Presented by: Robert Peterson

Prince bizimana

Zachary Lessard



SVR DRS Design

Specification

Version [1.1]

March 2, 2017

## Table of contents

1. System Overview ................................................................................................................................................................. 3

1.1 Problem Definition ....................................................................................................................................................... 3

1.2 Proposed Solution ........................................................................................................................................................ 3

1.3 System Scope ............................................................................................................................................................... 4

1.3.1 OFD .................................................................................................................................................................... 4

1.3.2 ITC ...................................................................................................................................................................... 5

1.3.3 Overview ............................................................................................................................................................ 5

1.3.4 Disclaimer ........................................................................................................................................................... 5

2. Design Conventions.............................................................................................................................................................. 6

2.1 Overview of Conventions ............................................................................................................................................. 6

2.2 Object Naming Convention .......................................................................................................................................... 7

2.3 References ................................................................................................................................................................... 8

3. Database Design ................................................................................................................................................................. 9

3.1 ERD .............................................................................................................................................................................. 9

3.2 ESG .............................................................................................................................................................................. 10

4. Operations Design .............................................................................................................................................................. 17

4.1 Pseudo-Code ............................................................................................................................................................... 17

4.1.1 Generic Add: ..................................................................................................................................................... 17

4.1.2 Generic Inquiry: ................................................................................................................................................ 18

4.1.3 Generic Modify: ................................................................................................................................................ 18

4.1.4 Generic Delete: ................................................................................................................................................. 19

4.2 O-Spec ......................................................................................................................................................................... 20

4.2.1 Disease Category (E01) – ADD/DELETE/MODIFY .............................................................................................. 20

4.2.2 Disease Category (E01) – INQUIRY/REPORT ..................................................................................................... 20

4.2.3 Disease Definition (E02) – ADD/DELETE/MODIFY ............................................................................................. 21

4.2.4 Disease Definition (E02) – INQUIRY/REPORT .................................................................................................... 21

4.2.5 Disease Symptoms (E03) – ADD/DELETE/MODIFY ............................................................................................ 22

4.2.6 Disease Symptoms (E03) – INQUIRY/REPORT ................................................................................................... 22

4.2.7 Disease-Symptom Matrix (E04) – ADD/DELETE/MODIFY................................................................................... 23

4.2.8 Disease-Symptom Matrix (E04) – INQUIRY/REPORT ......................................................................................... 23

4.2.9 Country (E05) – ADD/DELETE/MODIFY .............................................................................................................. 24

4.2.10 Country (E05) – INQUIRY/REPORT ................................................................................................................... 24

4.2.11 State (E06) – ADD/DELETE/MODIFY.................................................................................................................. 25

4.2.12 State (E06) – INQUIRY/REPORT ........................................................................................................................ 25

4.2.13 Visiting Patron (E07) – ADD/DELETE/MODIFY .................................................................................................. 26

4.2.14 Visiting Patron (E07) – INQUIRY/REPORT ......................................................................................................... 27

4.2.15 General Prescriptions (E08) – ADD/DELETE/MODIFY ....................................................................................... 28

4.2.16 General Prescriptions (E08) – INQUIRY/REPORT .............................................................................................. 28

4.2.17 Diagnosis Summary (E09) – ADD/DELETE/MODIFY .......................................................................................... 29

4.2.18 Diagnosis Summary (E09) – INQUIRY/REPORT ................................................................................................. 30

4.2.19 Diagnosis Symptoms (E10) – ADD/DELETE/MODIFY ........................................................................................ 31

4.2.20 Diagnosis Symptoms (E10) – INQUIRY/REPORT................................................................................................ 31

4.2.21 Diagnosis Prescriptions (E11) – ADD/DELETE/MODIFY .................................................................................... 32

4.2.22 Diagnosis Prescriptions (E11) – INQUIRY/REPORT ........................................................................................... 32

4.2.23 Physician (E12) – ADD/DELETE/MODIFY .......................................................................................................... 33

4.2.24 Physician (E12) – INQUIRY/REPORT ................................................................................................................. 33

5. User Interface Design .......................................................................................................................................................... 34

5.1 Patient UI ..................................................................................................................................................................... 34

5.2 Doctor UI ..................................................................................................................................................................... 38

6. Message and Help Specification .......................................................................................................................................... 41

6.1 Message Specification ....................................................................................................................................................... 41

6.2 Help Specification .............................................................................................................................................................. 41

7. Summary .............................................................................................................................................................................. 41

## System Overview

### 1.1 Problem Definition

When attending a doctor’s appointment, a large portion of time is consumed figuring out what symptoms the patient is experiencing. Every time a person goes to their doctor’s appointment, they experience the same redundant process, which usually involves being asked a bunch of questions. This system is broken and can lead to many miscommunication problems which are outlined below:

1. Miscommunication between the doctor and the patient. Is the doctor asking the correct questions? If the doctor is not asking the correct questions or taking note of the wrong information, it could lead to a huge problem when making the correct diagnosis.
2. Is the patient comfortable with verbally speaking their symptoms? Many problems can arise here if the patient is not comfortable with telling his/her doctor what they are experiencing and might leave some crucial details out that would be critical to identifying the patient’s conditions.
3. Time consumptions. This becomes a problem when the ratio between patients to doctors are too high. A large number of appointments might overwhelm the doctors and cause him/her to pay less attention when diagnosing their patients. Another problem is that a large portion of a patients appointment is spent figuring out the symptoms they are experiencing. If that portion can be removed from the equation, more time can be dedicated to the actual diagnosis and possibly more appointments could be made in a day.
4. Patient information needs to secure, and safe; the system must provide an environment that will do this.
5. Change of location. Every time a patient changes their location or goes to a different hospital, they get stuck filling out loads of paper work. This is inconvenient and takes up a huge chunk of time.

### 1.2 Proposed Solution

This project addresses these problems by developing a Sexual, Viral, and Respiratory Disease Recognition System (SVR-DRS) designed to specifically tackle and solve the many issues found in the gathering information process within a doctor’s appointment.

This project will aim at creating an expert system towards getting crucial information from the patient prior to the appointment or during the appointment if prior is not possible. This will create a system where patients can be comfortable describing their symptoms via text, if they are not comfortable saying it in person.

It will also focus on shortening the amount of time spent gathering personal information (such as symptoms, medical history, and other information) from the patient. Based on patient symptoms that are entered it will also propose possible diagnosis to shorten the time spent in making the diagnosis. (Note: This is not a permanent diagnosis, but more of a pointer to a starting point for the doctor.)

This software system must be platform independent and portable (if possible), otherwise it will be primarily available for the Windows Operating System. The recommended software development tools recommended are but not limited to: Java on the front-end with a JavaFX Graphical User Interface and MySQL on the back-end DBMS.

### 1.3 System Scope

Disease Information Center (subsystem) accesses the following entities:

* Disease Categories
* Disease Definitions
* Disease Symptoms
* Disease Symptoms Matrix

Disease Diagnosis Center (subsystem) accesses the following entities:

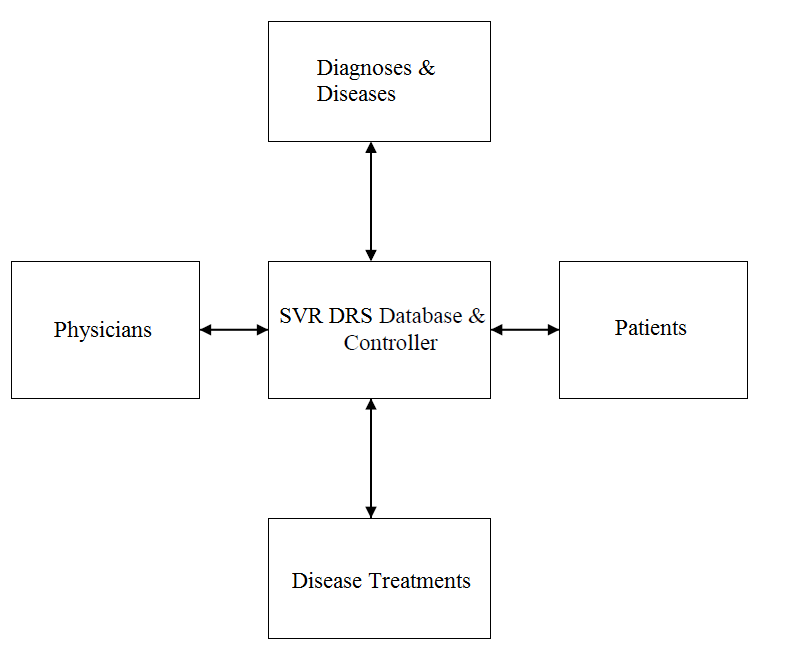
* Visiting Patrons
* General requirements
* Diagnosis Summaries
* Diagnosis Symptoms Details
* Diagnosis Prescriptions

Control Information Center (subsystem) accesses the following entities:

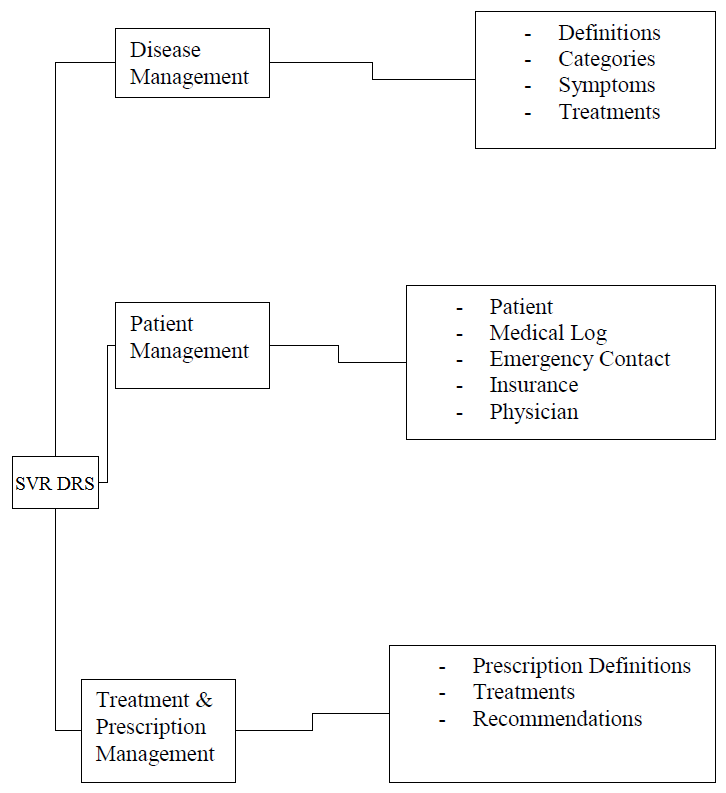
* Countries
* States
* Physicians

Other subsystems (read only) include the Inference Engine

### 1.3.1 OFD

****

### 1.3.2 ITC

****

### 1.3.3 Overview

The program will work by first obtaining disease, doctor, and patient information via a questionnaire embedded within the program. This information will be stored in a central database. The comparison between entered symptoms and disease symptoms will be calculated within the inference engine by pulling information from the

Central Database using the Disease Information Center. The information will then be transferred to the Disease Diagnosis Center for diagnosis summary and reports.

### 1.3.4 Disclaimer

Disclaimer: All information inside database before being put into production is not accurate and used for testing purposes of the program only. Test information must be deleted before implementation. The company is not responsible for any falsely entered information. The diagnosis system should not be used alone to diagnose any disease, but should be a pointer/starting point for the doctor, and make checking in for the patient easy

## Design Conventions

The Design Conventions will proceed under the following captions:

• Overview

• Object Naming Convention

• References

### 2.1 Overview

The object naming convention employed will be based on Foster’s proposal (see [Foster 2010b]) and paraphrased in the figure under section 2.2.

Database and object specification will observe the guidelines for the Entity Specification Grid (ESG) as described in [Foster 2010a]. For each ESG, the following conventions hold:

* The entities (object types) as presented, will easily transition into a set of normalized relational database.
* For entity, the reference name is indicated first, followed by the descriptive name, and the implementation name (indicated in square brackets). The reference name is used for cross referencing (for instance in the case of foreign key references). The implementation name is what will be used when the database and its related objects are created.
* Attributes that must be non-null are preceded by an asterisk (\*).
* All attributes are numbered sequentially to assist in referencing.
* For each attribute, the physical characteristics will be given (as described in the next section); the attributes implementation name will be indicated in square brackets; it will be indicated whether the attribute is a foreign key.
* The characteristic of each attribute is indicated in square braces ([ ]). An A denotes an alphanumeric data type, followed by the length in bytes. N denotes a numeric data type, followed by the length in bytes. L denotes “logical” data type. B denotes a BLOB. M denotes a variable length (memo) data type for free formatted text. T denotes time stamp.
* Implementation names are indicated (within square brackets) for all database objects that will be created.
* Data elements that will be implemented as foreign keys, in the normalized relational database, are identified by comment in curly braces, specifying what entity they reference.
* In specifying primary keys and indexes, the attribute number(s) is (are) specified in square brackets.
* Indexes (including primary key or candidate keys) to be defined on the entity are indicated.
* For each entity a comment describing the data to be stored is provided. Additionally, the entities implementation name is indicated in square brackets.
* Each operation defined on an entity will be given an implementation name, indicated in square brackets.

### 2.2 Object Naming Convention

|  |
| --- |
| Object Name: SSSXXXXXX\_MMn where  SSS represents the system or subsystem abbreviation (1-3 bytes);  MMn represents the object mode or purpose (1-3 bytes);  XXXXXXXXX represents the descriptive name of the object (6-8 bytes). |
| Valid subsystem abbreviations include  DIC: Disease Information Center  CIC: Control Information Center  DDC: Disease Diagnosis Center |
| Valid mode abbreviations include:  BR: A base relation (if relational DB model)  OT: An object type (if OO DB model)  LVn: A logical view (e.g. LV1, LV2, etc.)  NXn: An index to a base table or object type (e.g. NX1, NX2, etc.)  PK: Primary Key  FKn: Foreign Key (e.g. FK1, FK2, etc.)  ICn: Integrity Constraint (e.g. IC1, IC2, etc.)  AO: An ADD operation  MO: A MODIFY operation  ZO: A DELETE (Zap) operation  IO: An Inquire operation  FO: A Forecast operation  RO: A REPORT operation  XO: A UTILITY operation  DS: A database synonym or alias of a known database table  DC: A database constraint  DT: A database Trigger  DP: A database procedure or function  DK: A database package  MF: A Message File – a special purpose database table (file) to store the text (and other essential details) for diagnostic error and status messages |
| The descriptor used for a database base relation or object type is consistently used for other objects that directly relate to that object. For example, the objects used for the management of disease categories may be:   * DICDcMaster\_BR – a base relation to store data on disease categories * DICDcMaster\_NX1 – an index on the base relation * DICDcMaster\_AO – an operation to ADD disease categories * DICDcMaster\_MO – an operation to MODIFY disease categories * DICDcMaster\_ZO – an operation to DELETE disease categories * DICDcMaster\_IO – an operation to INQUIRE on disease categories * DICDcMaster\_RO – an operation to REPORT on disease categories * DICDcMaster\_XO – a utility operation related to disease categories * DICDcMaster\_LV1 – a logical view of the base relation |
| Attribute implementation name are mere abbreviations of their more descriptive names. |

### 2.3 References

[Foster, 2014a] Foster, Elvis C. with Shripad Godbole. Database Systems: A Pragmatic Approach. Bloomington, IN: Xlibris Publishing, 2010. See lectures 4 and 5.

[Foster, 2016b] Foster, Elvis C. Software Engineering: A Methodical Approach. Bloomington, IN: Xlibris Publishing, 2010. See lectures 9 and 10.

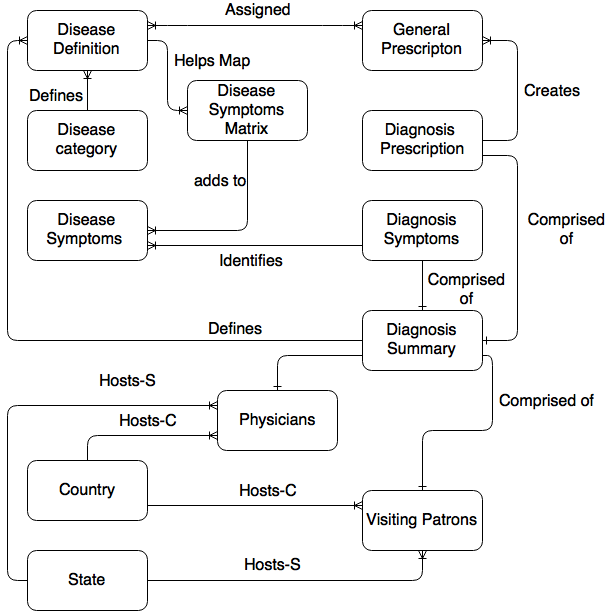
## Database Design

The Database Design will proceed under the following captions:

• ERD

• ESG’s

### 3.1 ERD



### 3.2 ESg’s

|  |
| --- |
| E01: Disease Category [DICDcMaster\_BR] |
| Attributes:  1. Disease Category Code (DCCode) [PK] [A3]  2. Disease Category Name (DCName) [A20]  3. Disease Category Description (DCDescription) [A30] |
| Comments:  1. Defines disease categories  2. Disease category code consists of 3 byte alpha code where the first 3 bytes represents the category. (CCC) (RES, SEX, VIR) |
| Indexes:  1. Primary Key: [1] (Constraint name is DICDcMaster\_PK)  2. DICDcMaster \_NX1 on [2] |
| Valid Operation:  1. Add Disease Category [DICDcMaster\_AO]  2. Delete Disease Category [DICDcMaster\_ZO]  3. Modify Disease Category [DICDcMaster\_MO]  4. Inquiry Disease Category [DICDcMaster\_IO]  5. Report Disease Category [DICDcMaster\_RO] |

|  |
| --- |
| E2 – Disease Definition [DICDdMaster\_BR] |
| Attributes:  1. Disease Code (DCode) [PK] [VarChar 7]  2. Disease Name (DName) [A30]  3. Disease Description (DDscr) [A30]  4. Disease Category Code (DCCode) (References E01) [A3] |
| Comments:  1. Stores definition of diseases  2. Disease Definition Code consists of 7 bytes alpha numeric code where the first 3 bytes are the disease category and the last 4 bytes are the disease number (CCCDDDD) |
| Indexes:   1. Primary Key: [1](Constraint Name is DICDdMaster\_PK) 2. DICDdMaster \_NX1 on [2, 3] |
| Valid Operations:  1. Add Disease Definition [DICDdMaster\_AO]  2. Delete Disease Definition [DICDdMaster\_ZO]  3. Modify Disease Definition [DICDdMaster\_MO]  4. Inquiry Disease Definition [DICDdMaster\_IO]  5. Report Disease Definition [DICDdMaster\_RO] |

|  |
| --- |
| E03: Disease Symptoms [DICDsMaster\_BR] |
| Attributes:  1. Disease Symptom Code (SCode) [PK] [N7]  2. Disease Symptom Name (SName) [A30]  3. Disease Symptom Description (SDscr) [A30] |
| Comments:  1. Definition for disease symptoms  2. Disease Symptom Code consists of a 7 byte numeric code where the first 4 bytes represents the disease number and the last 3 bytes represents the symptom number. (DDDDSSS) |
| Indexes:   1. Primary Key: [1](Constraint Name is DICDsMaster\_PK) 2. DICDsMaster \_NX1 on [2] |
| Valid Operations:  1. Add Disease Symptoms [DICDsMaster\_AO]  2. Delete Disease Symptoms [DICDsMaster\_ZO]  3. Modify Disease Symptoms [DICDsMaster\_MO]  4. Inquiry Disease Symptoms [DICDsMaster\_IO]  5. Report Disease Symptoms [DICDsMaster\_RO] |

|  |
| --- |
| E04: Disease-Symptom Matrix [DICDsmMaster\_BR] |
| Attributes:  1. Disease Code (DS\_DCode) (References E02) [VarChar 7]  2. Disease Symptom Sequence (DS\_Seqn) [N2]  3. Disease Symptom Code (DS\_SCode) (References E03) [N7]  4. Disease Symptom Rank (DS\_Rank) [N1]  5. Disease Frequency (DSFreq) [N2] |
| Comments:  1. Maps disease with symptoms |
| Indexes:  1. Primary Key: [1,2](Constraint Name is DICDsmMaster\_PK) |
| Valid Operation:  1. Add Disease Symptoms [DICDsmMaster\_AO]  2. Delete Disease Symptoms [DICDsmMaster\_ZO]  3. Modify Disease Symptoms [DICDsmMaster\_MO]  4. Inquiry Disease Symptoms [DICDsmMaster\_IO]  5. Report Disease Symptoms [DICDsmMaster\_RO] |

|  |
| --- |
| E05: Country [CICCntryMaster\_BR] |
| Attributes:  1. Country Code (CCode) [PK] [A5]  2. Country Name (CName) [A30]  3. Country Abbreviation (CAbbr) [A5] |
| Comments:  1. Stores basic information on countries  2. Country Code consists of 5 byte alpha code. (CCCCC) |
| Indexes:  1. Primary Key: [1](Constraint Name is CICCntryMaster\_PK)  2. CICCntryMaster\_NX1 on [2] |
| Valid Operations:  1. Add Country [CICCntryMaster\_AO]  2. Delete Country [CICCntryMaster\_ZO]  3. Modify Country [CICCntryMaster\_MO]  4. Inquiry Country [CICCntryMaster\_IO]  5. Report Country [CICCntryMaster\_RO] |

|  |
| --- |
| E06: State [CICStateMaster\_BR] |
| Attributes:  1. Province/State Code (PSCode) [PK] [A5]  2. Province/State Name (PSName) [A30] |
| Comments:  1. Stores basic information on Provinces or States  2. Province/State Code consists of 5 byte alpha code. (SSSSS) |
| Indexes:  1. Primary Key: [1](Constraint Name is CICStateMaster\_PK)  2. CICStateMaster\_NX1 on [2] |
| Valid Operations:  1. Add State [CICStateMaster\_AO]  2. Delete State [CICStateMaster\_ZO]  3. Modify State [CICStateMaster\_MO]  4. Inquiry State [CICStateMaster\_IO]  5. Report State [CICStateMaster\_RO] |

|  |
| --- |
| E07: Visiting Patron [CICVpMaster\_BR] |
| Attributes:  1. Patron Code (PCode) [PK] [VarChar 8]  2. Patron First Name (PFName) [A30]  3. Patron Last Name (PLName) [A30]  4. Patron Middle Initial (PMInitial) [A3]  5. Patron DoB (PDoB) [Date 8]  6. Patron Place of Birth (PPoB) [A30]  7. Patron Street Address (PStreet) [A30]  8. Patron City (PCity) [A30]  9. Patron Province/State (PPSCode) (References E06) [A5]  10. Patron Country Code ( PCCode) (References E05) [A5]  11. Patron Telephone (PTele) [N10]  12. Patron Email (PEmail) [A30]  13. Patron Primary care (PPC) (References E12) [A30] |
| Comments:  1. Stores personal information on visiting patrons  2. Patron Code consists of 8 bytes where the first 4 bytes are the patients first 4 letters of their last name and the last 4 digits the last 4 numbers of their Social Security Number. (LLLLSSSS) |
| Indexes:  1. Primary Key: [1](Constraint Name is CICVpMaster\_PK)  2. CICVpMaster\_NX1 on [2, 3, 4] |
| Valid Operations:  1. Add Patron [CICVpMaster\_AO]  2. Delete Patron [CICVpMaster\_ZO]  3. Modify Patron [CICVpMaster\_MO]  4. Inquiry Patron [CICVpMaster\_IO]  5. Report Patron [CICVpMaster\_RO] |

|  |
| --- |
| E08: General Prescriptions [DDCGpMaster\_BR] |
| Attributes:  1. Prescription Code (PrCode) [PK] [VarChar 8]  2. Disease Code (Pr\_DCode) (References E02) [VarChar 7]  3. Prescription Name (PrName) [A30]  4. Prescription Comment (PrComment) [A30] |
| Comments: The entity stores information about the Patient’s Medical Log |
| Comments:  1. Stores prescriptions identified for treating diseases  2. Prescription Code consists of 8 bytes where the first 4 bytes are the disease number and the last 4 digits the prescription ID number. (DDDDPPPP) |
| Indexes:  1. Primary Key: [1](Constraint Name is DDCGpMaster\_PK)  2. DDCGpMaster\_NX1 on [3] |
| Valid Operations:  1. Add General Prescriptions [DDCGpMaster\_AO]  2. Delete General Prescriptions [DDCGpMaster\_ZO]  3. Modify General Prescriptions [DDCGpMaster\_MO]  4. Inquiry General Prescriptions [DDCGpMaster\_IO]  5. Report General Prescriptions [DDCGpMaster\_RO] |

|  |
| --- |
| E09: Diagnosis Summary [DDCDsmmryMaster\_BR] |
| Attributes:  1. Diagnosis Reference Number (DgRef) [PK] [N12]  2. Diagnosis Date (DgDate) [Date 10]  3. Diagnosis Patron Code (DgPCode) (References E07) [VarChar 8]  4. Diagnosed Disease Code 1 (DgDCode1) (References E02) [VarChar 7]  5. Diagnosed Disease Code 2 (DgDCode2) (References E02) [VarChar 7]  6. Diagnosed Disease Code 3 (DgDCode3) (References E02) [VarChar 7]  7. Diagnosis Comment (DgComment) [A30]  8. Diagnosis Physician Code (DgPhyCode) (References E12) [VarChar 8] |
| Comments:  1. Stores information on the diagnosis of a patrons disease  2. Diagnosis Reference Code consists of a 12 byte numeric code. (NNNNNNNNNNNN) |
| Indexes:1. Primary Key: [1, 3, 4, 5, 6, 8](Constraint Name is DDCDsmmryMaster\_PK) |
| Valid Operations:  1. Add Diagnosis Summary [DDCDsmmryMaster\_AO]  2. Delete Diagnosis Summary [DDCDsmmryMaster\_ZO]  3. Modify Diagnosis Summary [DDCDsmmryMaster\_MO]  4. Inquiry Diagnosis Summary [DDCDsmmryMaster\_IO]  5. Report Diagnosis Summary [DDCDsmmryMaster\_RO] |

|  |
| --- |
| E10: Diagnosis Symptoms [DDCDsymptmsMaster\_BR] |
| Attributes:  1. Diagnosis Symptom Code (DSympCode) [PK] [N7]  2. Diagnosis Symptom Name (DSympName) [A30]  3. Diagnosis Symptom Description (DsympDscr) [A30]  4. Diseases Code (DSympSCode) (References E03) [FK1] [N12]  5. Diagnosis Symptom Summary (DgDCode2) (References E09) [VarChar 8] |
| Comments:  1. Stores information on the possible symptoms that are identified during the diagnosis  2. Diagnosis Symptom Code consists of a 7 byte numeric code. (NNNNNNN) |
| Indexes:  1. Primary Key: [1, 2](Constraint Name is DDCDsymptmstMaster\_PK) |
| Valid Operations:  1. Add Diagnosis Symptoms [DDCDsymptmsMaster \_AO]  2. Delete Diagnosis Symptoms [DDCDsymptmsMaster \_ZO]  3. Modify Diagnosis Symptoms [DDCDsymptmsMaster \_MO]  4. Inquiry Diagnosis Symptoms [DDCDsymptmsMaster \_IO]  5. Report Diagnosis Symptoms [DDCDsymptmsMaster \_RO] |

|  |
| --- |
| E11: Diagnosis Prescriptions [DDCDprscrptMaster\_BR] |
| Attributes:  1. Diagnosis Reference Number (DP\_DgRef) (References E09) [FK1] [N12]  2. Prescription Given (DP\_PrCode) (References E08) [FK2] [VarChar 8]  3. Prescription Instructions (DP\_Instruc) [A30]  4. Prescription Comment (DP\_Comment) [A30] |
| Comments: 1. Stores information of issued prescriptions |
| Indexes:  1. Primary Key: [1, 2](Constraint Name is DDCDprscrptMaster\_PK) |
| Valid Operations:  1. Add Diagnosis Prescriptions [DDCDprscrptMaster\_AO]  2. Delete Diagnosis Prescriptions [DDCDprscrptMaster\_ZO]  3. Modify Diagnosis Prescriptions [DDCDprscrptMaster\_MO]  4. Inquiry Diagnosis Prescriptions [DDCDprscrptMaster\_IO]  5. Report Diagnosis Prescriptions [DDCDprscrptMaster\_RO] |

|  |
| --- |
| E12: Physicians [CICPhyMaster\_BR] |
| Attributes:  1. Physician Code (PhyCode) [VarChar 8]  2. Physician First Name (PhyFName) [A30]  3. Physician Last Name (PhyLName) [A30]  4. Physician Street Address (PhyStreet) [A30]  5. Physician City (PhyCity) [A30]  6. Physician Credential(license Number) (PhyCred) [VarChar 12]  7. Physician Education Degree (PhyEduDeg) [A30]  8. Physician Specialization (PhySpec) [A30]  9. Physician Province/State (PhyPCode) (References E06) [A5]  10. Physician Country Code (PhyCCode) (References E05) [A5]  11. Physician Telephone (PhyTele) [N10]  12. Physician Email (PhyEmail) [A30] |
| Comments:  1. Stores information about the acting physician  2. Physician Code consists of 8 bytes where the first 8 byte are the physicians ID Number. |
| Indexes:  1. Primary Key: [1](Constraint Name is CICPhyMaster\_PK)  2. CICPhyMaster\_NX1 on [2, 3] |
| Valid Operations:  1. Add Physicians [CICPhyMaster\_AO]  2. Delete Physicians [CICPhyMaster\_ZO]  3. Modify Physicians [CICPhyMaster\_MO]  4. Inquiry Physicians [CICPhyMaster\_IO]  5. Report Physicians [CICPhyMaster\_RO] |

**Database design Incentive:**

The custom schema we will implement will have a great design to it as we take into account, ahead of time, the ways we would map different but important entities to each other. We solve the mapping issue by introducing a matrix entity that maps and connects the symptoms of a disease with what the disease would be after the diagnosis summary is completed. This would allow for more efficiency and flexibility in the context of future designs and implementations as we would not have to worry about getting and displaying information that is within our database for real time use while querying those different tables

## Operations Design

### In this section, operation specifications for selected operations in the system are provided. The algorithms needed for some of the operations (for different entities) are similar. Therefore, in the interest of brevity, instead of repeating the same pseudo code for these similar operations, the following generic operation outlines will be referenced.

The Operations Design will proceed under the following captions:

### • Generic Pseudo-codes

### • O-SPEC

### 4.1 Pseudo-Code

### 4.1.1 Generic Add:

START:

WHILE (User Wishes to Continue)

Accept Key Field(s);

Check Record Absence or Existence in the primary file;

IF (Record Absent)

Accept Non-Key Fields;

Validate Non-Key Fields based on Validation Rules;

WHILE (Any Error Exists)

Re-Display Non-Key Fields for Possible Update

Display appropriate error message(s);

Validate Non-Key Fields based on Validation Rules;

END-WHILE;

Re-Display full Record for confirmation;

IF (Confirmation Obtained)

Write New Record to primary file;

Write New Record to file \_\_\_\_\_ via operation \_\_\_\_\_;

END-IF;

ELSE Inform the User that nothing was saved; END-ELSE

END-IF

ELSE Display Message (“Record already exists”); END-ELSE;

Check if User wishes to quit and set an exit flag if necessary;

END-WHILE;

Generate Edit-List;

STOP

### 4.1.2 Generic Inquiry:

START:

While User Wishes to Continue

Prompt the user for Account Number or Account Name;

Prompt the user for preference (Accounts by Account Number or Account Name);

If by Code

Starting at that point in \_\_\_\_\_ Load a Virtual Data

Collection Object with all records until End-Of-File;

Display the Virtual Data Collection Object;

End-If;

If by Description

Starting at that point in \_\_\_\_\_ Load a Virtual Data

Collection Object with all record until End-Of-File;

Display the Virtual Data Collection Object;

End-If;

End-While;

STOP:

### 4.1.3 Generic Modify:

START:

WHILE (User wishes to continue)

Accept Key Field(s);

Check Record Absence or Existence in the primary file;

IF (Record Present)

Retrieve Record and update Audit Log Fields (with before-values);

Display Non-Key Fields for possible Update;

Validate Non-Key Fields based on Validation Rules;

WHILE (Any Error Exists)

Re-Display Non-Key Fields for possible Update;

Display appropriate error message(s);

Validate Non-Key Fields based on Validation Rules;

END-WHILE

Re-Display full Record for confirmation;

IF (Confirmation Obtained)

Update Audit Log Fields (With current values);

Write new Record to file \_\_\_\_\_ via operation \_\_\_\_\_

Update Record in the primary file;

END-IF;

ELSE Inform the User that nothing was saved; END-ELSE

END-IF

ELSE Display Message (“Record does not exist”); END-ELSE

Check if User wishes to quit and set an exit flag if necessary;

END-WHILE

Generate Edit-List;

STOP

### 4.1.4 Generic Delete:

START:

WHILE (User wishes to continue)

Accept Key Field(s);

Check Record Absence or Existence in the primary file;

IF (Record Present)

Retrieve Record;

Display full Record for confirmation;

IF (Deletion Confirmation Obtained)

Update Audit Log Fields (with current-values);

Write New Record to file \_\_\_\_ via operation \_\_\_\_\_

Delete Record from the primary file;

END-IF;

ELSE Inform the User that nothing was saved; END-ELSE;

END-IF;

ELSE Display Message (“Record does not exist”); END-ELSE;

Check if User wishes to quit and set and exit flag if necessary;

END-WHILE;

Generate Edit-List;

STOP

### 4.2 O-Spec

### 4.2.1 Disease Category (E01) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDcMaster\_AO/DICDcMaster\_ZO/DICDcMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Disease Category Table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDcMaster\_BR – Disease Categories (E01)  **Outputs:**  DICDcMaster\_BR – Disease Categories (E01)  **Validation Rules:**  1. All information must be entered by a licensed practitioner  2. PK must a direct representation of the category name  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.2 Disease Category (E01) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDcMaster\_IO/DICDcMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Disease Category Table  Operation Category: Important  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDcMaster\_BR – Disease Categories (E01)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  It will be possible to query Disease Category by Disease Category Code or Disease Category Name  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.3 Disease Definition (E02) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDdMaster\_AO/DICDdMaster\_ZO/DICDdMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Disease Definition table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDdMaster\_BR – Disease Definition (E02)  **Outputs:**  DICDdMaster\_BR – Disease Definition (E02)  **Validation Rules:**  1. All information must be entered by a licensed practitioner  2. Disease Category Code must already exist in DICDcMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.4 Disease Definition (E02) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDdMaster\_IO/DICDdMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Disease Definition table.  Operation Category: Important  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDdMaster\_BR – Disease Definition (E02)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  It will be possible to query Disease Definition by Disease Code or Disease Name  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.5 Disease Symptoms (E03) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDsMaster\_AO/DICDsMaster\_ZO/DICDsMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Disease Symptoms table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDsMaster\_BR – Disease Symptoms (E03)  **Outputs:**  DICDsMaster\_BR – Disease Symptoms (E03)  **Validation Rules:**  1. All information must be entered by a licensed practitioner  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.6 Disease Symptoms (E03) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDsMaster\_IO/DICDsMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Disease Symptoms table  Operation Category: Important  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDsMaster\_BR – Disease Symptoms (E03)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  It will be possible to query Disease Symptoms by Disease Symptom Code or Disease Symptom Name  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.7 Disease-Symptom Matrix (E04) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDsmMaster\_AO/DICDsmMaster\_ZO/DICDsmMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Disease-Symptom Matrix  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDsmMaster\_BR – Disease-Symptom Matrix (E04)  **Outputs:**  DICDsmMaster\_BR – Disease-Symptom Matrix (E04)  **Validation Rules:**  1. All information must be entered by a licensed practitioner  2. Disease Code must already exist in DICDdMaster\_BR  3. Disease Symptom Code must already exist in DICDsMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.8 Disease-Symptom Matrix (E04) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Information Center  Operation Name: DICDsmMaster\_IO/DICDsmMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Disease-Symptom Matrix table  Operation Category: Important  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DICDsmMaster\_BR – Disease-Symptom Matrix (E04)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  It will be possible to query Disease-Symptom Matrix by Disease Symptom Code or Disease Code  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.9 Country (E05) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICCntryMaster\_AO/ CICCntryMaster\_ZO/ CICCntryMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Country table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICCntryMaster\_BR – Country (E05)  **Outputs:**  CICCntryMaster\_BR – Country (E05)  **Validation Rules:**  None  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.10 Country (E05) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICCntryMaster\_IO/ CICCntryMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Country table  Operation Category: Important  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICCntryMaster\_BR – Country (E05)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  It will be possible to query Country by Country Code or Country Name  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.11 State (E06) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICStateMaster\_AO/ CICStateMaster\_ZO/ CICStateMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the State table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICStateMaster\_BR – State (E06)  **Outputs:**  CICStateMaster\_BR – State (E06)  **Validation Rules:**  None  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.12 State (E06) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICStateMaster\_IO/ CICStateMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the State table  Operation Category: Important  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICStateMaster\_BR – State (E06)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  It will be possible to query State by State Code or State Name  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.13 Visiting Patron (E07) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICVpMaster\_AO/ CICVpMaster\_ZO/ CICVpMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Visiting Patron table  Operation Category: Mandatory  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICVpMaster\_BR – Visiting Patron (E07)  CICStateMaster\_BR – State (E06)  CICCntryMaster\_BR – Country (E05)  CICPhyMaster\_BR – Physician (E12)  **Outputs:**  CICVpMaster\_BR – Visiting Patron (E07)  **Validation Rules:**  1. Province/State Code must already exist in CICStateMaster\_BR  2. Country Code must already exist in CICCntryMaster\_BR  3. Physician Code must already exist in CICPhyMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.14 Visiting Patron (E07) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICVpMaster\_IO/ CICVpMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Visiting Patron table  Operation Category: Mandatory  Complexity Rank: 10 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICVpMaster\_BR – Visiting Patron (E07)  CICStateMaster\_BR – State (E06)  CICCntryMaster\_BR – Country (E05)  CICPhyMaster\_BR – Physician (E12)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  1. It will be possible to query Visiting Patron by the following paths:  1.1 By Patron Code  1.2 By Patron First or Last Name  1.3 By Patron State and Patron Country  1.4 By Patron Telephone Number  1.5 By Patron DoB  1.6 By Patron Primary Care  2. Each option will invoke one of six sub-operations (CICVpMaster\_I1, CICVpMaster\_I2, CICVpMaster\_I3,  CICVpMaster\_I4, CICVpMaster\_I5, CICVpMaster\_I6)  3.Utilizes the logical view CICVpMaster\_LV1, which joins CICVpMaster\_BR with CICStateMaster\_BR,  CICCntryMaster\_BR and CICPhyMaster\_BR  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.15 General Prescriptions (E08) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCGpMaster\_AO / DDCGpMaster\_ZO/ DDCGpMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the General Prescriptions  Operation Category: Mandatory  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCGpMaster\_BR – General Prescriptions (E08)  DICDdMaster\_BR – Disease Definition (E02)  **Outputs:**  DDCGpMaster\_BR – General Prescriptions (E08)  **Validation Rules:**  1. Disease Code must already exist in DICDdMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.16 General Prescriptions (E08) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCGpMaster\_IO / DDCGpMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the General Prescriptions table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCGpMaster\_BR – General Prescriptions (E08)  DICDdMaster\_BR – Disease Definition (E02)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.17 Diagnosis Summary (E09) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCDsmmryMaster\_AO/ DDCDsmmryMaster\_ZO/ DDCDsmmryMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Diagnosis Summary table  Operation Category: Mandatory  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCDsmmryMaster\_BR – Diagnosis Summary (E09)  CICVpMaster\_BR – Visiting Patron (E07)  CICPhyMaster\_BR – Physician (E12)  **Outputs:**  DDCDsmmryMaster\_BR – Diagnosis Summary (E09)  **Validation Rules:**  1. Visiting Patron Code must already exist in CICVpMaster\_BR  3. Physician Code must already exist in CICPhyMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.18 Diagnosis Summary (E09) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCDsmmryMaster\_IO/ DDCDsmmryMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the General Prescriptions table  Operation Category: Mandatory  Complexity Rank: 10 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCDsmmryMaster\_BR – Diagnosis Summary (E09)  CICVpMaster\_BR – Visiting Patron (E07)  CICPhyMaster\_BR – Physician (E12)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  1. It will be possible to query General Prescriptions by the following paths:  1.1 By Diagnosis Reference Number  1.2 By Patron Code  1.3 By Physician Code  2. Each option will invoke one of six sub-operations (DDCDsmmryMaster \_I1, DDCDsmmryMaster \_I2,  DDCDsmmryMaster \_I3  3.Utilizes the logical view DDCDsmmryMaster \_LV1, which joins DDCDsmmryMaster \_BR with CICVpMaster\_BR, and  CICPhyMaster\_BR  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.19 Diagnosis Symptoms (E10) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCDsympMaster\_AO / DDCDsympMaster\_ZO/ DDCDsympMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Diagnosis Symptoms  Operation Category: Mandatory  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCDsympMaster – Diagnosis Symptoms (E10)  DICDsMaster\_BR – Disease Symptoms (E03)  **Outputs:**  DDCDsympMaster – Diagnosis Symptoms (E10)  **Validation Rules:**  1. Disease Symptoms Code must already exist in DICDsMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.20 Diagnosis Symptoms (E10) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCDsympMaster\_IO / DDCDsympMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Diagnosis Symptoms table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCDsympMaster – Diagnosis Symptoms (E10)  DICDsMaster\_BR – Disease Symptoms (E03)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.21 Diagnosis Prescriptions (E11) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCDprscrptMaster\_AO / DDCDprscrptMaster\_ZO/ DDCDprscrptMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Diagnosis Prescriptions  Operation Category: Mandatory  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCDprscrptMaster\_BR – Diagnosis Prescriptions (E11)  DDCGpMaster\_BR – General Prescriptions (E08)  DDCDsmmryMaster\_BR – Diagnosis Summary (E09)  **Outputs:**  DDCDprscrptMaster\_BR – Diagnosis Prescriptions (E11)  **Validation Rules:**  1. Prescription Code must already exist in DDCGpMaster\_BR  2. Diagnosis Reference Number must already exist in DDCDsmmryMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.22 Diagnosis Prescriptions (E11) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Disease Diagnosis Center  Operation Name: DDCDprscrptMaster\_IO / DDCDprscrptMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Diagnosis Prescriptions table  Operation Category: Mandatory  Complexity Rank: 8 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  DDCDprscrptMaster\_BR – Diagnosis Prescriptions (E11)  DDCGpMaster\_BR – General Prescriptions (E08)  DDCDsmmryMaster\_BR – Diagnosis Summary (E09)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

### 4.2.23 Physician (E12) – ADD/DELETE/MODIFY

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICPhyMaster\_AO/ CICPhyMaster\_ZO/ CICPhyMaster\_MO  Operation Description: Facilitates Addition/Deletion/Modification of items to the Physician table  Operation Category: Mandatory  Complexity Rank: 7 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICPhyMaster\_BR – Physician (E12)  CICStateMaster\_BR – State (E06)  CICCntryMaster\_BR – Country (E05)  **Outputs:**  CICPhyMaster\_BR – Physician (E12)  **Validation Rules:**  1. Province/State Code must already exist in CICStateMaster\_BR  2. Country Code must already exist in CICCntryMaster\_BR  **Special Notes:**  None  **Operation Outline:**  See generic ADD pseudo-code  See generic MODIFY pseudo-code  See generic DELETE pseudo-code |

### 4.2.24 Physician (E12) – INQUIRY/REPORT

|  |
| --- |
| **Operation Biography:**  System: SVR-DRS  Subsystem: Control Information Center  Operation Name: CICPhyMaster\_IO/ CICPhyMaster\_RO  Operation Description: Facilitates INQUIRY/REPORT of items to the Physician table  Operation Category: Mandatory  Complexity Rank: 10 out of 10  Spec. Author: GUP & AGU  Date: 04/01/2015  **Inputs:**  Input Form  CICPhyMaster\_BR – Physician (E12)  CICStateMaster\_BR – State (E06)  CICCntryMaster\_BR – Country (E05)  **Outputs:**  Monitor |Printer  **Validation Rules:**  None  **Special Notes:**  **Operation Outline:**  See generic INQUIRY pseudo-code  See generic REPORT pseudo-code |

## User Interface Design

The user interface design is based on Schneiderman’s object-action interface (OAI) model for user interfaces. The real benefit of this approach is that it is consistent with the waypeople tend to think: People do not think about the functional intricacies of their daily activities; rather, they think about objects and what they desire to do with them. Because of the natural fit to the typical thought process on the job, user learning will be enhanced.

The menu system will be hierarchical, as represented in the user interface topology chart (UITC). The user interface will be two GUI’s (one for the doctor and one for the patient) with the following features:

• For each subsystem, the menu options will point to each information entity managed in that subsystem.

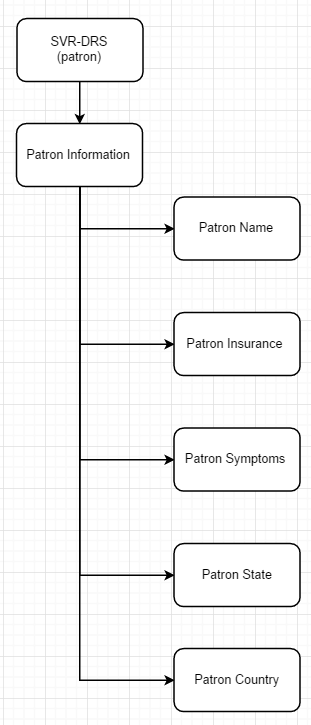
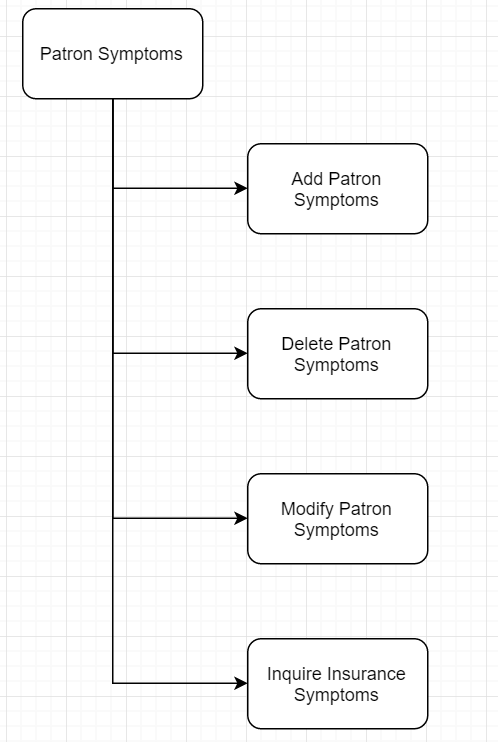
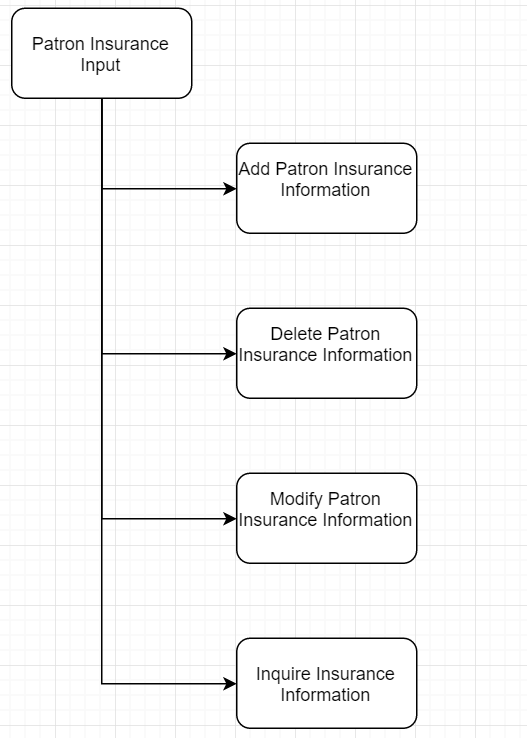
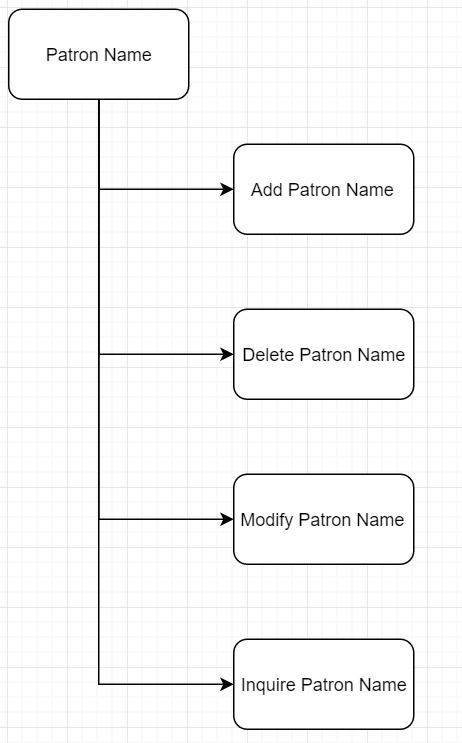
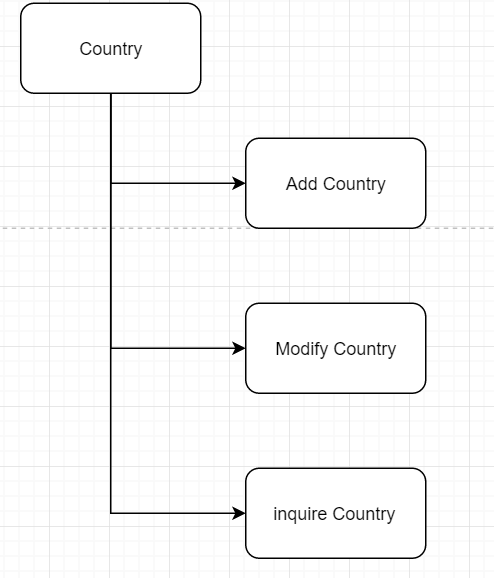
• Any option taken from a subsystem menu will invoke a pop-up menu with the operations relevant to that entity.

The User Interface Design will proceed under the following captions:

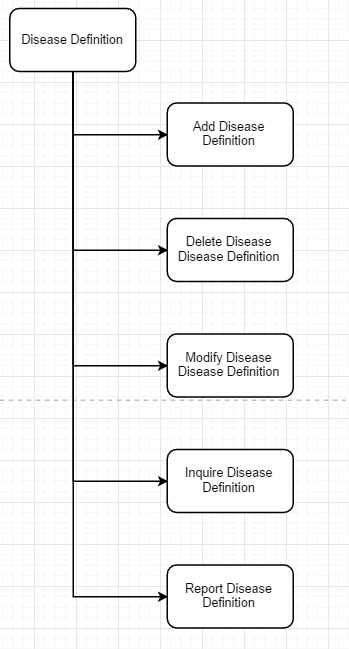
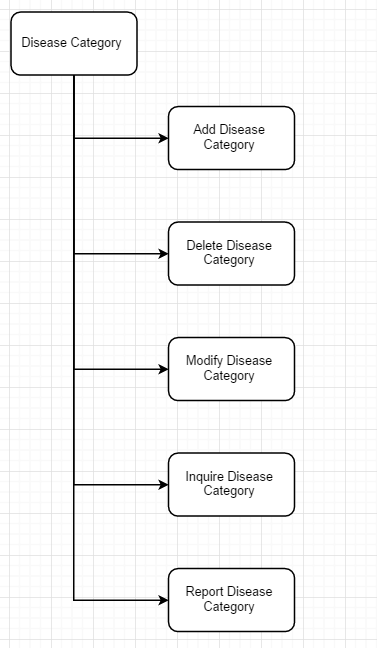
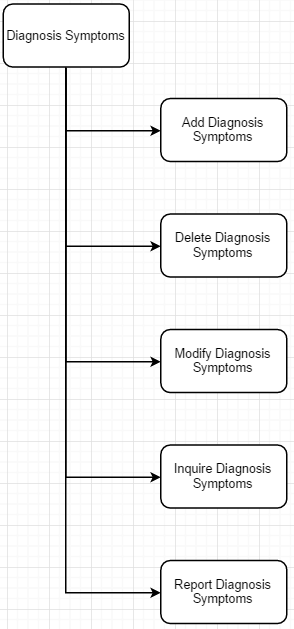
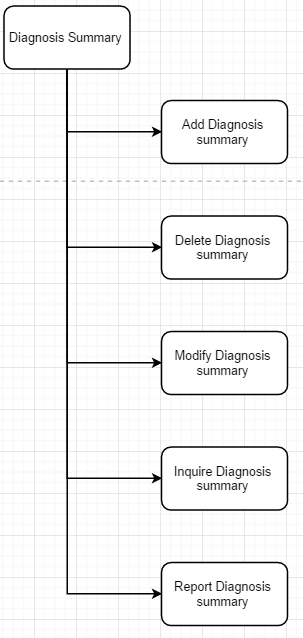
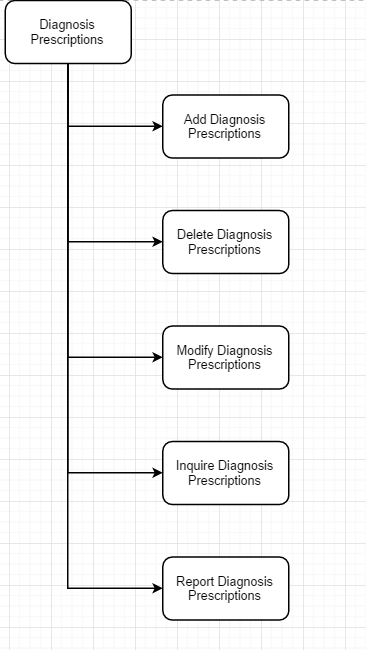
• Patient UI

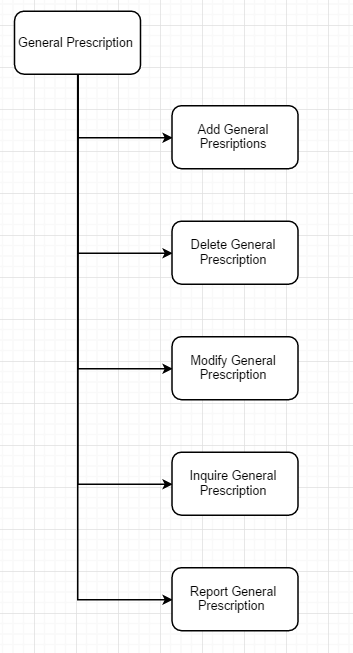
• Doctor UI

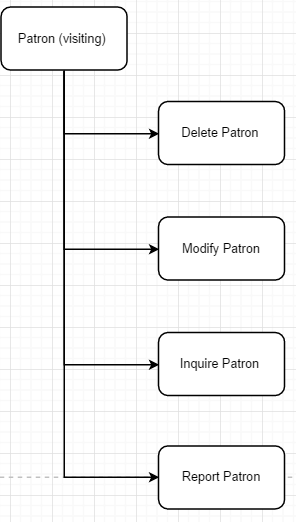
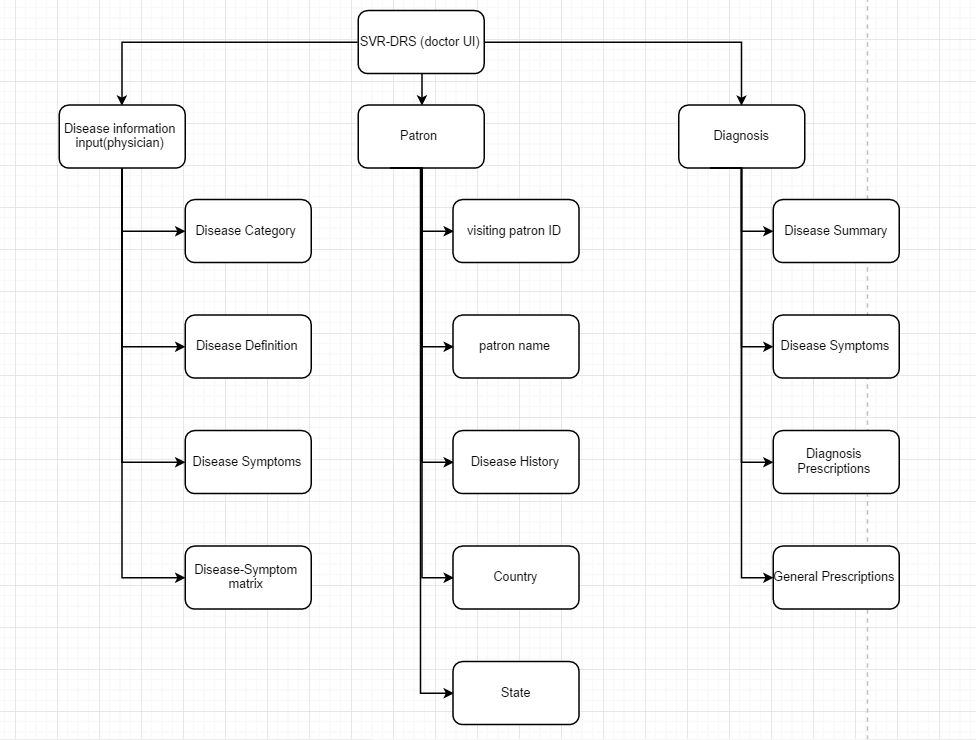
### patient UI



### Doctor UI







## Message and Help Specification

### message Specification

All user messages will be stored in a system-wide message file (as a database table). Unique codes will be given to each message so they can be accessed easier. Messages will be displayed in the form of pop-up messages.

### 6.2 help Specification

The system will host a hyperlink-based help system. Users will access the hyperlink help by clicking appropriate links until they get to the desired help they seek. Additionally, the help system will be organized per the UITC, so that help will be provided operation-by-operation. As the system matures, the help system can be improved to include context-sensitivity.

## 7. Summary and Conclusion

This document provided a basic blueprint for a Disease Diagnosis System’s database and related operations. It includes the following:

* An overview of the system, including problem definition, proposed solution, and system architecture.
* Storage requirements, including draft outlines for the basic information entities comprising the system.
* Operational requirements, including an initial list of basic user operations.
* Business rules, including derivation and procedural rules.

The design specification will use these requirements to prepare a more detailed set of specifications from which the system will be developed.