and update the patients' information.

C. Deactivate an existing doctor account by the database administrator. A record of the deactivation request is stored in the database and also a hard copy of the deactivation license form should be printed out.

D. Deactivate an existing site manager account by the database administrator. A record of the deactivation request is stored in the database and also a hard copy of the deactivation license form should be printed out.

E. Processing the request sent by doctor/site manager. The list of unprocessed requests is sorted by the date. Once a request is processed, it will be removed from the list but store in the database.

F. Processing the query sent by the doctor. Detail specification is included in the data retrieval function part.

G. Modifying the attributes being studied.
   Add new attributes to the patients' information. Set the default value of this new attribute to "N/A" when it is a string data type; set it to -1 when it is a numeric data type; set it to 1900-1-2(yyyy-mm-dd) when it is a date data type; set it to false when it is a Boolean data type.
   Update the attribute name.

2. Data entry function

Site manager performs this function. Site manager go through the following steps to enter a patient's information.

Step 1: Search a patient by his/her NRIC. If the patient is existed and the site manager has the permission to view it, then system will go to step 3 and display the patient's name and study ID. Otherwise, go to step 2.

Step 2: Create a new patient record with his/her name, NRIC, and then go to step 3.

Step 3: Create a new hospital entry for this patient. The hospital entry record should contain the current date and the hospital the site manager is in. Then move on to step 4

Step 4: For a hospital entry record, there are 4 types of information (The number of types could be increase in future). They are presentation record, histopathology record, surgery record and oncology record. The site manager could choose any one of them. If he chooses presentation, system will go to step 5; if he chooses histopathology, system will go to step 6; if he chooses surgery he will go to step 7; if he choose oncology, system will go to step 8.

Step 5: Enter the patient's presentation information. On submit, system can detect any inconsistency in the data type site manager has entered. For instance, if the data type for height is integer, if a string value in entered, system can inform the site manager. Following table is list of the data type for each attribute.

Attribute name	Data type	Default Value
Date of Visit	Date	1900-1-2
Gender	String	N/A
Birth Date	Date	1900-1-2
Age	Integer	-1
Ethnicity	String	N/A
Smoke	Integer	-1
Drink	Integer	-1
Occupation	String	N/A
Housing Type	String	N/A
Married	String	N/A
Children	String	N/A
Past history	String	N/A
Weight	Integer	-1
Height	Integer	-1
BMI status	String	N/A

Comment	String	N/A
Primary Symptom	String	N/A
Secondary Symptom	String	N/A
Surgery	String	N/A
Dukes	String	N/A

Table 2.1 Attributes in presentations records

Step 6: For a histopathology record, it contains several samples. This sample record has attributes – type, specimen Number and specimen date. The data type for each attribute is listed in the table. After the sample is created, system can go to step 9.

Attribute Name	Data Type	Default Value
Type	String	N/A
Specimen Number	String	N/A
Specimen Date	Date	1900-1-2

Table 2.2 Attributes in histopathology sample records

Step 9: For a (Histopathology) sample record, it contains either a tumor sample set record or a polys sample set record, but not both. Click the “Add tumor sample” button, system will go to step 10; click the “add polys sample” button, system will go to step 11. After the tumor/polys sample set record is added, user can also view the record.

Step 10: Enter the patients’ tumor sample set record. On submit, the system will detect any inconsistency in the data type site manager has entered. Following table is list of the data type for each attribute.

Attribute Name	Data Type	Default Value
Histology Dx (hist)	String	N/A
Malignant Polyp	String	N/A
Mets Site (hist)	String	N/A
Grade (hist)	String	N/A
Adhesion	String	N/A
Duke’s	String	N/A
T	String	N/A
N	String	N/A
M	String	N/A
Stage	String	N/A
N Positive	String	N/A
N Sampled	Integer	-1
Satellites Dep	String	N/A
Margins	String	N/A

Clin Dx (Ca)	String	N/A
Mucinous	String	N/A
Signet cells	String	N/A
Lymphoid	String	N/A
Perforated	String	N/A
Peritoneal fluid	String	N/A
Malignant cells	String	N/A

Table 2.2 Attributes in tumor sample set records

Step 11: Enter the patients' polys sample set record. On submit, the system will detect any inconsistency in the data type site manager has entered. Following table is list of the data type for each attribute

Attribute Name	Data Type	Default Value
Preexist	String	N/A
N polyps	String	N/A
Size cm (largest)	Integer	-1
Polyp adenoma	String	N/A
Polyp hyperplastic	String	N/A
Polyp serrated	String	N/A
Polyp others	String	N/A

Table 2.2 Attributes in polys sample set records

Step 8: For one oncology record, there are several line records. Each of the line record has attributes "Current Regimen", "Date of line started" and "Reason for changing regimen" (Data type of each attribute is listed in the following table). Once a line record is created, system can move onto step 12.

Attribute Name	Data Type	Default Value
Current Regimen	String	N/A
Date of line started	Date	1900-1-2
Reason for changing regimen	String	N/A

Table 2.2 Attributes in oncology records

Step 12: For one line record, it can have multiple checkup records or cycle records. When the user wants to create a new checkup record, system will go to step 13; when the user wants to create a new cycle record, system will go to step 14.

Step 13: Enter the patients' checkup record. On submit, the system will detect any inconsistency in the data type site manager has entered. Following table is list of the data type for each attribute

Attribute Name	Data Type	Default Value
Date of Visit	Date	N/A
Oncologist	Date	1900-1-2
T	String	N/A
N	String	N/A
M	String	N/A
Metastasis Site Adrenal	String	N/A
Metastasis Site Bone	String	N/A
Metastasis Site Brain	String	N/A
Metastasis Site Liver	String	N/A
Metastasis Site Lung	String	N/A
Metastasis Site Others	String	N/A
Response	String	N/A
Date of recurrenc	Date	1900-1-2
Recurrence Site Adrenal	String	N/A
Recurrence Site Bone	String	N/A
Recurrence Site Brain	String	N/A
Recurrence Site Liver	String	N/A
Recurrence Site Lung	String	N/A
Recurrence Site Others	String	N/A
Disease Free Status	String	N/A
Status	String	N/A
Overall Survival	String	N/A
Date of the Death	Date	1900-1-2
Cause of the Death	String	N/A

Table 2.2 Attributes in checkup records

Step 14: After entering the cycle information, including start date of the cycle, end date of the cycle and cycle number, a new cycle record is created. For one cycle, there are several visits, when user wants to add a new visit, system will go to step 15.

Step 15: Enter the patients' visit record. On submit, the system will detect any inconsistency in the data type site manager has entered. Following table is list of the data type for each attribute.

Attribute Name	Data Type	Default Value
Date of Visit (for Hematology)	Date	1900-1-2
Hb (for Hematology)	Date	1900-1-2
Neutrophil	String	N/A
WBC	String	N/A
Platelet	String	N/A

Date of sample	String	N/A
Unconjugated Biliribun	String	N/A
Conjugated Bilirubin	String	N/A
Total Bilirubin	String	N/A
Regimen Name	String	N/A
Comments	String	N/A
Dosage	String	N/A
Date of Chemotherapy	Date	1900-1-2
Weight of Patient	Integer	-1
Height of Patient	Integer	-1
Toxicity	String	N/A
Other Discomforts	String	N/A
Aspirin	String	N/A
Statin	String	N/A

Table 2.2 Attributes in visit records

3. Data Retrieval function

A doctor sends an English expression query to the database administrator, and then the administrator will interpret the query and make the necessary selection through the system. When the system returns the result to the database administrator, he/she will send an email with the result back to the doctor.

The database administrator is not a computer specialist and don't know how to write the SQL language. So the system must be capable in the following:

A. Being able to choose patient records in specified hospital(s).
B. Being able to choose patient records in specified department(s).
C. Being able to select the attributes of patient information to be displayed.
D. For the selected attributes in C, a filter can be used to each of the attributes, and filter out the unwanted records. For example, assume patient have attributes A, B, C, D, E, F, the query is asking for the attribute A, B, C, D which have value A = a1 or a2, and B = b1. The filter can filter out all record that A is not a1 or a2, and B is not b1.
E. They query must be stored in the database together with doctor's ID.
F. Once the query has been processed, it should be remove from waiting list.
G. There must be a pathway that can pick out the patient information who has taken some drug inclusively and exclusively. Inclusive means for certain drugs, as long as the patient has taken it, even he/she also took other drugs, and the output must include this patient. Exclusive means for certain drugs, the output only select patients who have taken the specific drugs and these drugs only. For instance, Patient A has taken drug A, drug B and drug C; patient B has taken drug A and drug B. If the query is "patient took drug A and drug C inclusively", then the result is patient A. If the query is "patient took drug A and drug B exclusively", then the result is patient B. But when the query is

"patient taken drug A", the output should be patient A and patient B.

H. User can continue to do A – F after G.

I. The output should be available in both HTML format and MS Excel format.

Challenges' possible solution:

- *Handling special symbols*.
  Because individual groups collected most of the data, there was no general agreement on using what data type for each of the attributes. Therefore, some group used Date for date value; while some group used string to represent date. This cause problem when they are finally store in the database; and need to do comparison between them.
  In order to standardize these different symbols in used, I need a program to differentiate the valid format from the invalid format and I choose the MS Access to do the job. In MS Access, we can import external data, then I import the MS excel file data into Access. During the process, all the invalid format data are marked. Then I do the manual checking to all the value and change it to valid format.
  This solution works for relative small data set; but it requires tremendous amount of work if the data set is large.
- *Varity types of queries need to be considered*
  Because we cannot write SQL query to directly access the database, the system must be able to generate executable SQL queries. Thus it is better to constraint the query structure to certain format.
  Fortunately, after discussing with the users, they agree to stick to only a few types of queries, which are listed in the following table.

Type of queries	Examples
Simple conjunctive queries.	Select patient ID where height = 170 and gender = "male" and race = "Chinese"
Disconjunctive queries. (Query involves "OR")	Select patient ID where height = 170 and gender = "male" and race = "Chinese" or race = "Malay"
Inclusive on drugs queries (Query involves inclusive drug filtering)	Select patient ID where height = 170 and gender = "male" and race = "Chinese" and (taken drug 5FU and drug CAPE inclusively)
Exclusive on drugs queries(Query involves exclusive drug filtering)	Select patient ID where height = 170 and gender = "male" and race = "Chinese" and (taken drug 5FU & drug CAPE and 5FU & CAPE only)
At-least condition on drugs queries (Query specifies that at least the patient must take these drugs, he/she can take more than these drug but no less)	Similar to "Inclusive on drugs queries", but the condition is stricter that the patient must take both 5FU and CAPE and can take more other drugs, but no less.
At-most condition on drugs queries (Query specifies that at	Similar to "Inclusive on drugs queries", but the condition is stricter that if the

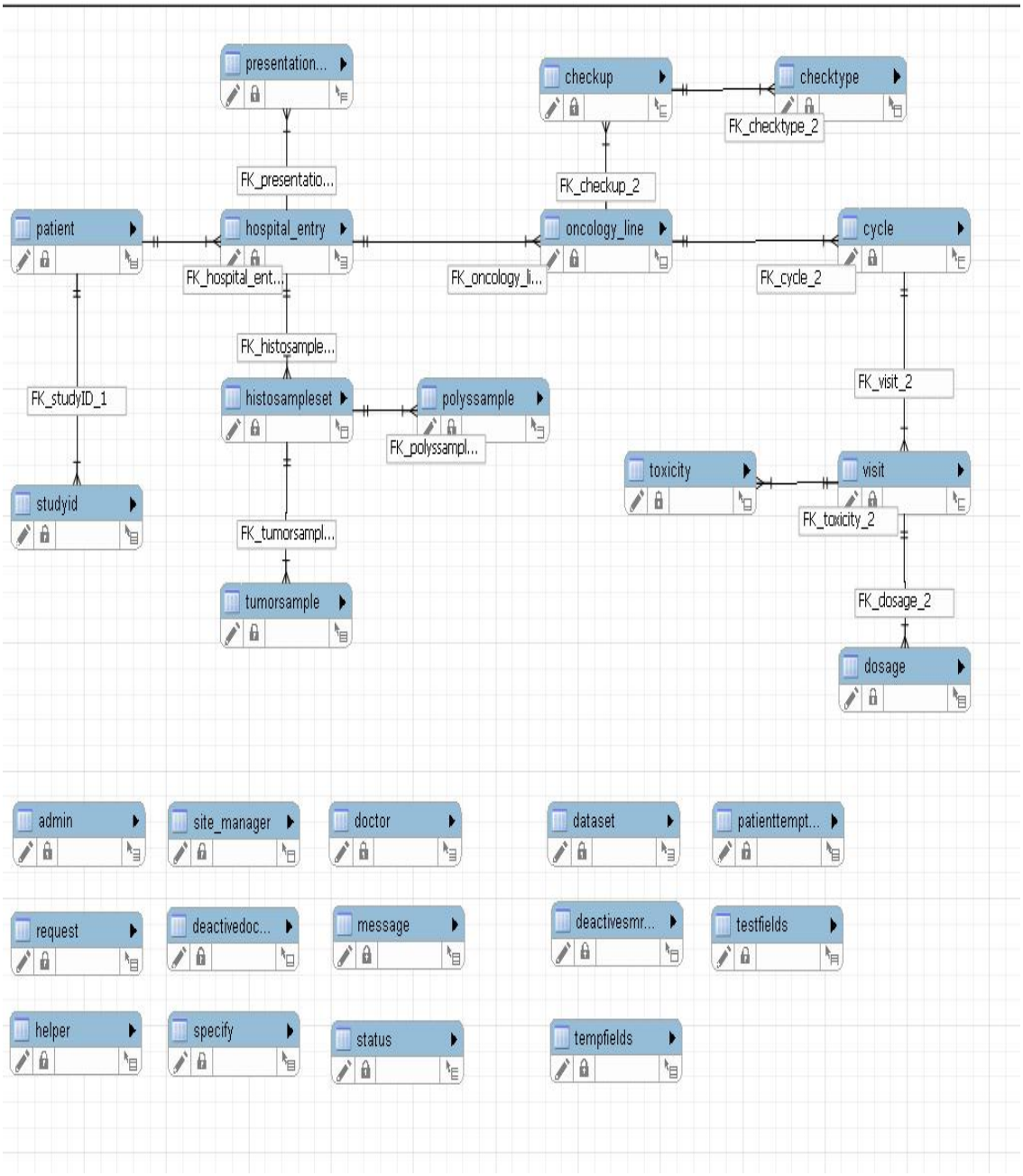
most the patient can take these drug, if he/she take more then these drugs, he/she is out of selection)	patient take drugs more then 5FU and CAPE, then he/she is out of selection.

Chapter 3:

Overview:

The interface of this system is built on JSP and servlet technology; a simple version of the STRUTS framework is also used.

Database design:



Solutions for the user requirements:

- **Types of Accounts used in the project**

  In my design for this system, there are three roles: doctor, site manager and database administrator. Doctor can only send requests and queries to database administrator. Site manager takes care of the data entry function. Database administrator has the reasonability to maintain all other administration stuff, like create new account/deactivate existed account; modify attributes being studied, etc. Also he/she is the only person can retrieve data from the database, thus he/she is responsible to correctly interpret the queries sent by doctors as well.

- **Procedure to deactivate a user account**

  Deactivating a doctor or a site manager requires permission from other officers. The flow of this procedure goes as following:

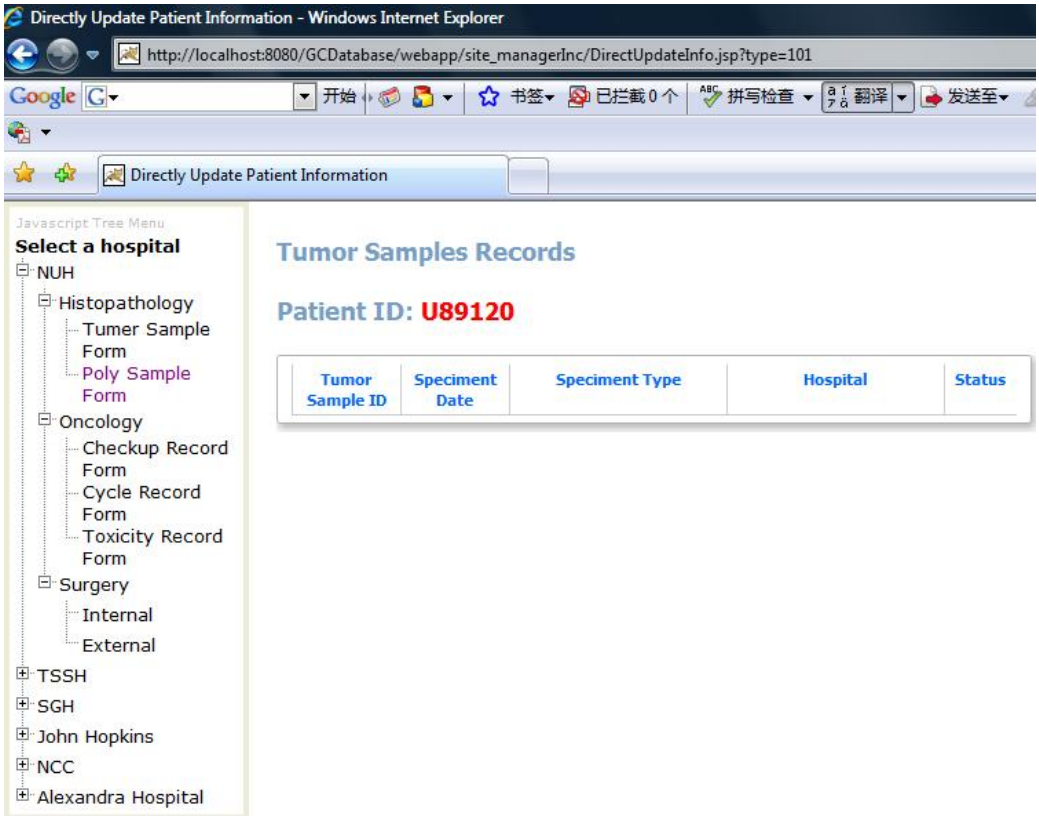
  1. Database administrator gets a request to deactivate a doctor.
  2. Database administrator searches for the doctor by his/her account ID. If the doctor's status is "activated", a database administrator changes his/her status to "processing" and prints out the deactivation license form. If the doctor's status is "processing", the database administrator can do nothing, or reprint the deactivate license form, or go to step 5. If the doctor's status is "deactivated", he needs to inform the request sender the error.
  3. After database administrator get the license form approved. He/she searches the doctor through the system and changes its status from "processing" to "deactivated".

- **Graphical interface for directly accessing patients information**

  In order to facilitate accessing to patient information, the system needs to provide a graphical view of the database's structure; and let the user to select the information on the view. For, instance, assume a user want to update the oncology record of a patient. With this interface, user needs to go through the step 1 through 8 mentioned in the data entry function requirement, before he/she can do the modification. However, this interface provides a shortcut that can directly go to the page to edit the information.

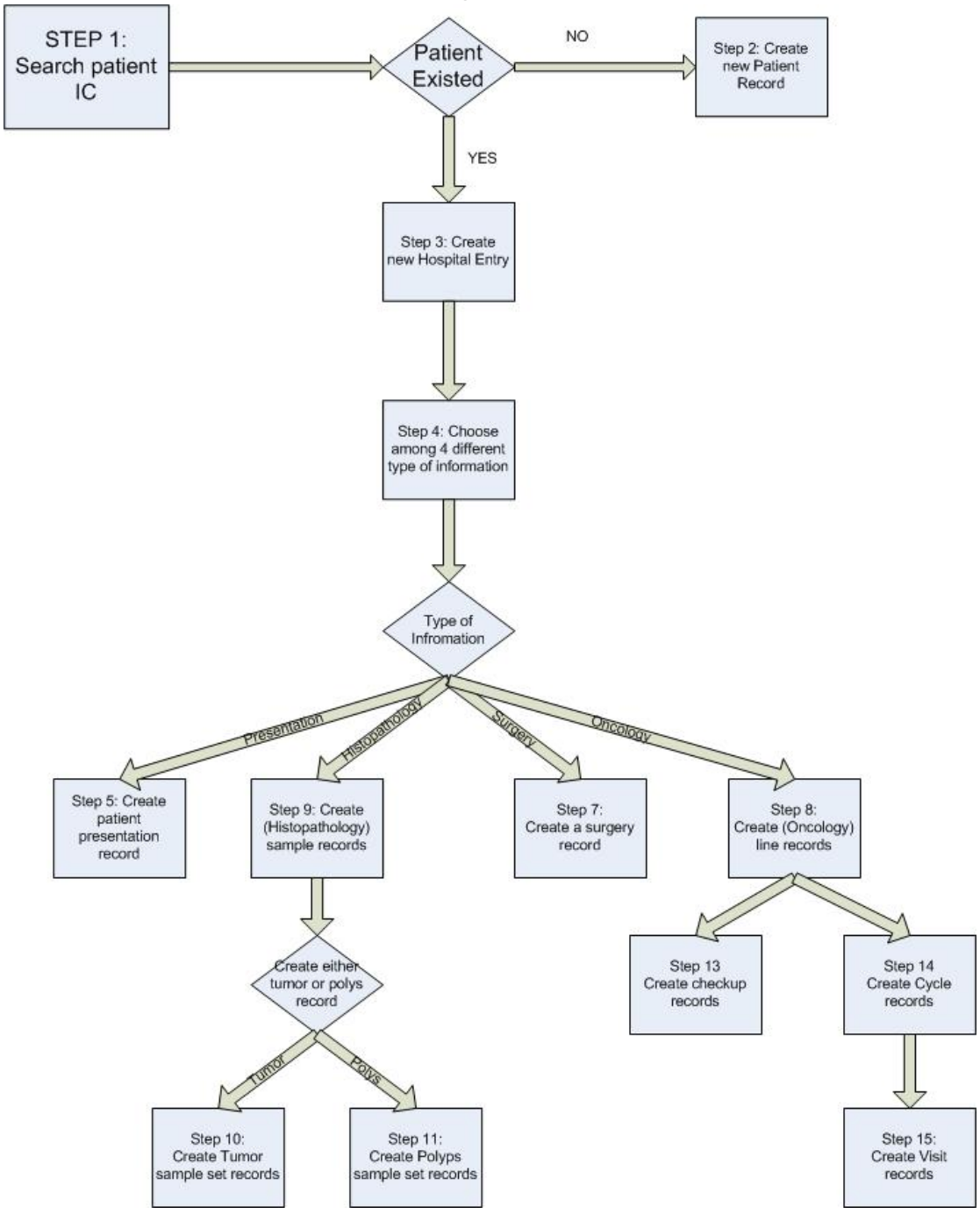
  The interface I implemented uses an open source package called JTreeView to draw the tree structure. Figure 3.1 is a snapshot of this interface.

Figure 5.1


Directly Update Patient Information - Windows Internet Explorer
http://localhost:8080/GCDatabase/webapp/site_managerInc/DirectUpdateInfo.jsp?type=101
Directly Update Patient Information
Javascript Tree Menu
Select a hospital
NUH
Histopathology
Tumer Sample Form
Poly Sample Form
Oncology
Checkup Record Form
Cycle Record Form
Toxicity Record Form
Surgery
Internal
External
TSSH
SGH
John Hopkins
NCC
Alexandra Hospital
Tumor Samples Records
Patient ID: U89120
Tumor Sample ID
Speciment Date
Speciment Type
Hospital
Status

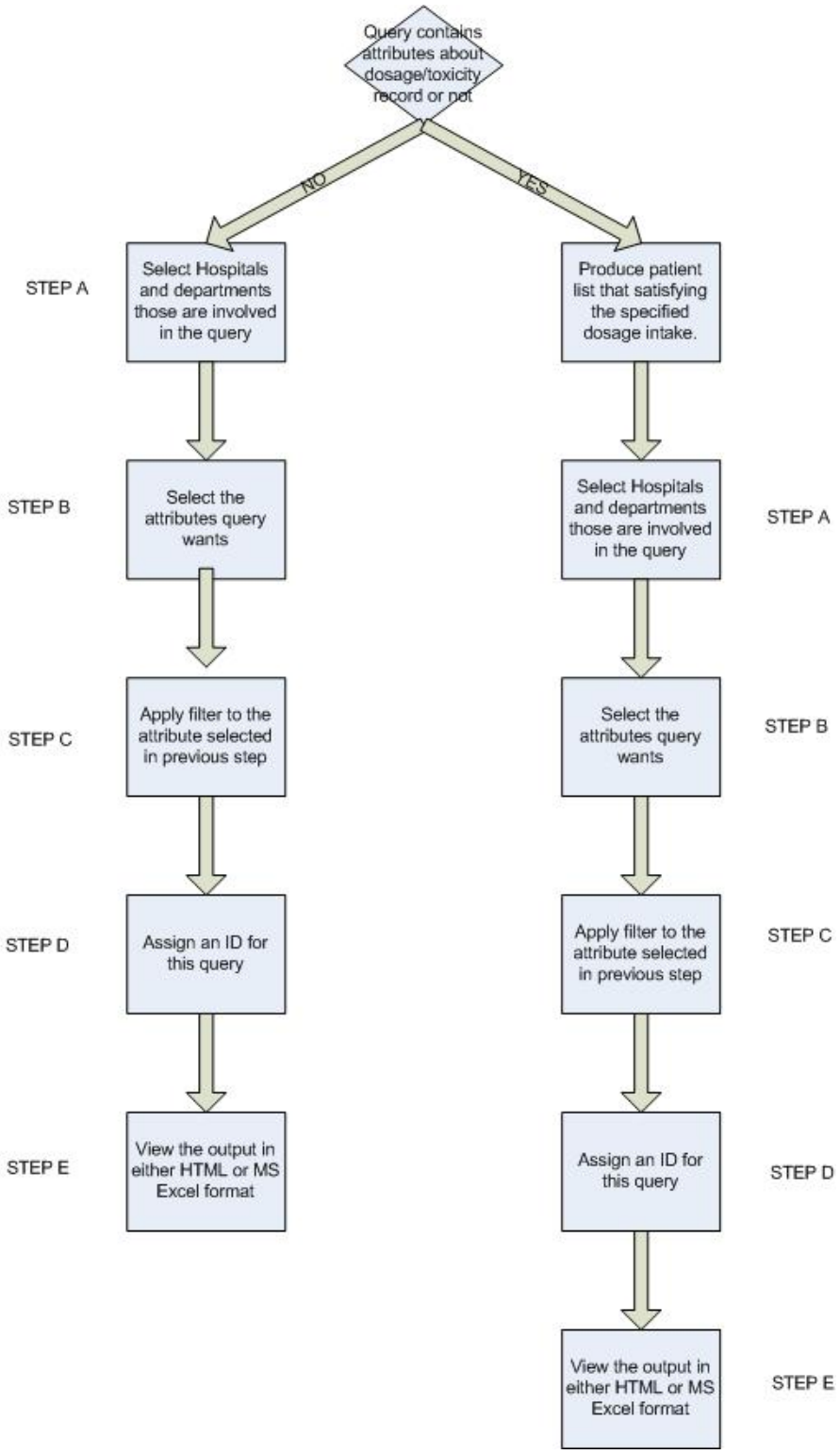
- **Data entry Flow**
  A diagram elaborating the flow of data entry function is in the following figure:

Figure 3.2



- **Generating SQL Queries Automatically**
  1. *Flow of generating SQL queries:*
     The flow is described in a step by step manner in the following diagram.

Figure 3.3



  2. *Structure of Generated Queries:*

For implementation purpose, all the SQL queries have a structure of

SELECT attri1, attri2, attri3, …… FROM table1 t1, table2 t2, table3 t3 …… WHERE cond1 AND cond2 AND cond3 AND ……
UNION
SELECT attri1', attri2', attri3', …… FROM table1' t1', table2' t2', table3' t3'…… WHERE cond1' AND cond2' AND cond3' AND ……
UNION ……

Within the WHERE clauses, only conjunctive clauses are allowed.

3. *Tricks to Implement "Inclusive" and "Exclusive":*
The most difficult part in generating SQL queries is to generate these inclusive and exclusive conditions. For instance, the doctor wants to query

"*SELECT patients' id where they have taken 5FU & CAPE, and 5FU & CAPE only with in the same cycle*"

"*SELECT patients' id where as long as they have taken 5FU & CAPE; I don't care what other drugs they have also taken.*"

The first case is exclusive query and the second case is the inclusive query. Professor Wong Limsoon helped me to develop a method to implement them using a temporary table. The SQL query generated is as follow:

```
1  create temporary table temp
2  select distinct dosageName from dosage
3  where dosageName = 'CAPE' or dosageName = '5FU';
4
5  create temporary table temp2
6  select distinct dosageName from dosage
7  where dosageName = 'CAPE' or dosageName = '5FU';
8
9  select pa.patient_id
10 from patient pa, hospital_entry he, oncology_line ol, cycle cy
11 where pa.patient_id = he.patient_id and he.Entry_ID = ol.entryID and ol.oncoLineID = cy.oncoLineID and
12 NOT EXISTS ( select t.dosageName from temp t where t.dosageName
13             NOT IN (select d.dosageName from dosage d, visit v where d.visitID = v.visitID and cy.cycleID = v.cycleID )
14           )
15 and
16 NOT EXISTS (select d.dosageName from dosage d, visit v where d.visitID = v.visitID and
17            cy.cycleID = v.cycleID and d.dosageName NOT IN (select x.dosageName from temp2 x)
18           );
```

The first "NOT EXISTED" block is to select patients who eat all the

specified drugs in the temporary tables.

The second “NOT EXISTED” block is to select patients who do not eat drugs that are not specified in the temporary tables.

- **Generate Result in a MS Excel Format**

  In order to write all the output into a MS Excel file, I used an open source package called JXL. This package can write data into excel file in a row-by-row manner.

  The problem with this package is it might have problem using the JDBC connector when it is used in UNIX system. However, with external JDBC connector installed in the UNIX system, this problem could be solved.

Chapter 4:

(Data currently not available)

Chapter 5:

(Data currently not available)