|  |
| --- |
|  |
| **SOFTWARE REQUIREMENT SPECIFICATION FOR WEB BASED BLOOD DONOR MANAGEMENT** |
|  |

|  |
| --- |
| SWATHY KRISHNA  RMCA B  ROLL NO: 31 |

**1.0 INTRODUCTION**

**1.1** Introduction

Blood Donor Recruitment (BDR) is the process of drawing blood from a voluntary Blood Donor (BD) for future blood transfusion, Wikipedia (2006). In Uganda, blood collection, safety and management is an activity that is carried out by Uganda Red Cross Society (URCS) in partnership with Uganda Blood Transfusion (UBTS). Founded in 1939, URCS is part of the world wide Red Cross Humanitarian Movement whose mission is to mobilize the power of humanity for improving the lives of the vulnerable in Uganda, Muller (2001). URCS fulfills this mission while adhering to the principles of impartiality, neutrality, independence, unity, universality and voluntary service for the Red Cross/Red Crescent Movement. It operates throughout Uganda with 45 branch offices. Besides providing adequate supply of blood for transfusion, URCS is involved in the first aid services, road safety, tracing, disaster mitigation/preparedness, mobilization for routine immunization, HIV homecare, youth empowerment and Community based HealthCare (CBHC).Full implementation of the safe blood policy has called the use of information technology (IT) in providing working solution to the identified challenges. The associated problems with the PCS included delays in accessing historical records, inconsistencies and errors in data entry that stem right from acquisition of data from the blood donors because the exercise is of routine nature and very tedious to the system users. The automation of the system using modern IT has improved the quality of service. Secondly, with the use of IT, now relevant and timely blood donor reports can easily be generated and hence facilitating planning and decision-making.

The study geographically limited itself at the URCS blood donation/collection centers. It focused more on the acquisition, distribution and management of blood units for BDR activities. The study specially emphasized the creation and implementation of an electronic management information system that automated blood donor data acquisition and dissemination of results. This in turn will ease and speeds up the planning, decision-making process because of the timely, secure, confidential and reliable reports.

Following correlative knowledge regarding this research project follows. In the first part of this chapter, the demand and requirements of the proposed system are discussed and analyzed , the entity relations model and the data dictionary. According to this analysis, the specification of the system is defined. foundation This chapter presents the various design techniques and processes available for building web based applications. It explains the design technique chosen, showing its advantages and disadvantages.

1.2 A different approach for designing web based applications

Traditionally, software has been broadly classified into different categories. Some of these categories include real-time software, personal computer software, artificial intelligence software and business software. Web-based systems and applications (WebApps) such as web sites and information processing applications that reside on the Internet or an intranet, require a somewhat different method of development than these other categories of computer software (Pressman, 2000) [xx]. This is because web based systems involve a mixture of print publishing, software development, marketing, computing, internal communications, external relations, art and technology. WebApps are network intensive, content driven, continuously evolving applications. They usually have a short development time, need strong security measures, and have to be aesthetically pleasing. In addition, the population of users is usually diverse. These factors all make special demands on requirements elicitation and modelling.

1.3 Requirements and Analysis

The requirement analysis stage of a software engineering project involves collecting and analyzing information about the part of the organization that is supported by the application. This information is then used to identify the users' requirement of the new system (Conolly et al, 2002) [xx]. Identifying the required functionality of the system is very important as a system with incomplete functionality may lead to it being rejected. A description of the aim of the project is given here along with details of the functional and non-functional requirements for the system. The test sheets for evaluating the completed system are also presented.

**1.3.1 Requirements**

The requirements of the Web-based management information system are to develop:

* a web based front end for entering donated blood details including the donor, his/her blood group, sex, age, and status of the donated blood
* a web based front end for searching the information relating to a given donor or a given blood group;
* a facility to still enter donor and donated blood information via Endnote and also maintain the Endnote database using those details entered via the web front end and
* a facility to produce summary information of donor and donated blood particulars and any other related activities.

**1.3.2 Functional Requirements**

In this research project we aim at developing a system which should improve on the current one with a lot of functionalities and therefore the Major target or goal here is to:

* to develop a blood donor database that can support the five above mention subdatabases that is to say; DonorDB, Donation DB, DiseaseDB, Transfusion DB and Statistical DB
* to develop a client interface that allows privileged users to carry out tasks such as inserting or modifying and deleting data in the database;
* to develop a searching functionality in order to allow normal and privileged users to search the details of a given donor, blood group, stakeholder and if necessary a type of disease common which causes one to need the donated blood
* to fully integrate the Web-based management information system to the WorldWide-Web and hence allow access from any Internet networked terminal and Web browser around the world;
* to develop a facility that can export details entered via the web front end to Endnote as well as import and confidential detail from the Endnote Database;
* to develop a functionality that produces summary information of required data to enhance decision making;
* to embed high security features in the Web DBMS to provide privacy, integrity;
* to allow privileged users to maintain the Web-based management information system by adding/deleting particulars, backing-up or resetting the database and extract online summary in the form of histograms for each donor and lists of free-format comments. Thus a graphical reporting tool should be provided for analyzing the data.
* and finally the system should be flexible enough to store data for several years and also be able provide sufficient User and Administration Guides.

**1.3.3 Non-functional Requirements**

The system must be developed to suit the particular needs of a user-friendly environment. This means that the system must accommodate a clearly understandable user interface as well as clear online help documentation at any stage of the user interaction with the system. A fast response time in obtaining and providing information to the system may also prove to be a significant advantage. In addition to these requirements, the system should also embrace the following requirements:-

**Security:** Each user is required to log in. The system should log staff that has been assigned user names and passwords. The system should be designed to make it impossible for anybody to logon without a valid username and password. Data encryption should be employed to keep the user login name and password secret.

**Reliability:** The system would be used by about 50 staff working at the Red Cross head quarters and also some other many staff in the collaborating clinics and hospitals. The system should have little or no downtime and be able to handle multiple concurrent users.

**Ease of Use:** The general and administrative views should be easy to use and intuitive. Online help and documentation should be provided.

**Performance:** The system should have a quick response time. For the purpose of this research project, this would be defined as less than 5 seconds.

**System and Browser compatibility Testing:** The system should be accessible on the following browsers - Microsoft Internet Explorer 5.5+, NetScape Navigator 6.0+ and Mozilla 1. 3+.

**System requirements:** Red Cross society Uganda has a UNIX server. This system would be designed to run on a minimum hardware configuration of 500MHz x86 machines.

Considering the vast hardware available at the society , this would not pose any problems.

Server Software:

Operating System: UNIX (Sun Solaris), Windows 2000, or Windows XP

PHP version: PHP 5.0+

Web Server: Apache Web Server. 2.0+

Database: MySQL 4.01+

**1.4** Access Level Analysis

In order to take closer look into what the system should do and how, it was necessary to decompose the system’s functionalities based on the user type and levels of access. The three main user groups and access levels are:

* Global User Group (normal access level)
* The Red Cross User Group (privileged access level)
* The Administration (privileged access level)

Therefore, the requirements could be efficiently analyzed depending on the user group and the functionalities they should be allowed to perform.

**1.4.1 Main System Page (Index)**

It is required for the system to provide a Main Page where any Global user (any user within and outside the Red Cross Organization) will be able to access. The main functionality of this page will be to allow any user to search the database by using information such as quantity of donated blood, available blood and the groups, or any other general information which may not be considered confidential. The search capabilities of the main page might not be limited to the exact blood donor, but may for example provide the means for displaying any information that might be relevant but not confidential. The Main Page should also include a Login facility for any privileged or normal user to be able to have access to more advanced functionalities of the System.

**1.4.2 The Red Cross User Group**

When a Red Cross user has successfully logged into the system via the Main Page Login facility, it will be necessary for the system to display a specific menu with all available option that can be carried out. Therefore by taking into account the system requirements, it will be necessary to include options such as Enter donor details, Search donor, Use Endnote Facilities, Produce Summary Information as well as an option that will be related to the appropriate User Guide. A Logout option will also be appropriate for the Red Cross user to be able to logout when desired.

**1.4.3 Entering-Amending Blood donor Details**

For a user to be able to amend and enter into the system’s database it will be essential to take into account that the blood donor system will be integrated to Endnote. Therefore, it will be essential for the system to provide to the user the exact fields as Endnote does for any particular type of details. In addition, when a particular of a given donor has successfully been submitted or amended into the database it will be essential for the system to display the appropriate message (i.e. Blood donor successfully entered into database).

**1.4.4 Searching the Blood Donor Database**

The Searching Facility for the Red Cross user should not differ from the facility that will be provided on the Main Page of the system for all users. Therefore, the Red Cross user will be able to search any type of information in the database using the same way as specified for the Global User.

**1.4.5 Producing Summary Information**

For this requirement it is essential to firstly understand why and when it will be used and to adjust the functionality to best suit these purposes. In order for the system to efficiently produce summary information it will have to provide a menu providing options such as Produce Annual Report, or Produce General Report etc.

**1.4.6 Endnote Facilities**

In order for the system to be effective, it will be necessary for it to be integrated with the Endnote software. Therefore, it will be very significant to accommodate two options that will include Importing blood particulars from Endnote and Exporting blood particulars to Endnote. How this will be done will mainly rely on taking full advantage of particular Endnote filters that are provided for these reasons.

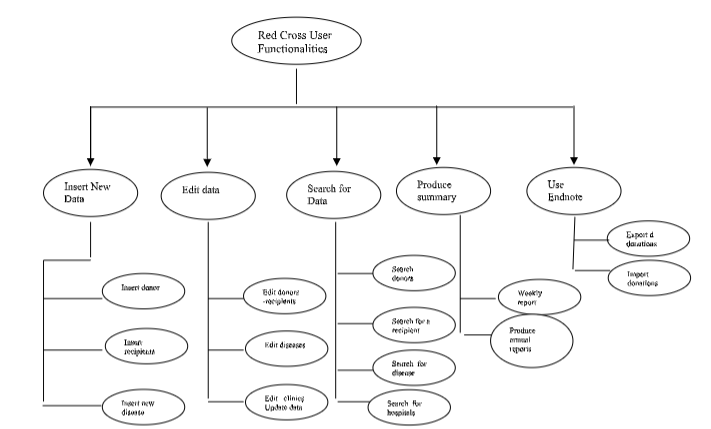
**1.4.7 Administrator**

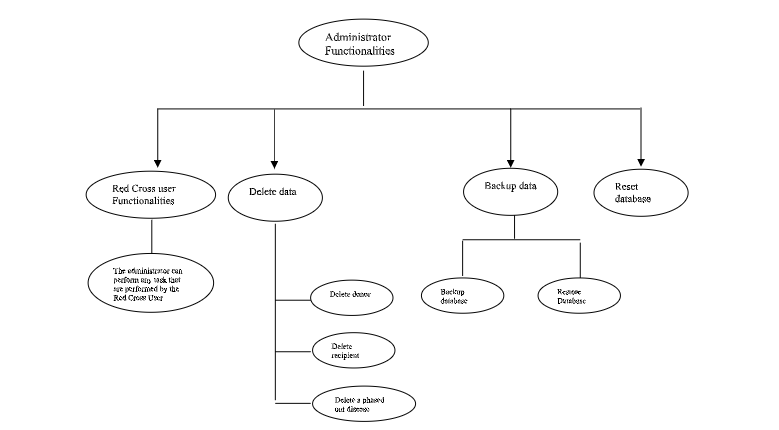
For maintenance purposes it will be of great significance to include advanced Administrator functionalities that can only be accessed by this particular user group. The most reasonable options for an administrator to perform may include tasks such as deleting donors (should not be provided to the Red Cross user group for security reasons), Backing-up and Restoring the database, Resetting the blood donors database etc. In addition to these functionalities the administrator may also be asked to perform tasks related to Red Cross or Global user (i.e. Entering new donors, Searching for a given donor or available blood group) and therefore any functionality provided by the system must be included in the administrator capabilities.

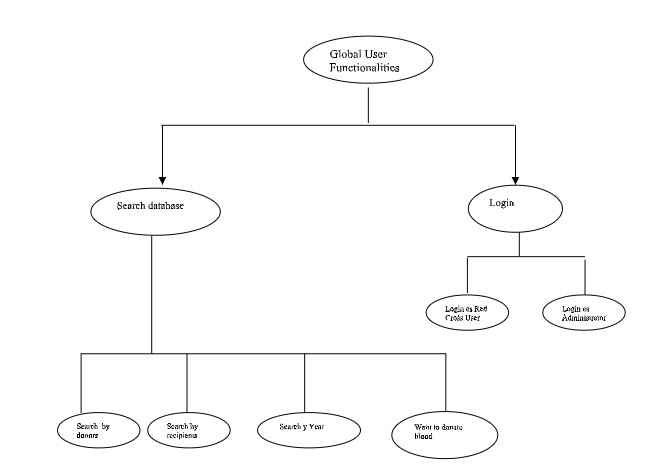
**1.5** Task Structure Diagrams

For the development of a more consistent and effective system, it was essential to firstly identify which information should be included accomplish this, it was first of great significance to group all the relevant tasks (system functionalities) depending on the users.

The way the systems tasks could be efficiently identified was by using a special technique from the Discovery method called Task Structure Sketching (Simons, 2002).  **The Red Cross User Task Structure Diagram ,The Administrator Task Structure Diagram And The Global User Task Structure Diagram** are given below:







**1.6** Tests

The requirement analysis stage involves the design of test cases for the completed system. Test cases are specifications of inputs to the test and the expected output from the system plus a statement of what is being tested (Sommerville, 2004) [xx].

**1.6.1 Designing of test cases**

The test cases designed at this stage are for system testing – testing the integrated system with all the components and functions in place. It is a black-box approach because the tester may not know how the system works but wants to know if it works.

The approach followed at this stage can be termed as requirements-based testing – test cases are designed to test the system requirements

**1.7** Web Engineering

Web engineering is the process used to create high quality Web-based systems and applications (WebApps). Web engineering (WebE) exhibits the fundamental concepts and principles of software engineering by following a disciplined approach to the development of computer-based systems, emphasizing the same technical and management activities (Pressman, 2000) [xx].The design and production of a software product (such as a web application) involves a set of activities or a software process (Sommerville, 2004) [xx]. A software process model is an abstract representation of a software process. Three generic process models usually adopted in projects are

* **The waterfall model –** This has distinct project phases, which can be easily monitored. These phases are requirements specification, software design, implementation and testing.

* **Evolutionary development** - An initial system is developed quickly from abstract specifications. This is later refined with the input of the user to produce a system that meets the users needs. It is an iterative model. Two refinements of this approach are the incremental and the spiral models. The incremental model of evolutionary development delivers software in small but usable “increments”, where each increment builds on those that have already been delivered. The spiral model couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model.

* **Component-based software engineering** - This is based on the existence of a large number of reusable components and is best suited in an object-oriented environment. A process model helps address the complexity of software, minimize the risk of project failure, deal with change during the project and help deliver the software quickly. For this project two process models were considered: 1. Spiral model 2. A waterfall model.

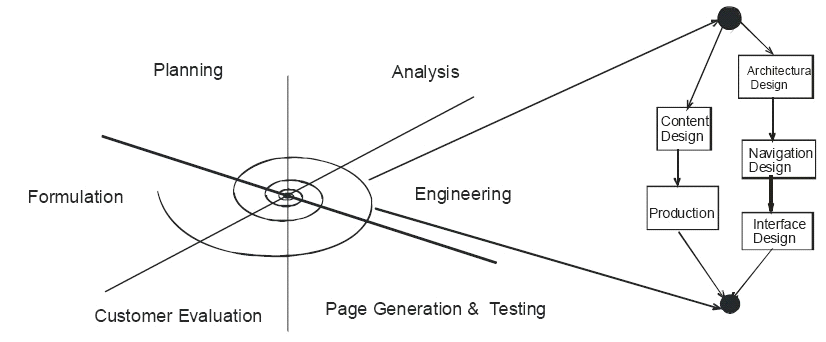
1.8 A WebE Spiral model

The spiral model shown in Fig 4.4 is suggested by Pressman (2000)[xx]. The process consists of 6 main stages, outlined below:

* **Formulation:** This is an activity in which the goals and objectives of the WebApp are identified and the scope for the first increment in the process is established.

* **Planning:** This stage estimates overall project cost, evaluates risks associated with the development effort, prepares a detailed development schedule for the initial WebApp increment and defines a more coarsely granulated schedule for subsequent increments.

* **Analysis:** This stage is the requirement analysis stage for the WebApp. Technical requirements and content items to be used are identified. Graphic design requirements are also identified.



**Fig 4.4: The WebE Spiral Model**

* **Engineering:** Two parallel set of tasks make up the engineering activity. One set involves content design and production, which is non-technical work. This involves gathering text, graphics, and other content to be integrated into the WebApp. At the same time, a set of technical tasks (Architectural design, Navigation design, and Interface Design) are carried out.

* **Page generation:** This is the construction activity that makes use of automated tools for WebApp creation and the content is joined with the architectural, navigation and interface designs to produce executable Webpages in HTML.

* **Customer Evaluation:** During this stage, each increment of the WebEprocess is reviewed. Powell (2002) [xx] presents a waterfall model for web engineering (Fig 5.2).

The advantage of this model is that it helps developers plan most of the work up front.

1. 9 Design Phase

The design involves the production of technical and visual prototypes. This stage has some non-technical aspects such as gathering of web content. Powell (2002)[xx] points out that content gathering can be one of the biggest problems in web projects. This clearly is not the case with this survey application as there is very little content required. For the server side programming and other technical aspects of the design emphasis will be laid on such design concepts and principles as effective modularity (high cohesion and low coupling), information hiding and stepwise elaboration. The goal is to make the system easier to adapt, enhance, test and use (Pressman, 2000) [xx].

**1.9.1 Producing HTML**

There are basically 4 methods of producing HTML –

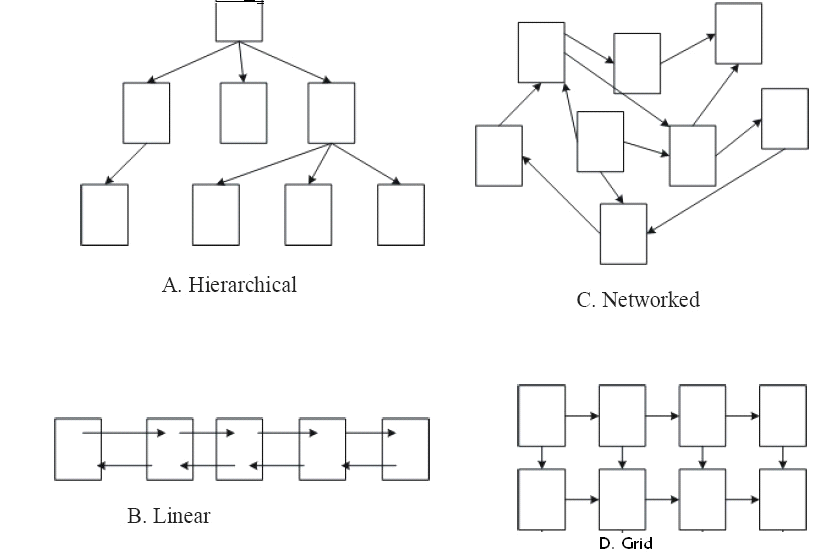
* Coding by hand using a simple text editor
* Translation in which content produced in a tool such as note pad is saved as a HTML document.
* Using a tagging editor that helps fill in the required tags
* Using a “What you see is what you get editor” (WYSIWYG) such as MS FrontPage or Macromedia Dreamweaver©.

All these methods have their advantages and disadvantages.

While coding by hand may be slow and error prone, it does provide great control over markup, as well as help address bugs and new HTML/XHTML elements immediately. At the other extreme, “What You See Is What You Get” (WYSIWYG) editors provide visual representation of a page and require no significant knowledge of HTML or CSS. However they often generate incorrect or less than optimal markup and tend to encourage fixed size presentations that do not separate the look and the structure (Powell, 2003) [xx]. Putting all these into consideration, a tagging editor, HTML-kit© was chosen for this work. While tagging editors can be slow and require intimate knowledge of HTML and CSS, they provide a great deal of control and are a lot faster than hand editing.

1.10 Architectural Design

WebApps fall into 4 main structures. They can be linear, grid, hierarchical, or networked (fig 4.5). In practice most web sites are a combination of some of these structures.



**Fig. 4-5. Navigational Structures of websites/Web Applications ( Lemay, 2000)**

Considering the nature of this web application, a combination of both hierarchical and linear structures will be adopted. The actual survey web pages will have a linear structure while the Admin pages will have a more hierarchical nature.

1.11 Database Design

Database design involves the production of a model of the data to be stored in the database. A data model is a diagram of the database design that documents and communicates how the database is structured. The database design methodology followed in this project is that suggested by Connolly et al(2002)[xx]. Connolly presents quite a detailed guide to designing databases, but not all of those steps may apply here, as this project is not too complex.

The design process is divided into three main stages – conceptual, logical and physical database design. The purpose of the conceptual database design is to decompose the design into more manageable tasks, by examining user perspectives of the system. That is, local conceptual data models are created that are a complete and accurate representation of the enterprise as seen by different users. Each local conceptual data model is made up of entity types, relationship types, attributes and their domains, primary keys and integrity constraints. For each user view identified a local conceptual data model would be built. (Connolly et al, 2002) [xx]. In building the conceptual data model, a data dictionary is built to identify the major entities in the system.

An entity relationship (ER) diagram is used to visualize the system and represent the user’s requirements. The ER diagram is used to represent entities and how they relate to one another. The ER diagram also shows the relationships between the entities, their occurrence (multiplicities) and attributes. Following the view integration approach, a different data model (ER diagram) is made for each user

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** |  |  |
| Donors | A person who donates blood |  |  |
| Recipients | A person who receives blood |  |  |
| Diseases | The diseases which are found in the infected donated blood |  |  |
| Blood group | The blood that is donated by the donors |  |  |
| Hospital/Clinic | Hospitals to which donated blood is distributed |  |  |
| Staff | Red Cross staff |  |  |
| District | Districts from which donors and recipients originate from |  |  |

**Table 4.1: Data Dictionary**

**1.11.1 Conceptual Database Design**

In this stage, a local conceptual data model is built for each identified view in the system. A local conceptual data model comprises of entity types, relationship types, attributes and their domains, primary and alternate keys, and integrity constraints. The conceptual data model is supported by documentation such as a data dictionary.

The entity types are the main objects the users are interested in. Entities have an existence in their own right. Entity types are identified and their names and description are recorded in a data dictionary. Care is taking to ensure that all relationships in the users requirements specification are identified.

An Entity-Relationship diagram is used to represent the relationship between entities. The multiplicity of each relationship is included. This is because a model that includes multiplicity constraints gives a better representation of the enterprise. Relationship descriptions and the multiplicity constraints are recorded in the data dictionary. Each model is validated to ensure it supported the required transactions.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Data**  **Type** | **Size** | **Nulls** | **Multi valued** |
| Donors | donorId (PK)    -dNames    -sex     * dob      * distId (FK)      * doreg | Donor identification  number Donor’s names    Donor’s sex    Date of birth    District of origin    Date of registration | Text    Text    Text    Date    Int    Date | 8    30    6    30    3    30 | No    No    No    No    No    No | No    No    No    No    No    No |
| Recipients | -rId (PK)    -rNames    -sex     * dob      * distId (FK)      * doreg | Recipient’s identification number  Recipients names    recipient’s sex    Date of birth    District of origin    Date of registration | Text    Text    Text    Date    Int    Date | 8    30    6    30    3    30 | No    No    No    No    No    No | No    No    No    No    No    No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| Diseases | -dId (PK)    -dNames    -drating | Disease identification  number Disease names    Disease rating on how  people are infected from it | Text    Text    text | 8    30    20 | No    No    No | No    No    No |
| Blood | bGroup(PK)    donorId (FK)    rId (FK)    status | Blood group    Donor identification number  recipient identification number  status of the donated blood whether infected or not | Text    Text    Text    text | 2    8    8    15 | No    No    No    No | No    No    No    No |
| Hospital/  Clinic | hId (PK)    hNames    distId (FK) | Hospital identification number  Hospital name    District identification number | text    text    int | 8    100    3 | No    No    No | No    No    No |
| Staff | staffId (PK)    staffNames    sex    dob    department | Staff identification number  Staff names    Sex    Date of birth    Department to which the staff belongs | text    text    sex    date    text | 8    50    6    15    100 | No    No    No    No    No | No    No    No    No    No |
| District | distId distName | District number  District name | int text | 3  100 | No  No | No  No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Multiplicity** | **Relationship** | **Entity Name** | **Multiplicity** |
| Donors | 1 | Donates | Blood | 1 |
| Recipients | 1 | Receives | Blood | 1 |
| Diseases | 1 | Contained in | Blood | 0 ..\* |
| Blood | 1 | Donated by | Donor | 1 ..\* |
| Hospital/  Clinic | 1 | Receives | Blood | 1 ..\* |
| Staff | 1 | Registers | Donors | 1 ..\* |
| District | 1 | Has | Recipients | 1 ..\* |

Table 4.2: An extract from the data dictionary showing a description of the relationships between the entities.

**1.11.2 Logical Database Design**

The process of logical database design constructs a model of the information used in an enterprise based on a specific data model, such as the relational model, but independent of a particular DBMS and other physical considerations (Connolly et al, 2002)[xx]. The logical database design consists of an ER diagram, a relational schema, and any supporting documentation for them. In the logical data model, all attributes of entities are primitive.

Producing a logical data model involves normalization. The aim of normalization is to eradicate certain undesirable characteristics from a database design. It removes data redundancy and thus prevents update anomalies. Normalization helps increase the clarity of the data model.

Integrity constraints are imposed in order to protect the database from becoming inconsistent. There are five types of integrity constraints – required data, attribute domain constraints, entity integrity, referential integrity and enterprise constraints. The resulting relations are validated using normalization. For this project, producing relations in third normal form (3NF) will suffice. Non-relational features, such as many-to-many relationships and some one-to-one relationships, are removed from the conceptual data model. The design is also reviewed to make sure it meets all the transaction requirements.

1**.11.3 Physical Database Design**

Physical database design translates the logical data model into a set of SQL statements that define the database for a particular database system. In other words, it is the process of producing a description of the implementation of the database on secondary storage. It describes the base relations and the storage structures and access methods used to access the data effectively, along with associated integrity constraints and security measures. The target DBMS in this case is MySQL.

The following translations occur:

* Entities become tables in MySQL.
* Attributes become columns in the MySQL database.
* Relationships between entities are modeled as foreign keys.

2.0 REFERENCE

* [www.google.com](http://www.google.com/)
* [www.sourcecodeonline.com](http://www.sourcecodeonline.com/)