PATIENTS INFORMATION MANAGEMENT SYSTEM FOR NATIONAL STD / AIDS CONTROL PROGRAM

W. V. D. D. S WARAKAGODA
BIT Registration No: R041314
Index No: 0413143

Supervisor(s) : Mr Muhammed Nilufer Nilar
Mr. Janitha Chanaka Basnayake

2009/ 2010

BIT

UCSC

This dissertation is submitted in partial fulfillment of the requirement of the Degree of Bachelor of Information Technology (External) of the University of Colombo School of Computing.

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

..... Date : 06-08-2010

WVDDS Warakagoda

Countersigned by:

..... Date : 06-08-2010

Muhammed Nilufer Nilar

..... Date : 06-08-2010

Janitha Chanaka Basnayake

Abstract

National STD/AIDS Control Program (NSCP) is one of the major programs of Health Ministry in Sri Lanka. Its aim is to control and prevention of sexually transmitted diseases by giving treatment and consultancy services through outdoor clinics.

The details of the patients such as registration, consultancy, categorization, appointments and visit of clinic etc records were maintained manually. The system which was to be implemented in lieu of manual process was a very efficient, time saving and reliable system. Registration of patients for the clinic, consultancy records, categorization of patients, appointment and visit details of patients and generation of various statistical reports will be automated by the system.

To do this, a clear statement of the requirements was obtained from the relevant personnel and proposed a system, which conformed to the requirements, and which was best optimized to effectively influence the process to increase the efficiency of development activities.

The system was totally web based and platform independent (Microsoft® Windows, UNIX and Linux flavors). It can be accessed from anywhere in the world. The system architecture was based on the Model View Controller (MVC) design pattern. Unified Modeling Language (UML) was used to model the system. PHP and Symfony framework had been used for implementing the system.

Acknowledgement

In fulfilling this project, I wish to express my heartfelt thanks and appreciation to my supervisor Mr. Nilufer Nilar for all his ideas and suggestions and especially for his guidance throughout the difficulties encountered during the course of the project.

Also I would like to thank and give my heartiest appreciation to the director of NSCP and Mr Janitha Chanaka Basnayaka who is my supervisor from the client side. . Without his great support accomplishing this project would not have been a reality.

My sincere and heartfelt thanks are going to my parents, my brother and my friends who help me in various ways to accomplish this project.

Finally I would like to thank all others who helped me in various ways, whose names are not mentioned.

Table of Contents

DECLARATION ..... ERROR! BOOKMARK NOT DEFINED.

ABSTRACT III

ACKNOWLEDGEMENT IV

TABLE OF CONTENTS V

LIST OF FIGURES VII

LIST OF TABLES IX

LIST OF ACRONYMS X

1. INTRODUCTION 2

1.1 Overview of Client and Project Background 2
1.2 Problem Specification 3
1.3 Motivation for Having Patients Information Management System 3
1.4 Scope and Objectives of Patients Information Management System 4
1.4.1 Scope of the Project *4*
1.4.2 Objectives of the Project *5*
1.5 User Requirements of Patients Information Management System 6
1.6 System Requirements of Patients Information Management System 7
1.7 Structure of the Project Dissertation 8

2. ANALYSIS 12

2.1 Initial Situation of the Manual System 12
2.2 Goal of the Proposed Web Based System 13
2.3 Existing Similar Systems 13
2.4 Requirement Gathering and Analysis 13
2.4.1 Functional Requirements *14*
2.4.2 Non- Functional Requirements *17*
2.5 Methodological Approach of System Development 18
2.5.1 Use Case Diagram *18*

3. DESIGN 23

3.1 Alternate System Design Solutions Evaluation 23
3.2.1 Selected Solution Justifications *25*
3.3 Software Process Model 26
3.4 The Selected Development Process Model of the System 30
3.4.1 Phases of Rational Unified Process *31*
3.4.2 Advantages of Rational Unified Process *34*
3.5 Model View Controller (MVC) Architecture 34
3.6 System Design 36
3.6.1 Identify Classes *36*
3.6.2 Class Diagram *37*
3.6.3 Activity Diagrams *39*
3.6.4 Sequence Diagrams *41*
3.6.5 State Chart Diagrams *43*
3.7 Database Design 43
3.7.1 The Entity Relationship Model *44*
3.8 User Interface Design 46
3.9 Main User Interfaces of the System 48

3.9.1 Login Module *49*
3.9.2 Registration Module *51*
3.9.3 Consultancy Module *52*

4. IMPLEMENTATION 55

4.1 IMPLEMENTATION ENVIRONMENT 55
4.2 PLATFORM DEPENDENCIES 56
4.3 DEVELOPMENT TECHNOLOGIES 56
4.4 OPERATING SYSTEM 57
4.5 DATABASE MANAGEMENT SYSTEM 57
4.6 SYMFONY AND DOCTRINE 57
4.7 CODE AND MODULE STRUCTURE 58
4.8 THIRD PARTY LIBRARIES AND SOURCE CODES 64

5. EVALUATION 66

5.1TEST PLAN, TESTING TECHNIQUES AND STRATEGIES 66
5.2 ACCEPTANCE TESTING 68
5.3 SELECTING TEST DATA 69
5.4 SELECTING TEST CASES 69
5.5 OVERALL PROJECT EVALUATION 75
5.6 ERROR DETECTION AND ERROR HANDLING 75
5.6.2DATA REDUNDANCY ERRORS 76
5.6.3 DATA TYPE AND INVALID DATA ERRORS 77

CHAPTER 6 - CONCLUSION 78

6.1 PROBLEMS ENCOUNTERED 79
6.2 GENERAL ACHIEVEMENTS 79
6.3 SUGGESTIONS FOR FUTURE ENCASEMENTS 80
6.4 MISTAKES MADE 80
6.5 LESSONS LEARNT 80

REFERENCES 82

APPENDIX A : SYSTEM DOCUMENTATION 84

APPENDIX B: DESIGN DOCUMENTATION 89

APPENDIX C - USER DOCUMENTATION 93

APPENDIX D - MANAGEMENT REPORTS 115

APPENDIX E – TEST RESULTS 121

APPENDIX F - CODE LISTING 126

APPENDIX G - CLIENT CERTIFICATE 134

LIST OF FIGURES

FIGURE 2.1 MAIN USE CASE DIAGRAM 18

FIGURE 3.1 THE RATIONAL UNIFIED PROCESS 34

FIGURE 3.2 MVC ARCHITECTURE 35

FIGURE 3.3 CLASS DIAGRAM 38

FIGURE 3.4 ACTIVITY DIAGRAM FOR PATIENT REGISTRATION 39

FIGURE 3.5 ACTIVITY DIAGRAM FOR SYSTEM LOGIN 40

FIGURE 3.6 SEQUENCE DIAGRAM FOR PATIENT REGISTRATION 41

FIGURE 3.7 SEQUENCE DIAGRAM FOR USE CASE VIEW APPOINTMENT LIST 43

FIGURE 3.8 SEQUENCE DIAGRAM FOR SYSTEM LOGIN 43

FIGURE 3.9 ER DIAGRAM 45

FIGURE 3.10 MAIN MENU BAR 46

FIGURE 3.11 PATIENT REGISTRATION FORM 47

FIGURE 3.12 LIST VIEW 47

FIGURE 3.13 ACCORDION MENU 48

FIGURE 3.14 LOGIN PAGE OF THE SYSTEM 49

FIGURE 3.15 INVALID LOGIN BY USER 49

FIGURE 3.16 MAIN MENUS OF THE SYSTEM 50

FIGURE 3.17 LIST VIEW FOR SEARCHING, ADDING, EDITING AND DELETING DATA 50

FIGURE 3.18 MAIN REGISTRATION FORM 51

FIGURE 3.19 CONTACT DETAILS TAB IN MAIN REGISTRATION FORM 51

FIGURE 3.20 SEARCH APPOINTMENT DETAILS OF PATIENTS FOR A PARTICULAR DAY 52

FIGURE 3.21 APPOINTMENT LIST OF PATIENTS 52

FIGURE 3.22 SEARCH EPISODE DETAILS OF A PATIENT 52

FIGURE 3.23 EXISTING EPISODES OF THE PATIENT 53

FIGURE 3.24 STD EPISODE CARE DETAILS 53

FIGURE 4.1 1 INITIAL SYSTEM STRUCTURE 59

FIGURE 5.1 TEST PLAN SCOPE 67

FIGURE 5.2 BLANK USERNAME AND PASSWORD ENTERED BY THE USER 75

FIGURE 5.3 SERVER SIDE AUTHENTICATIONS FOR INVALID LOGIN 76

FIGURE 5.4 TRYING TO ADD DUPLICATE USER GROUP NAME 76

FIGURE 5.5 DATA TYPE AND INVALID DATA ERRORS 77

LIST OF TABLES

TABLE 2.1 FLOW OF EVENTS FOR LOGIN USE CASE 20

TABLE 2.2 FLOW OF EVENTS FOR REGISTER NEW PATIENT USE CASE 20

TABLE 2.3 FLOW OF EVENTS FOR CREATE USER ACCOUNT USE CASE 21

TABLE 2.4 FLOW OF EVENTS FOR VIEW APPOINTMENT LIST OF PATIENTS USE CASE 21

TABLE 2.5 FLOW OF EVENTS FOR TRACE DEFAULT PATIENT USE CASE 22

TABLE 4.1 HARDWARE CONFIGURATION OF DEVELOPED MACHINE 55

TABLE 4.2 SOFTWARE CONFIGURATION OF DEVELOPED MACHINE 55

TABLE 4.3 THIRD PARTY LIBRARIES USED IN THE PROJECT 64

TABLE 5.1 TEST CASES FOR LOGIN MODULE 69

TABLE 5.2 TEST CASES FOR ADMIN MODULE 70

TABLE 5.3 TEST CASES FOR CONSULTANCY MODULE 71

TABLE 5.4 TEST CASES FOR REGISTRATION MODULE 73

TABLE 5.5 DATA TYPE VALIDATION 74

LIST OF ACRONYMS

PIMS - Patients Information Management System

NSCP - National STD/AIDS Control Program

STD - Sexually Transmitted Disease

UML - Unified Modeling Language

RUP - Rational Unified Process

OOD - Object Oriented Design

OOP - Object Oriented Programming

ERD - Entity Relationship Diagram

PHP – Hypertext pre processor

SDLC - System Development Life Cycle

IDE - Integrated Development Environment

SQL - Structured Query Language

MVC - Model View Controller

RAD - Rapid Application Development

RAM - Random Access Memory

ERD - Entity Relationship Model

UI - User Interface

Chapter 1 - Introduction

1. Introduction

This chapter gives the background to the project, project scope and describes the structure of the project. The motivation for the project is argued here. Then a brief introduction to the project is provided indicating its objectives and scope. Finally, a paragraph containing an outline of the remaining chapters is recommended.

1.1 Overview of Client and Project Background

The "Sexually Transmitted Diseases Control Program" (STD) was initiated as the "Venereal Diseases Control Program" in the island in 1921 by the Government of Sri Lanka through the Ministry of Health and was established at clinics of the outdoor department in the General Hospital of Colombo. In 1985 HIV/AIDS control program too was integrated with the STD program as the "National Sexually Transmitted Disease and AIDS Control Program". Its aim is to control and prevention of sexually transmitted diseases giving treatment and consultancy services through outdoor clinics. The National STD / AIDS Control Program provides clinic services for diagnosis and treatment of patients, program management which consists of surveillance, monitoring and evaluation and operational research and link up with several organizations outside the health sector who work towards in preventing the transmission of HIV infection. [WWW1]

Currently all the details of the patients such as registration, categorization, consultancy, appointments and clinic visits, report generation etc are maintained manually. They have recently carried out an investigation to increase the efficiency and accuracy of the manual operations. After analyzing the results the directors have found that the most effective areas need to be automated through a computerized system.

1.2 Problem Specification

Currently the patients details are managed manually. The registration, consultancy and clinic information of patients are maintained in flat files. Querying of patient specific data is very slow and consumes more time. Once a patient is checked by the doctor, all the flat files related to the diseases and next appointment dates etc have to be updated by the PHI or sister. Average of forty to fifty patients files are updated manually every day. This is a time consuming task and employees get tired when they finish the task and cause mistakes. The existing manual system is silent in many ways and do not generate valuable information when necessary and employee satisfaction is vastly decreasing. For example, generating various statistical reports in terms of patients and various diseases is a tedious task and currently those are not accurate.

The decision was taken to move on to a new web based system to manage patient information and related activities are inevitable in considering the drawbacks mentioned above. So the proposed system which is to be implemented in lieu of manual process is very efficient, time saving and reliable.

1.3 Motivation for Having Patients Information Management System

This has been identified as a major requirement for the organization to automate the existing manual system to a web based system. Introducing a web based system, users are able to access from anywhere.

The proposed system will enable better maintenance of large amount of patients' details, their consultancy history, adding, updating, deleting data, easy access to data, generation of various types of reports, providing different access levels to different users, sorting, searching for specific data etc. Also by having proposed system, all functionalities can be done in a more reliable and efficient way.

1.4 Scope and Objectives of Patients Information Management System

1.4.1 Scope of the Project

The proposed system is focused to automate the following main areas of the manual system at general clinic in Colombo, which is the main requirement of the client.

1. Registration of patients
2. Categorization of patients.
3. Handling the appointment and visit details of patients.
4. Tracking default patients of the clinic.
5. Maintaining the consultancy details of patients.
6. Generating various statistical reports in yearly and quarterly.
7. Handling the administrative functions in the system. For example, creating user groups, user accounts etc.

So that proposed system will be divided in to the following modules.

1. Administration
2. Registration
3. Consultancy
4. Report
5. Login

Brief description for the above modules is given bellow.

Administration Module

All the administration functions of the system such as creating user accounts, creating user groups, setting access rights, change password etc are handled by this module.

Registration Module

Registration of patients, searching records, appointment and visit details of the clinic, tracing default patients etc are handled by the Registration module.

Consultancy Module

Maintaining all the consultancy records of patients are handled by the consultancy module.

Report Module

Various statistical reports and normal day to day reports are generated by the report module. These reports will help to make various decisions.

Login Module

The system login of the users are handled by this module.

1.4.2 Objectives of the Project

1. Identify the requirements of the client
    - Study the existing manual system and find out how that can be improved by using a computerized system.
    - Requirements are gathered by using fact gathering techniques such as interviews, observations
    - Research about the similar systems.

2. Design the system
    - Using the object oriented design techniques, identify the elements that could be converted to objects and identify their attributes and behaviors.
    - Model View Controller (MVC) design pattern has been followed to design the entire system architecture.
    - Using a modeling language like Unified Modeling Language (UML), model the architecture of the system.
3. Implement the system

- System will be implemented by following the Object Oriented Programming concepts (OOP) and the Model View Controller (MVC) design pattern.
- PHP and the Symfony framework have been used for writing codes. Symfony is a free open source PHP framework which supports to implement the MVC pattern.
- MySQL database server has been used for storing data.

4. Test the system
    - Create a proper test plan for testing process.
    - Write all the possible acceptance test cases which will be covered the entire system functionalities.
5. Provide a user guide and system document.
6. Apply knowledge gathered during BIT degree program.
7. Learn and apply practical software development methodology.
8. Complete the project before the given deadline.

1.5 User Requirements of Patients Information Management System

1. Privacy of the patients' information must be protected by the system. So all the sensitive data such as name, contact numbers, address, must be encrypted before saving to the database.
2. System must be user friendly and the user interfaces should be clear and consistent. All the messages should be clearly understood and should not be in technical terms.
3. Proper and accurate reports should be generated by the system.

1.6 System Requirements of Patients Information Management System

Hardware Requirements

- Processor - Intel Pentium IV or higher
- RAM – 512MB or higher
- Hard Disk-20 GB or higher
- CD ROM Drive

Software Requirements

Operating Systems

- Windows 2000 / XP , Vista, Windows 7
- Linux

Programming Languages

- PHP 5 or later

Web Server

- Apache HTTP server

Database Server

- MySQL 5 or later

Web Browsers

- IE 6 or later / Mozila FireFox 3 or later

Development Tools

- Eclipse IDE
- Netbeans® ID
- Macromedia Dreamweaver 8
- Rational Rose

Other

- XAMPP
- Symfony (Open source web PHP framework)
- JQuery Library (Open source javasript library)

1.7 Structure of the Project Dissertation

Following is a brief description about the forth coming chapters and their contents.

Chapter 1. Introduction

This chapter illustrates the system introduction, reason for introducing this system, motivation of this project, scope, and objectives. At the end of this chapter, user requirements and system requirements are discussed. It also describes a paragraph containing an outline of the remaining chapters.

Chapter 2. Analysis

This chapter describes about detailed and specific requirements of the project. Project goals and determination of goals and the system environment are also discussed. Existing similar systems, project success factors, functional and non functional requirements, background and main literature review are listed and described here.

Chapter 3. Design

This chapter describes the methodical approach which is used to design the system and it includes some of the important diagrams such as Activity, Sequence, Class, ER diagrams. User interface design is also described here.

Chapter 4. Implementation

This chapter describes the implementation process of the system. Functions and implementation framework details related to the development of each module will be detailed.

Chapter 5. Evaluation

This chapter describes the testing process and the test plan which is used during this project and all the test data with their outputs. It describes the evaluation process of the project.

Chapter 6 . Conclusion

This chapter will discuss dead ends and difficulties encountered during the project's life cycle and how it has been deviated from the original project and the road ahead for the project in the future. Also include the summary of the project.

References

This section will acknowledge all the references used in the project.

Appendix A . System Documentation

Technical documentation is included here and would provide program installation, and execution details.

Appendix B . Design Documentation

Design diagrams are included here.

Appendix C. User Documentation

All the functionalities which will be used by the system users are explained here. Relevant screen will also be provided to explain the functionalities.

Appendix D. Management Reports

Various management reports are included here.

Appendix E. Test Results

Test cases and test results are included here.

Appendix F. Code Listing

Most important code segments of the system modules are listed here.

Appendix G. Client Certificate

Client certification document is attached here.

Chapter 2 – Analysis

2. Analysis

Requirements analysis is the process of understanding the needs and expectations from the proposed system or application and is a well-defined stage in the software development life cycle (SDLC). Requirements are a description of how a system should behave or a description of system properties or attributes. It can alternatively be a statement of 'what' an application is expected to do.

The software requirement analysis process covers the complex tasks of eliciting and documenting the requirements of all the users, modeling and analyzing these requirements and documenting them as a basis for system design.

2.1 Initial Situation of the Manual System

As mentioned in the earlier chapter, National STD/AIDS Control Program, currently the details of the patients such as registration, categorization, consultancy, and appointments, reports etc are maintained manually. Various forms, documents have been used to record day to day activities. Some of the main documents involved with the manual system are listed below.

1. Registration form of patients.
2. STD episode care form.
3. Attendance recording book of patients.
4. Excel reports.

Main drawback of the manual system is to maintain all the manual documents in an efficient way. Various issues of the manual system have been found by the staff and the management of National STD/AIDS Control Program such as

1. Maintaining clinical record files of patients for a longer period of time is very difficult. They have been using various box files and cupboards for this and it will be a problem in the future.

2. Searching a past record file of a patient takes at least fifteen to twenty minutes time and that is a big issue when the number of files to be searched is increased.
3. Keeping clinical details of patients up to date is a tedious and time consuming task in the manual system. STD episode care details of patients, attendance and visit dates etc have to be updated at the end of the day by the PHI or Sister. As far as the users are concerned minimum ten minutes time is taken to update all the relevant documents of a patient.
4. Generating accurate reports in an efficient way. In manual system, it is very difficult to produce the required reports because of the slowness of the data storage and retrieval mechanisms.

2.2 Goal of the Proposed Web Based System

The proposed system which is to be implemented for addressing the main issues identified by National STD/AIDS Control Program. The main goal of the proposed system is to provide a better, reliable service for both employees of the National STD/AIDS Control Program and the patients in an efficient way.

2.3 Existing Similar Systems

Researching about the similar systems is a key factor of any software development project. So this key factor was considered when the Patients Information Management System project started. Clinic Manager EMR 1.0 was studied to get an idea abut the domain and the software features. Clinic Manager is free Electronic Medical Records (EMR) software that was designed to serve small private clinics.

2.4 Requirement Gathering and Analysis

The functional and non-functional requirements of the system were gathered basically through the interviewing, studying the existing manual system and observation. Interviews were conducted with the staff of the National STD/AIDS Control Program to get feedbacks for the proposed system. Some of the requirements provided by end users were

ambiguous, i.e. they were not clear. Those were clarified by meeting them again and asking various questions to get their ideas. The findings were reviewed and presented to the National STD/AIDS Control Program, in order to get the approval from the director board. This effort was most successful in the context of gathering the exact user requirements. A small presentation on how the system would look and feel was conducted. This was done to get the end user suggestions and their ideas for more improvement to the interfaces.

2.4.1 Functional Requirements

Functional requirements define a function of a software system or its components. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. [WWW2]

Following is the list of functional requirements for the proposed system.

- Register new patient
  New patient is registered to the system by PHI / Sister. Necessary information of the patient is gathered and the clinical card is issued to the patient.

- Create STD Episode
  System should create a new STD episode when a new patient is registered or an existing patient comes to get treatment again.

- Search patient
  System should be able to search details of the patients by their names or patient numbers.

- Create patient category
  System should be able to create different categories of patients.

- Mark and create visit

  PHI / Sister should be able to update the patients' visit and next date for the clinic.

- View appointment list of patients

  PHI / Sister should be able to view the appointment details of patients.

- Export appointment list as a PDF file

  Appointment list should be exported as a PDF file.

- View attendance list of patients

  PHI / Sister should be able to view the attendance details of various kinds of patients.

- Export attendance list as a PDF file

  Attendance list should be exported as a PDF file.

- Enter STD episode care details

  PHI / Sister should be able to update the STD episode care details of patient.

- Trace default patient

  System should be able to trace the patients who are in the default status. Patients become to the default status if they did not come to the given clinic date.

- Create input question category for STD episode care

  Authorized persons should be able to add various categories of input questions to the system.

- Create input question for STD episode care
- Authorized persons should be able to add new input questions for a relevant input category.

- Create input results for input questions

  Authorized persons should be able to add the answers / results for selected input questions.
- Log in to the system

  To log in to the system, user must enter a valid user name and password on the log in page.

- Log out

  Users should be able to log out from the system when they needed.

- Create user account

  System administrator is allowed to create new user accounts for the system. The password given by the admin when the account is created can be changed by the user later on.

- Create user group

  System administrator is allowed to create different user groups for the system.

- Assign system modules for user group

  System administrator should be able to assign system modules for user a group.

- Assign rights for user group

  System administrator should be able to assign rights for different modules in the system.

- Create patient number automatically according to the gender and the year

  Patient number should be created automatically according to the year and gender of the patient when the registration is taken place.
- Change password

2.4.2 Non- Functional Requirements

Non-functional requirements define constrains on the services or functions offered by the system and system properties. Non-functional requirements arise through user needs, organizational policies or interoperability with other systems. Some of the non- functional requirements of the proposed system are listed as follows.

Security

Security is a key factor of the system. No one other than the authorized users should be able to login to the system.

Privacy

Privacy of the patients must be protected. Data like patients' name, NIC number, address, and telephone numbers must be saved in encrypted format.

User friendliness

The system should be user friendly and anyone who is not familiar or new to the system should be able to handle the system with less effort. Also system must use simple words, commands such that every user can easily understand. All the user interfaces should have consistent look and feel.

Reliability

Any system failure should not be observed when system is at real environment. End users should be able to rely on the system. There should not be any unexpected errors or the error rate should be very low.

Efficiency

System should handle the services, functionalities efficiently. That is for an operation, it should not take long time such that user has to wait for a long time.

Maintainability

The system should be implemented in a way to utilize re-usability and maintainability of both the design and code. In future alternate functions and requirements should be added to the system without major impacts on design. Coding standards of the system play an important role in this aspect.

2.5 Methodological Approach of System Development

2.5.1 Use Case Diagram

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram (Figure 2.1) displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

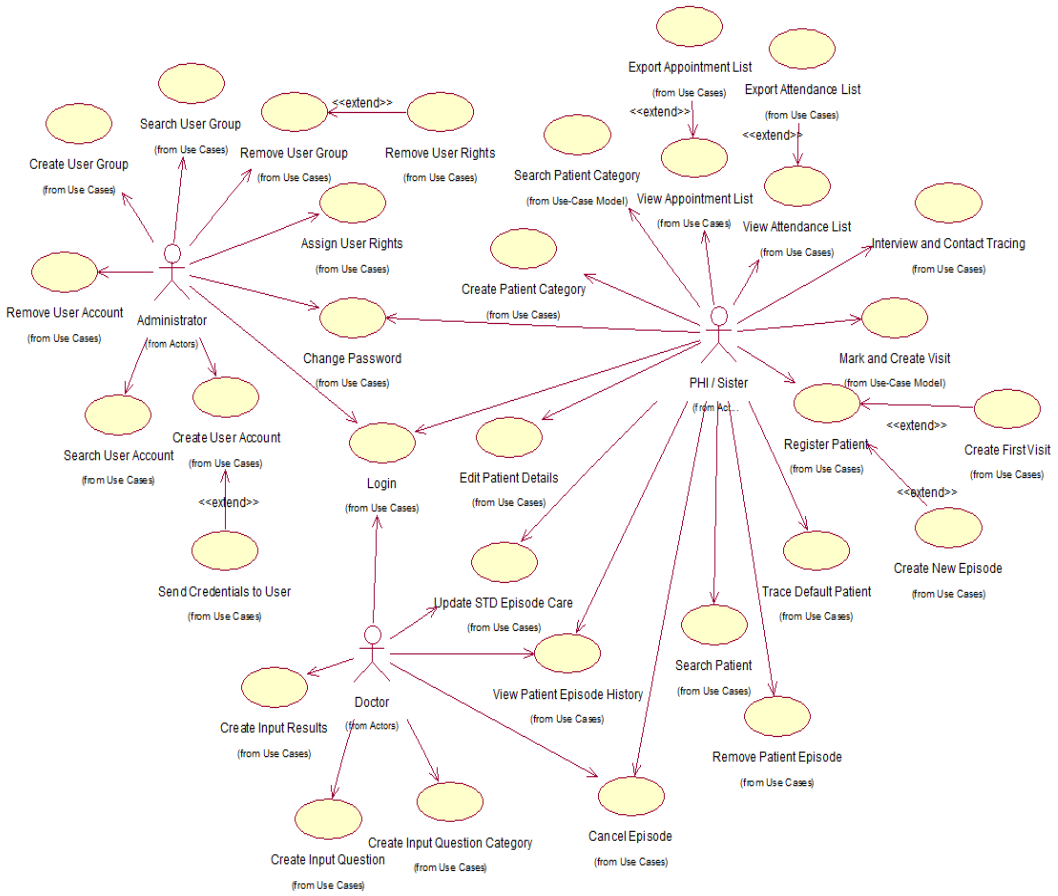

Figure 2.1 Main use case diagram

Table 2.1 Flow of events for Login use case

Use Case	Login
Actor /s	PHI , Sister, Doctors, Receptionist ,Administrator .
Overview	
Login to the system.	
Preconditions	
Users should have a user name and password.	
Flow of Events	
1. Enter the user name and password on the login page. 2. Click on the "Login" button. 3. Display the assigned menus of the system.	
Alternative Flows	
1. Should display an error message if the user name or password are invalid. 2. Should display an error message if the user account is disabled.	
Post Conditions	

Table 2.2 Flow of events for Register New Patient use case

Use Case	Register New Patient
Actor /s	PHI , Sister, Receptionist
Overview	
Register a new patient to the system.	
Preconditions	
Authorized users should login to the system and "Registration" menu should be displayed with relevant menu options.	
Flow of Events	
1. Click on the "New Patient " button which is in the patients' registration page. 2. Display the registration form with the tabs "Personal Information", "Contact Information" and "Reason for Clinic Attendance Information". 3. Current system date should be displayed by default as the registered date. 4. Click on the "Save " button after filling the necessary fields.	
Alternative Flows	
1. Should display error messages if the required fields are blank.	
Post Conditions	

1. Patient number should be created according to the gender of the patient and the year registered. 2. Record is saved to the database

Table 2.3 Flow of events for Create User Account use case

Use Case	Create User Account
Actor /s	Administrator
Overview	
Create an user account for the system.	
Preconditions	
Necessary user groups should have been created.	
Flow of Events	
1. Select "User Accounts" menu option in the admin menu. 2. Display the account creation form. 3. Enter a user name and password for the account. 4. Select the user group that the user gains rights to access, 5. Select the status and click on the "Save" button.	
Alternative flows	
Post Conditions	
User should be added to the system and should be able to access the system if the status is "Enabled".	

Table 2.4 Flow of events for View Appointment List of Patients use case

Use Case	View Appointment List of Patients
Actor /s	PHI , Sister
Overview	
View the list of appointments of a particular clinic day	
Preconditions	
PHI / Sister should log in to the system and "Registration" menu should be displayed with relevant menu options .	
Flow of Events	
1. Click on the " Appointment List" menu option in the registration menu. 2. Enter the date and select the gender . 3. Click on the "View" button. 4. Display the details of appointments. 5. Click on the "Export" button to export the list.	
Alternative Flows	

Should display a message if no records were found.
Post Conditions
The users should be able to export the appointment list as a PDF file

Table 2.5 Flow of events for Trace Default Patient use case

Use Case	Trace Default Patient
Actor /s	PHI , Sister
Overview	
View the list of patients who did not come to the given clinic date.	
Preconditions	
PHI / Sister should log in to the system and “Registration” menu should be displayed with relevant menu options .	
Flow of Events	
1. Click on the “ Trace Default Patients” menu option in the registration menu. 2. Enter the date. 3. Click on the “View” button. 4. Display the list of default patients. 5. Click on the “Export” button to export the list.	
Alternative Flows	
Should display a message if no records were found.	
Post Conditions	
The users should be able to export list as a PDF file	

Chapter 3 - Design

3. Design

In the design phase of the project, Object Oriented Design (OOD) concepts were followed. This technique was used in order to address designing issues like, re-usability, modularity, abstraction, encapsulation, support maintainability etc. Unified Modeling Language (UML) was employed to provide a range of different notations that were used to document the object oriented design.

3.1 Alternate System Design Solutions Evaluation

As mentioned in the analysis chapter, system has been divided into four sub systems . The main alternative ways of designing those sub systems into a one system are described bellow.

Developing the whole system as a web based system

In this approach , the system is installed in one machine called web server and it can be accessed over a network such as the Internet or an Intranet by using a web browser as a client.

One drawback of this approach is, it is required to have a high end computer as the server. However that cost can be covered up by buying some low end client machines because in this approach it is not to have powerful client computers.

Below are some of the core benefits of web based applications.

- **Cross platform compatibility**

  Most web based applications are more compatible across platforms than traditional installed software. Typically the minimum requirement would be a web browser such as Internet Explorer, Firefox, Netscape. These web browsers are available for a multitude of operating systems and so whether users use Windows, Linux or Mac OS.

- **More manageable**

  Web based systems need only be installed on the server placing minimal requirements on the end user workstation. Maintaining and updating the system much simpler as usually it can all be done on the server. Any client updates can be deployed via the web server with relative ease.

- **Reduced cost**

  Web based applications can dramatically lower costs due to reduced support and maintenance, lower requirements on the end user system and simplified architecture.

- **Secure live data**

  Typically in larger more complex systems data is stored and moved around separate systems and data sources. In web based systems these systems and processes can often be consolidated reducing the need to move data around. Web based applications also provide an added layer of security by removing the need for the user to have access to the data and back end servers.

Designing the whole system as a standalone application

In this way all the sub systems of the system are developed as a single desktop based application. Main drawback of this way is, each client machine should be installed the complete application and the relevant software packages. Upgrading the application to a new version involves installations on each workstation, which can become cumbersome for organization.

Users are tied to the specific platform for which the application is written.

When discussing about alternate system design architectures, it is able to use a pattern like one-tier , two-tier or three-tire architecture. By increasing the layers of a application , it is able to increase the system maintainability and the code readability because whole software logic is divided among several layers.

3.2 Selected System Design Solution Evaluation

After considering the advantages and the drawbacks of main alternative solutions , web based approach was selected as the design solution for the system. Three-tier architecture is used with the Model-View-Controller design pattern to design the system.

As far as the system is concerned, web based approach is the best way to follow in terms technical feasibility, economic feasibility, and future changes of the system.

3.2.1 Selected Solution Justifications

The justification regarding the selection of this methodology are justified as follows.

1. Main reason for using a web based system is, all the other STD clinics of government hospitals will be connected to the system which is in the main STD clinic at Colombo. So system should be platform independent and should be able to access from anywhere.
2. PHP has been selected as the programming language for implementing the system. Developing a desktop based application by using PHP is quiet difficult because PHP is not fully supporting for desktop based applications and some other libraries like GTK have to be used with PHP to develop a desktop based application. So after considering the time constrains and the possible technical issues, web based application approach was selected as the design solution.
3. Reduce the cost of system maintenance .
4. Immediate update in case of releasing new versions, As soon as a new version of the web application is released, all the users get it at once as there is only one copy of the application.
5. Web based applications are safe. With traditional applications, the user has to store the data on their PC or laptops, which is not safe at all. While using web applications, your databases are stored safely and backed-up every day.

3.3 Software Process Model

Software process models are used to represent a networked sequence of activities, objects, transformations, and events that embody strategies for accomplishing software evolution. Such models can be used to develop more precise and formalized descriptions of software life cycle activities.

The process model for this system was selected by comparing different types of other process models. The advantages and the disadvantages of the process models were considered with the selected process model for selecting the best option.

There are various process models that can be used to develop the system. Some of the feasible models and advantages / disadvantages are described bellow.

Waterfall Model

Waterfall model approach was the first process model to be introduced and followed widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate process phases. The phases in Waterfall model are: Requirement Specification phase, Software Design, Implementation , Testing and Maintenance. All these phases are cascaded to each other so that second phase is started as and when defined set of goals are achieved for first phase and it is signed off.[WWW3]

Advantages of the waterfall model

- Easy to explain to the user.
- Stages and activities are well defined.
- Helps to plan and schedule the project.
- Verification at each stage ensures early detection of errors / misunderstanding.

Disadvantages of the waterfall model

- Difficult to define all requirements at the beginning of the project.

- Can not go back to the previous stage, for example, if the design phase has gone wrong, things can be very complicated in the implementation phase.
- Clients are not very clear of what they exactly want from the software. Any changes that they mention in between may cause a lot of confusion.
- Small changes or errors that arise in the completed software may cause a lot of problem.

Spiral Model

The Spiral model was designed to include the best features from the waterfall and Prototyping model, and introduces a new component risk assessment. The term "spiral" is used to describe the process that is followed as the development of the system takes place. Similar to the Prototyping model, an initial version of the system is developed, and then repetitively modified based on input received from the customer evaluations. Unlike the Prototyping model, however, the development of each version of the system is carefully designed using the steps involved in the Waterfall model. With each iteration around the spiral (beginning at the center and working outward), progressively more complete versions of the system are built.

Risk assessment is included as a step in the development process as a means of evaluating each version of the system to determine whether or not development should continue. If the customer decides that any identified risks are too great, the project may be halted. For example, if a substantial increase in cost or project completion time is identified during one phase of risk assessment, the customer or the developer may decide that it does not make sense to continue with the project, since the increased cost or lengthened time frame may make continuation of the project impractical or unfeasible . [WWW4]

The Spiral Model is made up of the following steps.

- **Project Objectives**: Similar to the system conception phase of the Waterfall Model. Objectives are determined, possible obstacles are identified and alternative approaches are weighed.

- **Risk Assessment:** Possible alternatives are examined by the developer, and associated risks / problems are identified. Resolutions of the risks are valuated and weighed in the consideration of project continuation. Sometimes prototyping is used to clarify needs.
- **Engineering and Production**: Detailed requirements are determined and the software piece is developed.
- **Planning and Management**: The customer is given an opportunity to analyze the results of the version created in the engineering step and to offer feedback to the developer.

Rapid Application Development Model (RAD)

RAD is a linear sequential software development process model that emphasis an extremely short development cycle using a component based construction approach. If the requirements are well understood and defined, and the project scope is constraint, the RAD process enables a development team to create a fully functional system with in very short period of time. [WWW5]

RAD model consists following phases.

Business Modeling: The information flow among business functions is defined by answering questions like what information drives the business process, what information is generated, who generates it, where does the information go, who process it and so on.

Data Modeling: The information collected from business modeling is refined into a set of data objects (entities) that are needed to support the business. The attributes (character of each entity) are identified and the relation between these data objects (entities) is defined.

Process Modeling: The data object defined in the data modeling phase are transformed to achieve the information flow necessary to implement a business function. Processing descriptions are created for adding, modifying, deleting or retrieving a data object.

Application Generation: Automated tools are used to facilitate construction of the software.

Testing and Turn over: Many of the programming components have already been tested since RAD emphasis reuse. This reduces overall testing time. But new components must be tested and all interfaces must be fully exercised.

Advantages

- RAD reduces the development time and re-usability of components help to speed up development. All functions are modularized so it is easy to work with.

Disadvantages

- For Large (but scalable) projects, RAD requires sufficient resources to create the right number of RAD teams.
- Not all applications are compatible for RAD. If a system cannot be properly modularized, building components for RAD will be problematic.
- RAD is not appropriate when technical risks are high. This normally occurs when a new application makes heavy use of new technology or when the new software requires a high degree of interoperability.

Prototyping Model

Prototyping is most beneficial in systems that will have many interactions with the users and also Prototyping is very effective in the analysis and design of on- line / web based systems. There were two major types of prototyping called Throwaway Prototyping / Rapid prototyping and Evolutionary Prototyping. [WWW6]

Throwaway prototyping / Rapid prototyping is the prototyping process which is started with the requirements which are poorly understood and creating a working model of various parts of the system at a very early stage. Evolutionary prototyping is the prototyping process, the development is started with the requirements which are best understood. The system evolves by adding new features as they are proposed by the client.

There were several advantages and disadvantages with Throwaway prototyping / Rapid prototyping model.

Advantages

- Users are actively involved in the development.
- It provides a better system to users, as users have natural tendency to change their mind in specifying requirements and this method of developing systems supports this user tendency.
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier as the system is modeled side by side.
- Quicker user feedback is available leading to better solutions.

Disadvantages

- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.

3.4 The Selected Development Process Model of the System

After considering the nature of the system and the advantages / disadvantages of each main software process models, Rational Unified Process (RUP) model was selected as the software process model to carry out the development process.

The Rational Unified Process is a Software Engineering Process. A disciplined approach to assigning tasks and responsibilities within a development organization is provided by Rational Unified Process . Its goal is to ensure the production of high-quality software that meets the needs of its end-users, within a predictable schedule and budget.

The best practices of the Rational Unified Process are listed bellow.

- Develop software iteratively.

- Manage requirements.
- Use component-based architectures.
- Visually model software.
- Verify software quality.
- Control changes to software.

3.4.1 Phases of Rational Unified Process

The Rational Unified Process divides one development cycle in four consecutive phases. Each phase is concluded with a well-defined milestone. Phases of the process are listed as follows.[WWW7]

- Inception phase
- Elaboration phase
- Construction phase
- Transition phase

Inception Phase

During the inception phase, establishing the business case for the system and delimiting the project scope are carried out. Following tasks are done in order to accomplish this.

- Identify all external entities with which the system will interact (actors) and define the nature of this interaction at a high-level.
- Identifying all use cases and describing a few significant ones.
- Estimating the resources needed.
- Risk assessment.

The outcome of the inception phase

- An initial use-case model (10%-20% complete).
- An initial risk assessment.

- A vision document.
- An initial project glossary.
- A project plan, showing phases and iterations.

Elaboration Phase

Following major activities are carried out during the elaboration phase.

- Analyzing the problem domain.
- Establishing a sound architectural foundation.
- Developing the project plan, and eliminate the highest risk elements of the project.

The outcome of the inception phase

- A use-case model (at least 80% complete) - all use cases and actors have been identified, and most use-case descriptions have been developed.
- Supplementary requirements capturing the non functional requirements and any requirements that are not associated with a specific use case.
- A Software Architecture Description.
- An executable architectural prototype.
- A revised risk list and a revised business case.
- A development plan for the overall project, including the coarse-grained project plan, showing iterations and evaluation criteria for each iteration.

Construction Phase

During the construction phase, all remaining components and application features are developed and integrated into the product, and all features are thoroughly tested.

The outcome of the construction phase is a product ready to put in hands of its users. At minimum, it consists of,

- The software product integrated on the adequate platforms.
- The user manuals.
- A description of the current release.

Transition Phase

The purpose of the transition phase is to transition the software product to the user community. Once the product has been given to the end user, issues usually arise that require to develop new releases, correct some problems, or finish the features that were postponed.

The transition phase focuses on the activities required to place the software into the hands of the users. Typically, this phase includes several iterations, including beta releases, general availability releases, as well as bug-fix and enhancement releases. Considerable effort is expended in developing user-oriented documentation, training users, supporting users in their initial product use, and reacting to user feedback.

Rational Unified Process is shown by the following figure [Figure3.1].

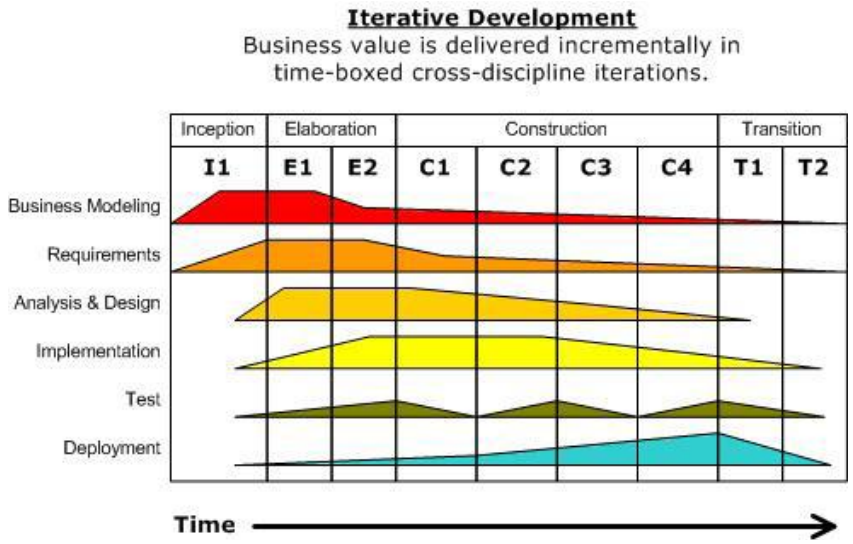

Figure 3.1 Rational Unified Process (RUP)

3.4.2 Advantages of Rational Unified Process

The selected process model has several advantages with comparison to other models. For example,

- The most important innovation of Rational Unified Process (RUP) is the separation of workflows and phases.
- Phases are dynamic and have goals.
- The work flows are static.
- Technical activities are not associated with a single phase but may be used throughout the development to achieve the goals.
- When considering the spiral model due to the relative newness in spiral model, it is difficult to assess its strengths and weaknesses.

3.5 Model View Controller (MVC) Architecture

Model View Controller (MVC) is a software architecture, currently considered an architectural pattern used in software engineering. The pattern isolates the application logic for the user from input and presentation (GUI), permitting independent development, testing and maintenance of each. [WWW8]

The model is used to manage information and notify observers when that information changes. The model is the domain-specific representation of the data upon which the application operates. Domain logic adds meaning to raw data (for example, registering a patient, tracing default patients). When a model changes its state, it notifies its associated views so they can be refreshed.

The view renders the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes. A view port typically has a one to one correspondence with a display surface and knows how to render to it.

The controller receives input and initiates a response by making calls on model objects. A controller accepts input from the user and instructs the model and view port to perform actions based on that input.

Following [Figure 3.2] shows the flow of Model View Controller (MVC) architecture.

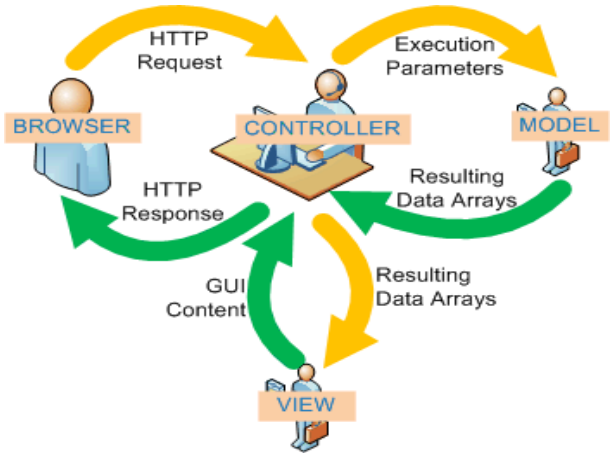

Figure 3.2 MVC Architecture

3.6 System Design

System design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. [WWW9]

Good design is the key to effective engineering, allows efficient code to be produced.
The proposed system is designed by using the object oriented design concepts and modeled using various diagrams.

Object-Oriented Design (OOD) is a software engineering approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its class, its state (data elements), and its behavior. Various models can be created to show the static structure, dynamic behavior, and runtime deployment of these collaborating objects.

Unified Modeling Language (UML) has been used for representing the notations of various models created in object oriented design approach. For example, class diagram, activity diagrams, sequence diagrams.

3.6.1 Identify Classes

The Rational Unified Process (RUP) [Terry Quatrani, 2002] advocates finding the class for a system under development by looking for boundary, control and entity classes. These three stereotypes conform to a “model-view-controller” point of view and allow analyst to partition the system by separating the view from the domain from the control needed by the system.

Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes.

3.6.2 Class Diagram

The class diagram [Figure 3.3] is used to design the static behavior of the system. Set of classes were filtered using identified set of candidate classes. Then the attributes and the main methods were also identified. Diagram was drawn to provide a picture or view of all of the classes in the system.

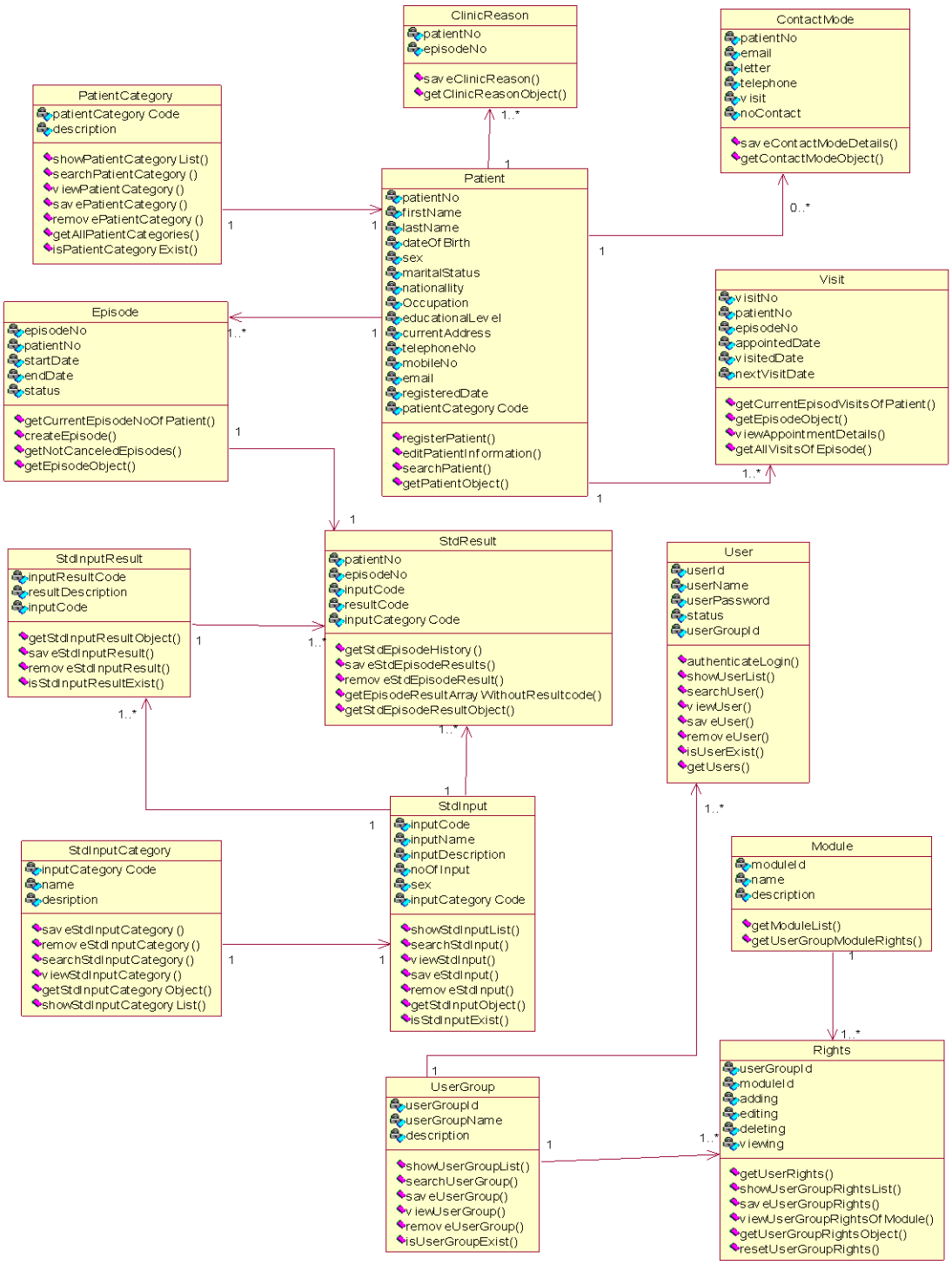

Figure 3.3 Class diagram

3.6.3 Activity Diagrams

Activity diagrams represent the business and operational work flows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state. Also activity diagrams can be used to show activities that are conditional or parallel. Main activity diagrams are illustrated below.

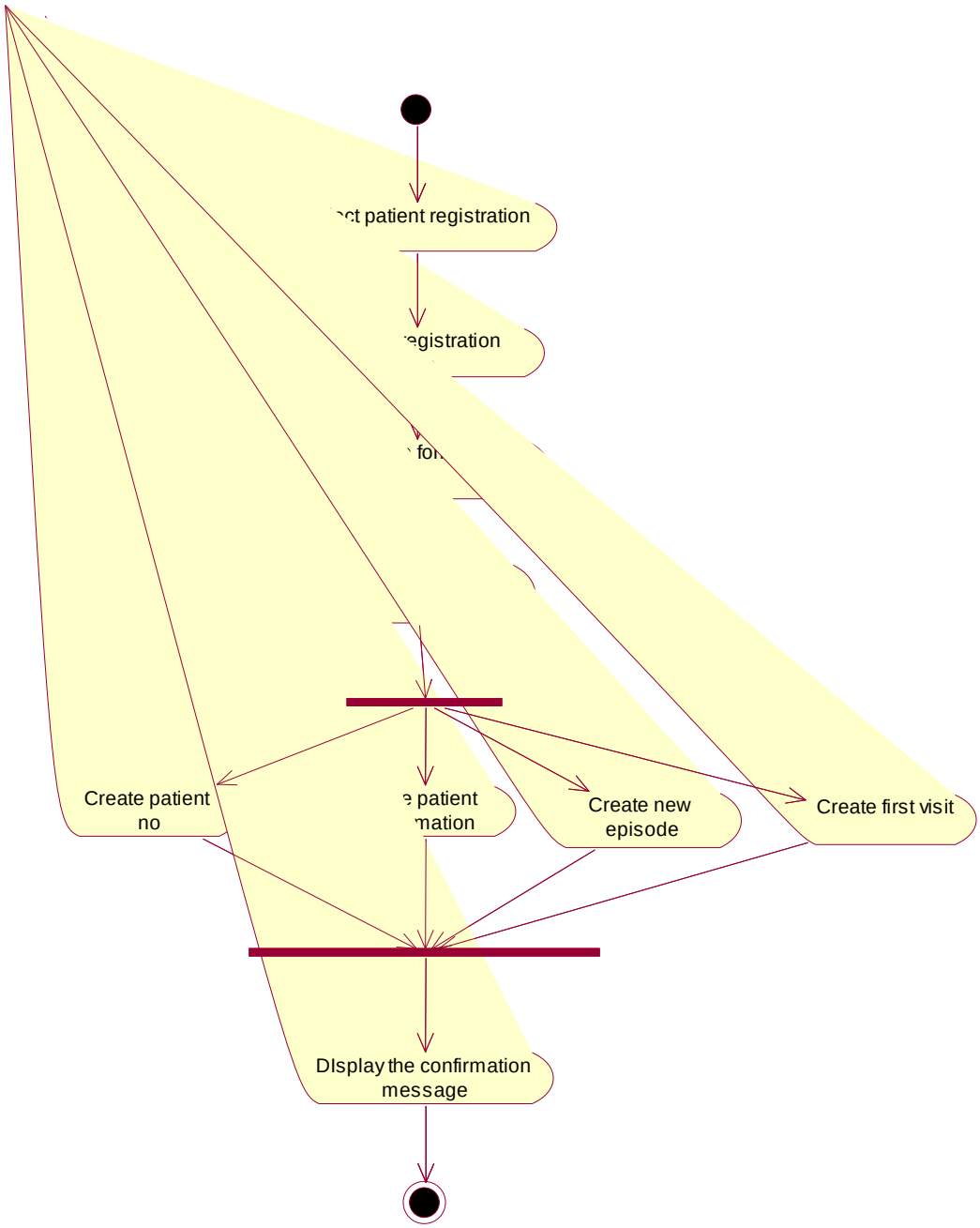

Figure 3.4 Activity diagram for the use case Register Patient [Table 2.2]

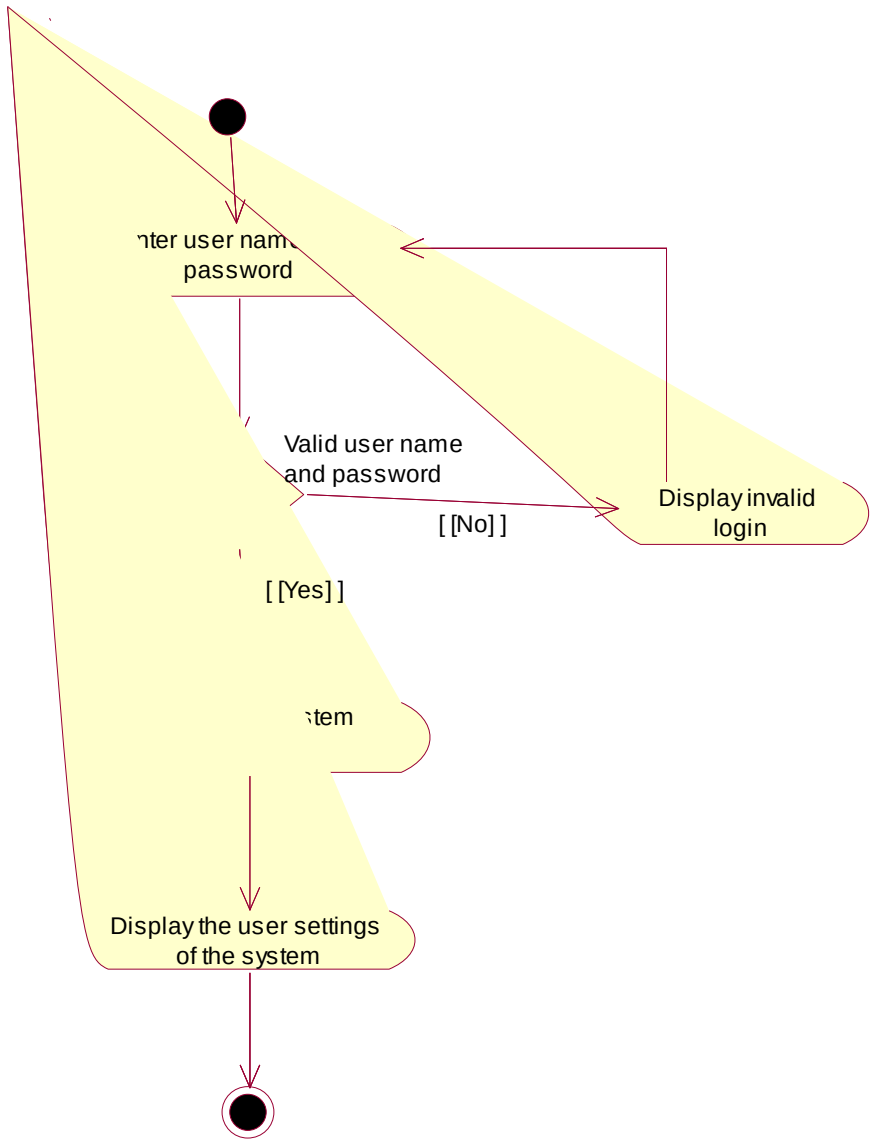

Figure 3.5 Activity diagram for the use case System Login [Table 2.1]

3.6.4 Sequence Diagrams

Sequence diagrams show object interactions arranged in time sequence. It models the objects and the messages exchanged to carry out a scenario. The sequence diagrams of the main use cases of the system are shown bellow.

The objects interaction of the use case “Register Patient” [Table 2.2]] is described by the following sequence diagram [Figure 3.6].

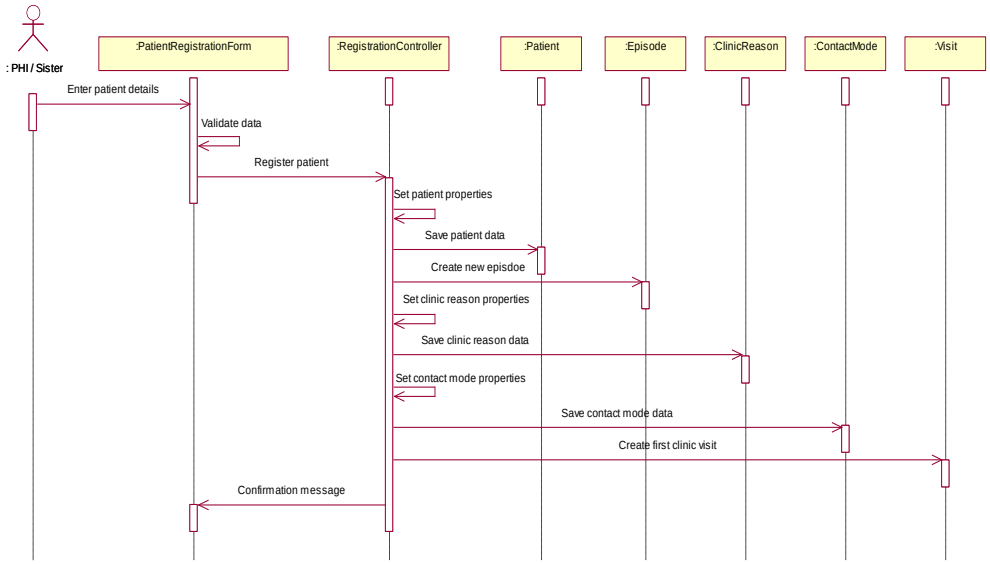

Figure 3.6 Sequence diagram for the use case Register Patient

The objects interaction of the use case “View Appointment List” [Table 2.4] is described by the following sequence diagram[Figure 3.7].

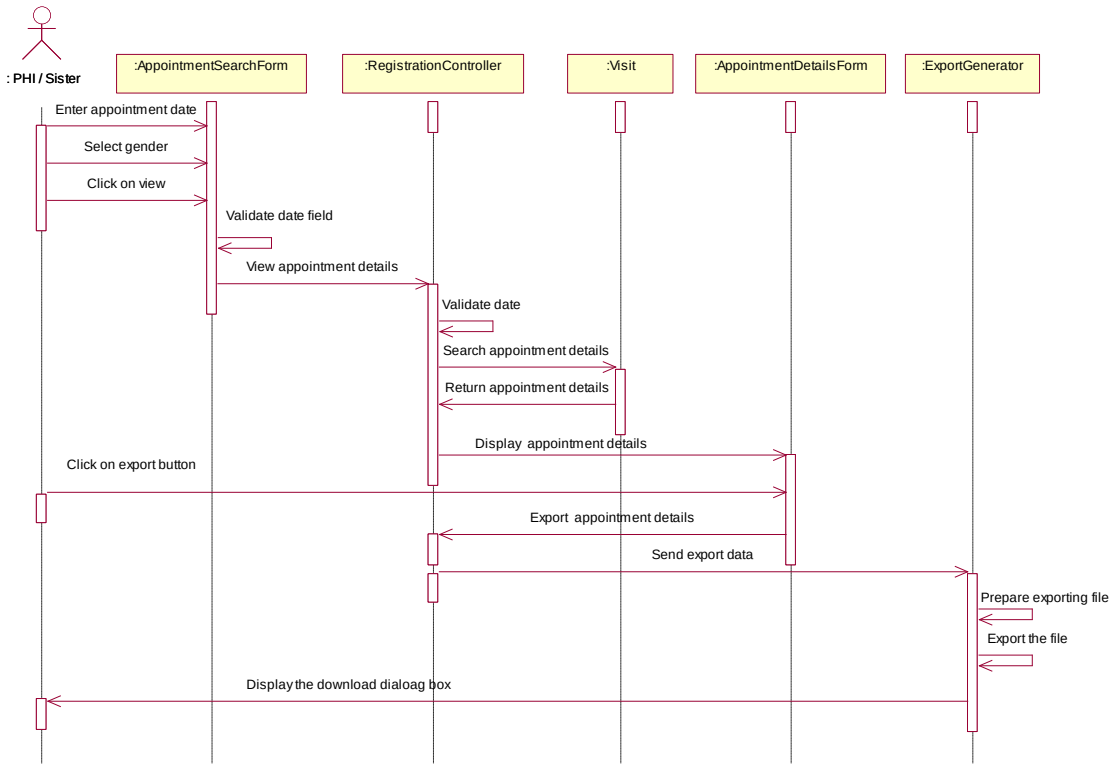

Figure 3.7 Sequence diagram for use case View Appointment List

The objects interaction of the use case “Login” [Table 2.1]] is described by the following sequence diagram [Figure 3.8].

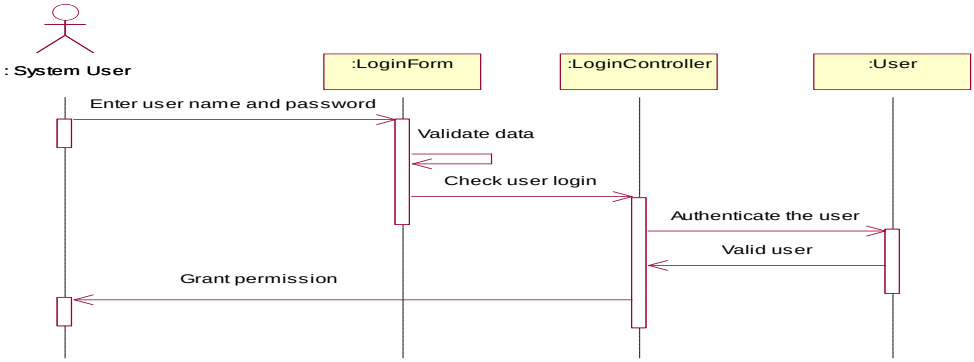

Figure 3.8 Sequence diagram for the use case system login

3.6.5 State Chart Diagrams

State diagrams are used to describe the behavior of a system. State diagrams describe all of the possible states of an object as events occur. Each diagram usually represents objects of a single class and tracks the different states of its objects through the system.

3.7 Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

One of the key factors of computerization of a manual system is to group information in to a single place and there by effectively manipulate it and maintain it. Any software application in order to be successfully developed and put into operation the data storage section has to be accurate, complete and robust.

3.7.1 The Entity Relationship Model

An entity relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs.[WWW10]

Following ER diagram [Figure 3.9] was drawn after identifying the entities, attributes and their relationships of the system.

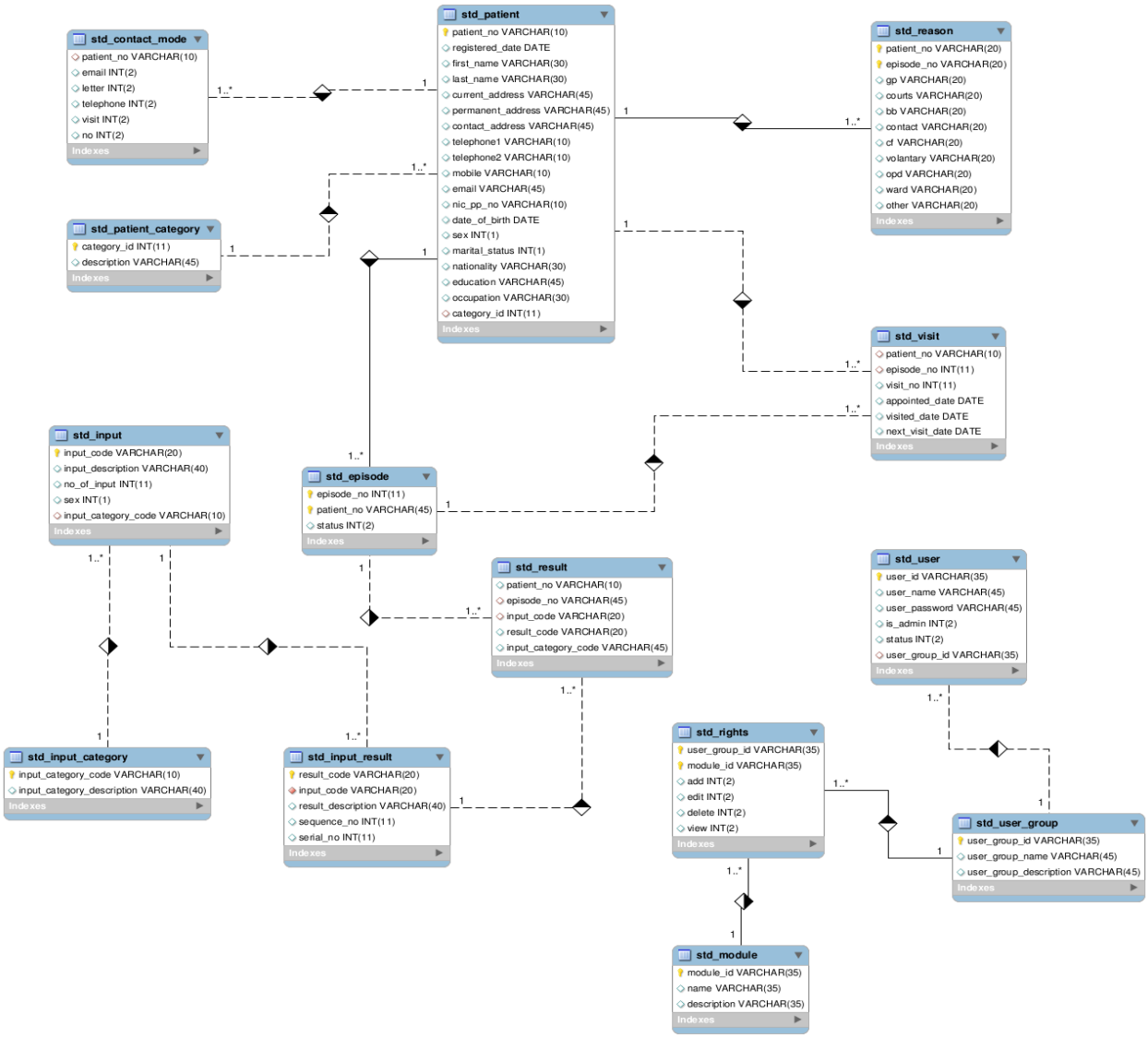

Figure 3.9 ER diagram

All the remaining activity diagrams and sequence diagrams are listed in the Appendix B.

3.8 User Interface Design

Major user interface design considerations of the system are explained bellow.

Menu Based Navigation

One main template has been used with a dropdown menu bar for the entire application. All the web pages of the system are loaded inside this template. Main menu bar of the system is shown by the following figure [Figure 3.10].

All the main modules of the system are grouped into different main menus and the functionalities of each module are listed as sub menus of the main menu. For example, the sub menu items such as “Patient Registration”, “View Appointment List”, “Mark /Create Visit” are listed under the main menu called “Registration” .

Access rights of system modules have been given through the menus.

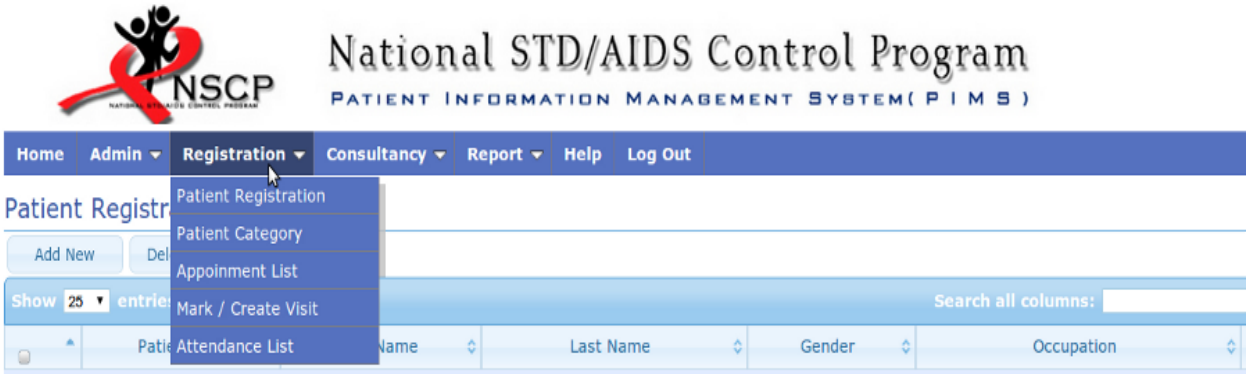

Figure 3.10 Main menu bar.

Tab Based Patient Registration Form

Patient registration form is the main web form of the application. According to the client’s requirements, that form should be divided into different sections such as “Personal Information” , “Contact Information”, “Reason for Attendance Information” etc. After considering different ways for designing the registration form , tab based method was used. All the form fields are included in relevant tab.

Main registration from of the system is shown by the following figure [Figure 3.11].

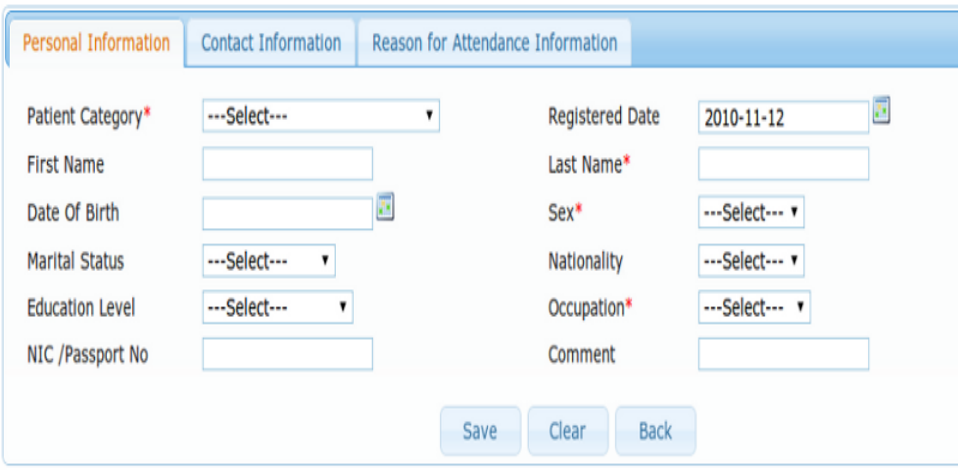

Figure 3.11 Patient registration form.

List Views with Deleting, Sorting and Searching Features

List view [Figure 3.12] with sorting and searching features has been used for displaying various details. For example, patient's information , user groups, users, patient categories etc.

There are columns included in the list and those columns can be used to sort the records. Also any record in the list can be searched by using the column name as the searching filed.

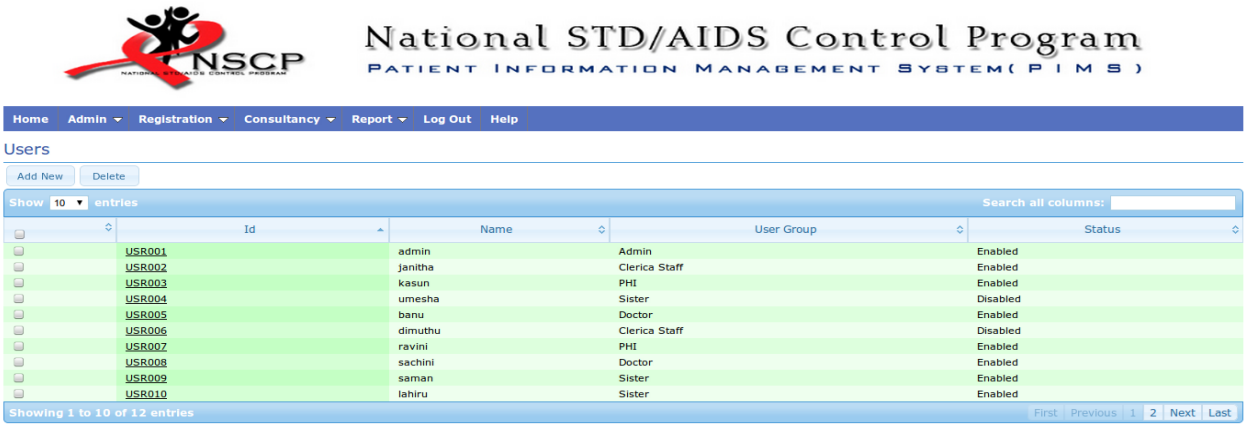

Figure 3.12 List view.

Accordion Menu for Dynamically Generated STD Episode Care From

STD episode care form is generated dynamically by using the input questions and the results in the database. This form should be divided into different sections and the questions and answers should be included in relevant sections.

The form will be expanded vertically on the screen if it got many input questions. So that will cause users to scroll the webpage . To avoid this usability issue, accordion menu [Figure 3.13] feature has been used to design the STD episode care from. By using the accordion menu, users are given to navigate the necessary section very easily**.**

Figure 3.13 Accordion menu.

All the forms and forms fields should be consistent.

Same colors, fonts, heading styles should be followed by all the forms and forms fields.

3.9 Main User Interfaces of the System

The main user interfaces are shown in the following figures that affect initial system functionalities. Remaining user interfaces are listed in the Appendix B.

3.9.1 Login Module

System login page of the users and the main menus of the system are shown bellow.

Figure 3.14 Login page of the system

Figure 3.15 Invalid login by user.

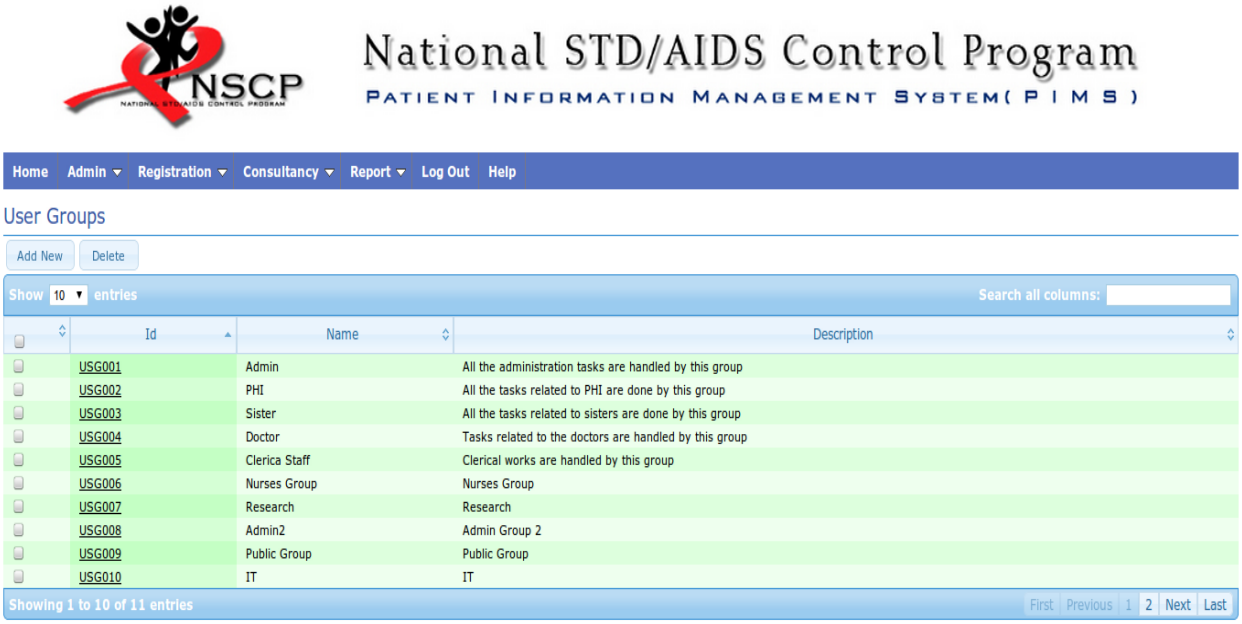

Figure 3.16 Main menus of the system.

List View

In one view, searching , adding , deleting and editing data options are provided by the list view. This will help to increase the usability of the system.

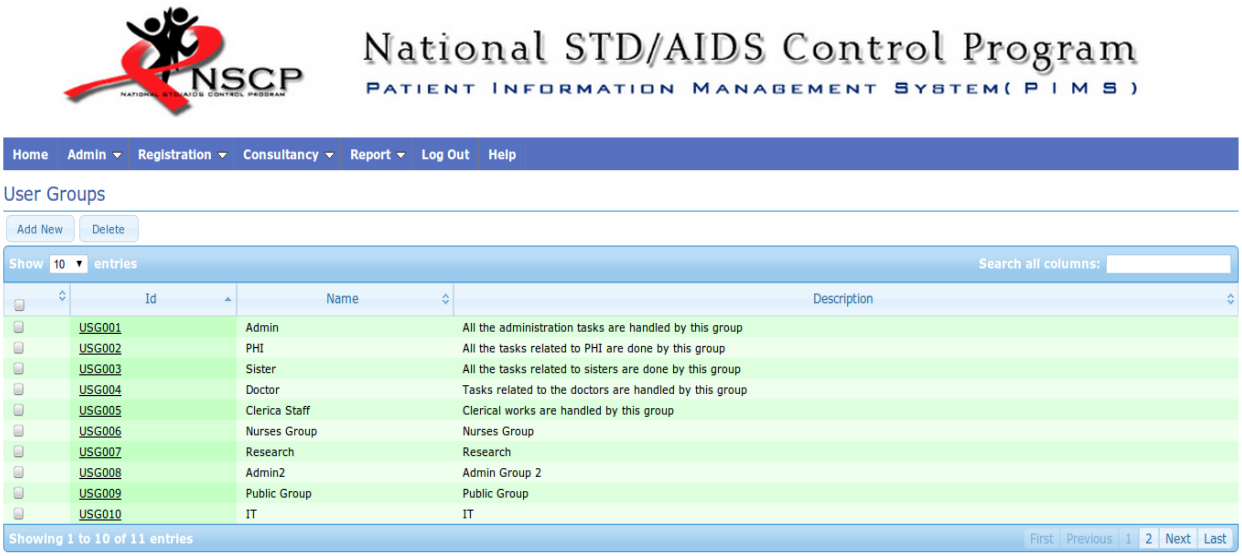

Figure 3.17 List view for searching, adding, editing and deleting data.

3.9.2 Registration Module

Main areas of the registration module are shown by the following user interfaces.

Patient Registration

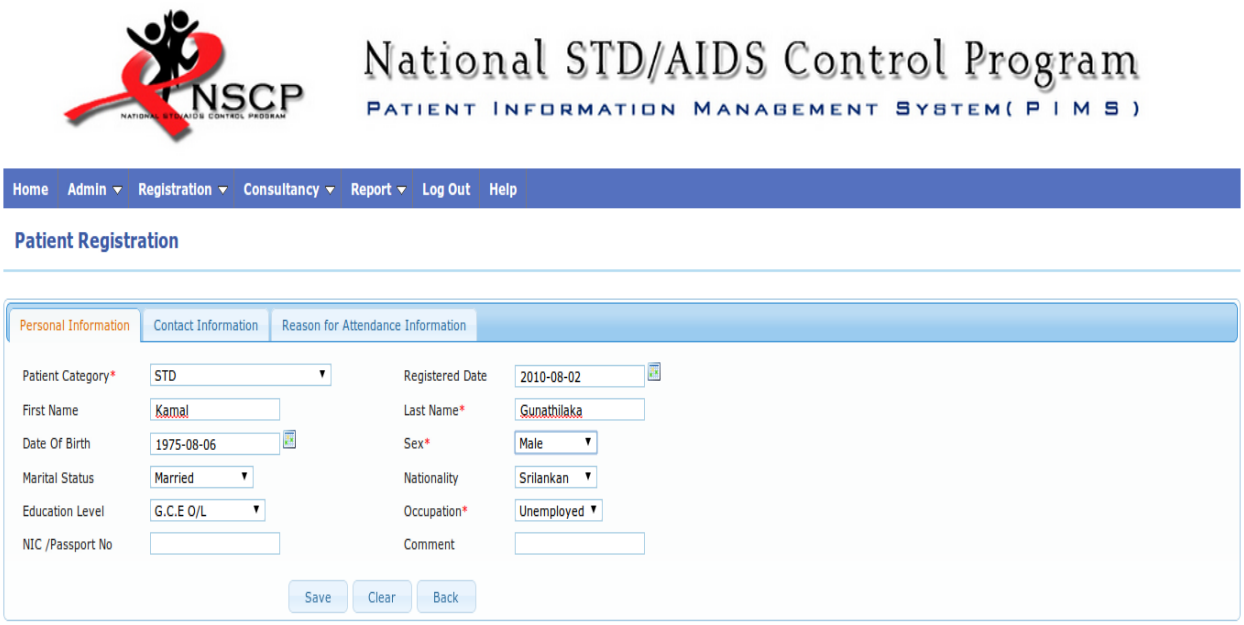

Figure 3.18 Main registration form.

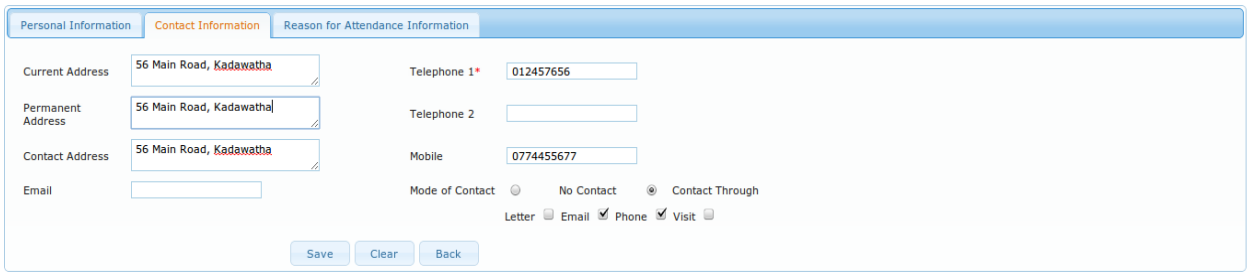

Figure 3.19 Contact details tab in main registration form

View Appointment List

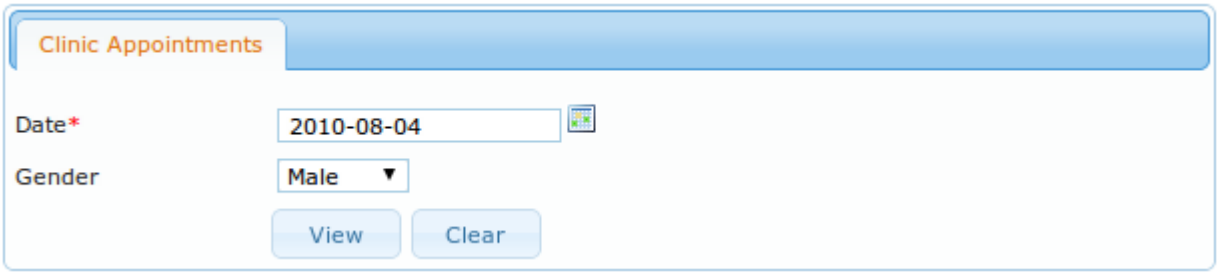

Figure 3.20 Search appointment details of patients for a particular day.

Figure 3.21 Appointment list of patients.

3.9.3 Consultancy Module

Following user interfaces are used to give an overview of the consultancy module.

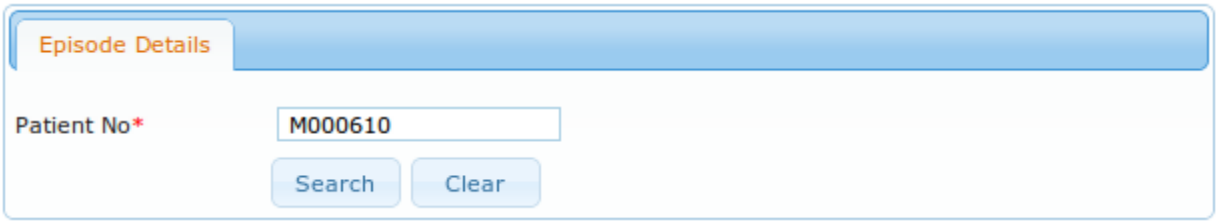

Figure 3.22 Search episode details of a patient.

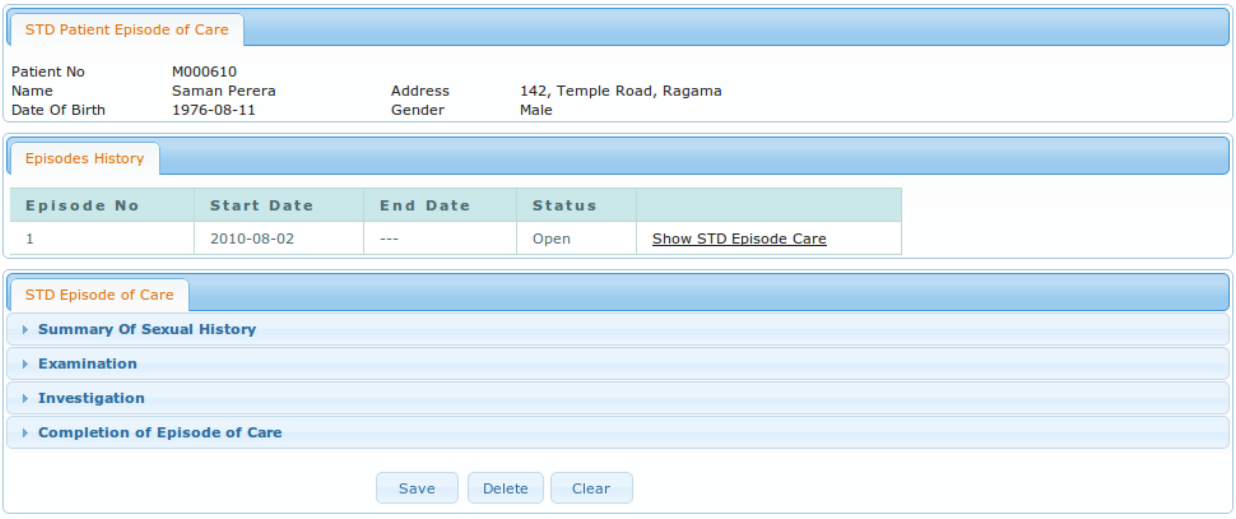

Figure 3.23 Existing episodes of the patient.

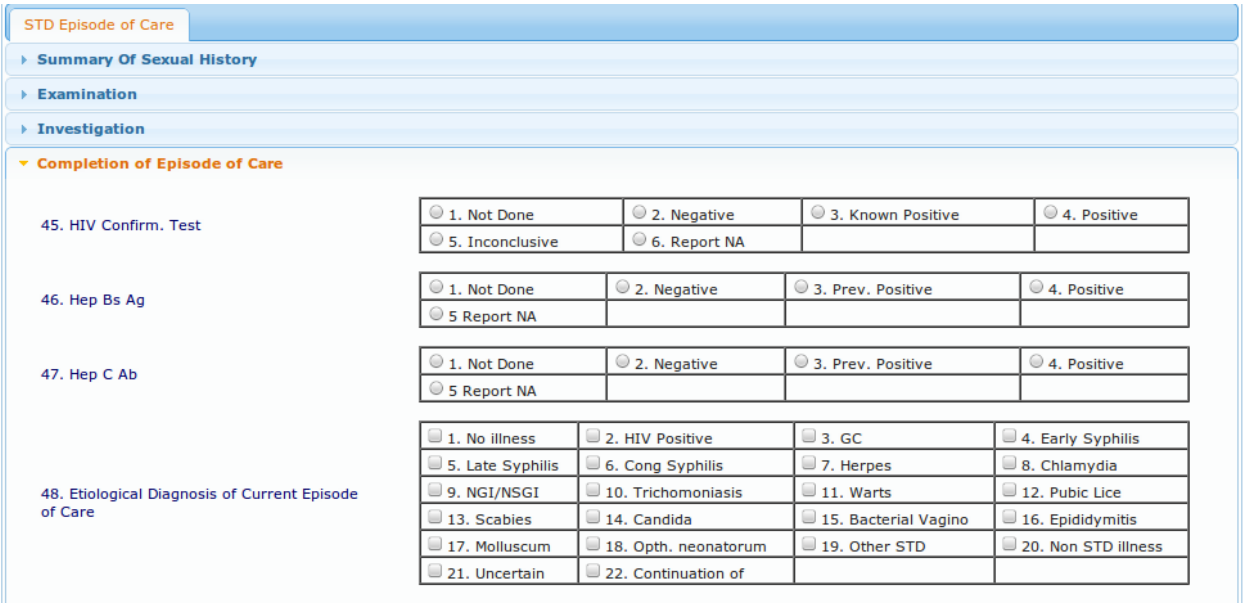

Figure 3.24 STD episode care details.

Chapter 4 - Implementation

4. Implementation

This chapter mainly describes the implementation of the system. This includes the major code segments and the module structure of the system. The implementation environment (hardware and software), any existing code that was reused, development tools used and the database management system are also addressed in this chapter.

4.1 Implementation Environment

Hardware Configuration

Figure 4.1 Hardware configuration of developed machine

Processor Clock speed	Intel(R) Core(TM) Duo Cpu 2.8GHz
Memory	2GB
Hard Disk Capacity	220GB
VGA memory	1GB
Monitor	Minimum Resolution 800*600

Software Configuration

Table 4.2 Software configuration of developed machine

Operating System	Linux (Ubuntu 10.04)
Programming Languages	PHP 5 with the Symfony frame work
Web Server	Apache 2.2
Database Server	MySQL 5.1
Web Browsers	Mozila FireFox 3.6.7
Open Source Libraries	Jquery , Jquery- UI libraries , TCPDF
Development Tools	Netbeans 6.9, Eclipse, Dreamweaver 8, Rational Rose
Other	XAMPP

4.2 Platform Dependencies

The proposed system has been implemented using platform independent development technologies that use free and open source platform independence software such as Apache, MySQL, PHP , Jquery and the Symfony frame work. It will provide high portability to the system. Therefore system can be accessed from anywhere, any platform.

4.3 Development Technologies

PHP

PHP stands for Hypertext Preprocessor. PHP is a widely used scripting language that is especially suited for web development and can be embedded into HTML. It is also an open source, platform independent software.

HTML

HTML stands for Hypertext Markup Language, HTML is used to markup the web pages. Various tags are used to markup different objects in a web page. For example, heading , fonts, images etc.

Apache

Apache is an open source web server. It has been used as a web server to execute the PHP pages. This is one of the widely used web servers in the world.

MySQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is a free open source, platform independent database software often used on the web.

Symfony Framework

Symfony is a web application framework written in PHP which follows the model-view-controller (MVC) pattern. It provides an architecture, components and tools to build complex web applications faster.

JQuery / JQuery-UI

jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. Jquery-UI is an extension library of JQuery which is used for creating user interfaces in web applications.

4.4 Operating system

The operating system selection was very straightforward. Since all the used technologies are platform independent we can use any operating system for web server and client machines. For example we can use Microsoft® Windows NT/ 2000 / XP / Vista, 7, UNIX and Linux flavor (Cent OS, Redhat®, Macintosh, Ubuntu) for client machines.

4.5 Database Management System

MySQL server was chosen as the database management system for the system.

4.6 Symfony and Doctrine

Symfony is a web application framework for PHP projects. It aims to speed up the creation and maintenance of web applications, and to replace the repetitive coding tasks by power, control and pleasure. Using Symfony is so natural and easy for people used to PHP and the design patterns of Internet. The clean design and code readability will keep your delays short. Developers can apply agile development and focus on applicative logic without losing time to write endless XML configuration files. [WWW11]

Doctrine is a PHP ORM (object relational mapper) for PHP 5.2.3+ that sits on top of a powerful PHP DBAL (database abstraction layer). One of its key features is the ability to optionally write database queries in an OO (object oriented) SQL-dialect called DQL inspired by Hibernate's HQL. This provides developers with a powerful alternative to SQL that maintains a maximum of flexibility without requiring needless code duplication .[WWW12]

4.7 Code and Module Structure

The system has been divided into five modules such as Login, Admin, Registration, Consultancy and Report. In code level, modularization of the system has been achieved through the Symfony framework. Symfony has provided a proper structure for modularizing and developing web applications. Mainly MVC architecture pattern is followed by Symfony for the implementation.

The MVC design pattern defines a way to organize the code according to its nature. This pattern separates the code into three layers.

- The Model layer defines the business logic (the database belongs to this layer). Symfony framework stores all the classes and files related to the Model in the lib/model/ directory.
- The View is what the user interacts with. In Symfony, the View layer is mainly made of PHP templates. They are stored in various templates/ directories
- The Controller is a piece of code that calls the Model to get some data that it passes to the View for rendering to the client.

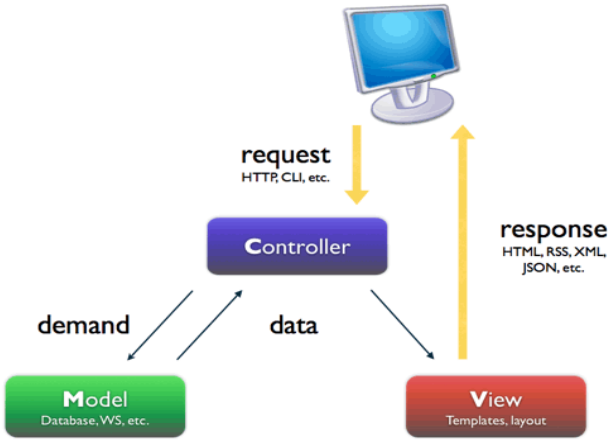

Figure 4.1 Initial system structure

All the requests of a module are handled by a separate controller and business logics of the module are processed by different model classes. For example,

Registration module of PIMS, all the requests such as patient registration, searching records of patients, creating categories of patients etc are handled by a controller called r**egistrationActions** and different model classes such as **Patient**, **PatientCategory** etc associated with the controller.

Code segments of some of the main functions are listed below. . Remaining code segments are listed in Appendix F.

Code 4.1 All the requests for Registration module are handled by **registrationActions** controller class

```
<?php
class registrationActions extends sfActions {
//Save patient information
public function executeSavePatientInformation(sfWebRequest $request) {
    if ($request->isMethod('post')) {
      //Create the patient no
      $patientNoTracker = new PatientNoTracker();
      $registeredYear = date("Y");
      $sex = $request->getParameter('cmbSex');
      $patientNo = $patientNoTracker->getNextPatientNo($registeredYear, $sex);
       //Save patient object
      $patient = new Patient();
      $patient->setPatientNo($patientNo);
      $this->_savePatientInformation($patient, $request);
      //Create a new episdoe for the patient
      $episode = new Episode();
      $episodeNo = $episode->getCurrentEpisodeNoOfPatient($patientNo);
      $episode->setPatientNo($patientNo);
      $episode->setEpisodeNo($episodeNo);
      $episode->setStatus(Episode::EPISODE_STATUS_OPEN);
      $episode->setComment("New Patient");
      $episode->setStartDate(date('Y-m-d'));
      $episode->setEndDate(NULL);
      $episode->save();
      //Save contact mode details
      $contactMode = new ContactMode();
      $contactMode->setPatientNo($patientNo);
      $this->_saveContactModeInformation($contactMode, $request);
      //Save linic reasons
      $clinicReason = new ClinicReason();
      $clinicReason->setPatientNo($patientNo);
      $clinicReason->setEpisodeNo($episodeNo);
      $this->_saveClinicReasonInformation($clinicReason, $request);
      //Add the first visiting record
      $visit = new Visit();
      $visit->setPatientNo($patientNo);
      $visit->setEpisodeNo($episodeNo);
      $visit->setVisitNo(1);
      $visit->setAppointedDate(date('Y-m-d'));
      $visit->setVisitedDate(date('Y-m-d'));
      $visit->setNextVisitDate(NULL);
      $visit->save();
    }
       $this->getUser()->setFlash('notice', 'Record saved successfully');
                $this-
      >redirect('registration/showPatientList?mode=search&searchMode=patient_no&searchVa
      lue='.    $patientNo);    } }
```

Code 4.2 All the functions related to the user groups are processed by **UserGroup** model class

```
<?php
 //All the functions related to the user groups are handled by this class
class UserGroup extends BaseUserGroup {
     /**
    * Show user group list
    * @param $orderField, $orderBy
    * @return objects of $userGroups
    */
   public function showUserGroupList($orderField = 'user_group_id', $orderBy = 'ASC') {
      try {
         $q = Doctrine_Query::create()
               ->from('UserGroup');
         $userGroups = $q->execute();
         return $userGroups;
      } catch (Exception $e) {
         throw new Exception($e->getMessage());
      }
   }
   /**
    * Search an user group object
    * @param $searchMode, $searchValue
    * @return object of $userGroup
    */
   public function searchUserGroup($searchMode, $searchValue) {
      try {
         $q = Doctrine_Query::create()
               ->from('UserGroup')
               ->where("$searchMode = ?", $searchValue);
          $userGroup = $q->execute();
         return $userGroup;
      } catch (Exception $e) {
         throw new Exception($e->getMessage());
```

```
    }   }}
```

Code 4.3 . Patients appointments and visited data processed by **Visit** model class

```
class Visit extends BaseVisit {
  //Get current episodes visits
   public function getCurrentEpisodeVisitsOfPatient() {
     try {
         $q = Doctrine_Query::create()
            ->from('Episode e')
            ->leftJoin('e.Visit v ON e.patient_no = v.patient_no')
            ->where('e.patient_no = ?', $this->getPatientNo())
            ->andWhere('e.episode_no  = ?', $this->getEpisodeNo());

            $visits = $q->fetchArray();
            return $visits;
     }catch (Exception $e) {
         throw new Exception($e->getMessage());
      }
   }
//View appointment details of patients
public function viewAppointmentDeatils($date, $gender) {
      try {
         $q = Doctrine_Query::create()
               ->from('Patient e')
               ->leftJoin('e.Visit v ON e.patient_no = v.patient_no')
               ->where('v.appointed_date = ?', $date)
               ->andWhere('e.sex   = ?',$gender)
               ->andWhere('v.visited_date = ?','0000-00-00')
               ->andWhere('v.next_visit_date = ?','0000-00-00');
               $appointments = $q->fetchArray();
               return $appointments;
      }catch (Exception $e) {
         throw new Exception($e->getMessage());
      }
```

```
    }
}
```

Code 4.4 Input validation of patient registration form**.**

```
<script type="text/javascript">
//Validate the patient category drop down list
    jQuery.validator.addMethod(   "selectNone",   function(value, element) {
        if (element.value == "0")  {
            return false;
        }  else {
             return true;
      }
    },
  "Please select the patient category."    );
  $(document).ready(function() {
        //Validate the form
        $("#frmPatient").validate({
            rules: {
                cmbPatientCategory: {
                 selectNone: true
                },
                cmbSex: {
                    selectSex: true
                },
                cmbOccupation: {
                    selectJob: true
                },
                txtLastName: {
                    required: true
                },
                txtTel1: {
                    required: true
                }
            },
```

```
        messages: {
            txtLastName: "Last name is required.",
            txtTel1: "Telephone number is required."
        }   });   });
</script>
```

Form validation was done by reusing the JQuery validate method[WWW13] and some user defined custom functions were also created to validate some form fields.

4.8 Third Party Libraries and Source Codes

Few third party libraries and codes were used in order to implement this system in more interactive way.

Table 4.3 Third party libraries used in the project

Description	Web	Author
Symfony framework	http://www.symfony-project.org/	
JQuery library	http://jquery.com/	
JQuery UI	http://jqueryui.com/	
TCPDF library for creating PDF report	http://www.tecnick.com/	
JavaScript library for creating charts	http://www.filamentgroup.com/	

Chapter 5 - Evaluation

5. Evaluation

In any software development life cycle, testing plays a major roll. That is to ensure the product accomplishes the required quality and completeness according to the user requirements. One major objective of software testing is to identify the defects using software testing techniques.

Following are the major objectives of testing the developed system using testing techniques.

- Verifying all acceptance test cases were implemented as defined in the specification. This will ensure the product is implemented according to the specification.
- UI testing has been performed to identify the UI issues, consistency of the UI and the usability of the product. This is to ensure that the software delivers user-friendly and appealing environment to the client.
- User Acceptance Testing has been conducted to ensure that the product possess the client's expectations.
- Security testing has been conducted to ensure that the product is not vulnerable to common injection attack.
- Performance testing has been conducted to ensure that the product responses in a timely manner even with high volume of data.

5.1Test Plan, Testing Techniques and Strategies

Based on clients expectations and industry standards the test plan and the strategies were decided. Scope of the test plan covers the functional and non - functional requirements of the system. Following mind map was used to elaborate the test plan scope.

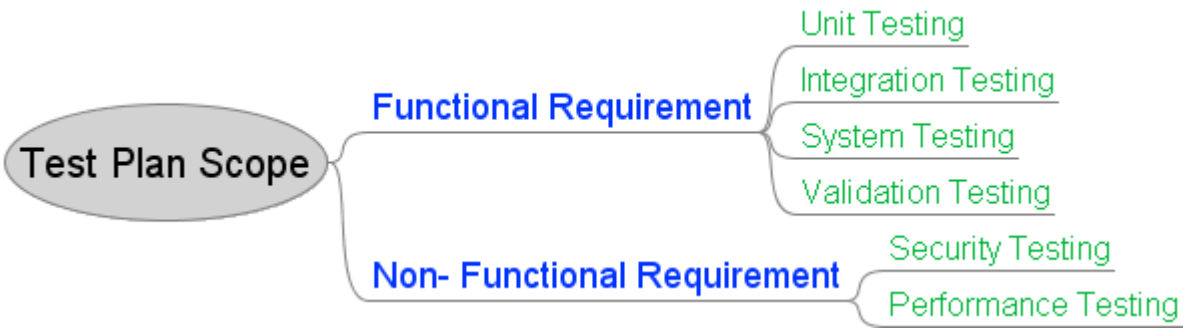

Figure 5.1 Test plan scope

Above scope is covered using widely used testing techniques.

Black Box Testing

This technique is mainly used to test the product against the specifications. This involves giving inputs to the system and monitoring the output for their behavior.

White Box Testing

White box testing mainly used to identify the internal logic's of the system are properly developed.

Unit Testing

Unit testing is used to test units as a whole, this can be used to test the interactions between many functions but the scope still remains in the unit. White box testing technique is used to create the test cases.

Integration Testing

A group of classes which act in combination to provide a set of services should be tested together. This is termed as cluster testing. Clusters have to be created using the knowledge of their operations and the features of the system that are implemented by these clusters.

System Testing

The software is tested as a whole to monitor and validate the system functionalities. The system is tested for the recovery ability in a failure, security of the system to monitor unauthorized access, and performance test to monitor run time performance.

Validation Testing

All the inputs of data are validated before saving to the database to check the data is in correct format.

Security Testing

Security is one major concern on web applications. The system has been tested against SQL Injection attacks.

SQL injections: A malicious sql script which is used to extract unauthorized information from the application or bring down the application by destroying the internal database structure or data. These scripts can be used in the form of the web application as input data.

Black box and White box testing techniques has been used to identify the SQL injection vulnerabilities.

Performance Testing

Performance testing has been used to identify whether the system perform according to the expected efficiency.

5.2 Acceptance Testing

After finishing the integration testing, acceptance testing was conducted. Acceptance testing also known as the user testing and it was the final stage of testing and was carried out with the client. System was installed at the client's place and a full demonstration was given to the system users. Then, system was tested according to the acceptance test cases.

Feedback from the client was considered and certain things were changed according to their need.

5.3 Selecting Test Data

The documents of the manual system were used to select the test data. Main documents of the manual system such as registration form of patients, clinical card, attendance files, STD episode care from, various reports etc were mainly referred for selecting the required test data.

5.4 Selecting Test Cases

Test cases were developed to do the Unit and Integration testing. These test cases have been tested several times with different test data. Following are some of the generated test cases and the expected outputs. Remaining test cases are listed in Appendix E.

Table 5.1 Test Cases for Login Module

Test	Test Description	Expected Result	Status
1	Enter correct user name and password.	Successful login to the system and show the assigned menu options.	Succeeded
2	Enter correct user name and wrong password.	Error message.	Succeeded
3	Enter wrong user name and correct password.	Error message.	Succeeded
4	Only enter the user name .	Error message.	Succeeded
5	Only enter the password.	Error message.	Succeeded
6	Both user name and password are blank.	Error message.	Succeeded

Table 5.2 Test Cases for Admin Module

Test	Test Description	Expected Result	Status
1	Create user group	New user group should be added to the PIMS	Succeeded
2	Edit user group details	Existing user group details should be edited	Succeeded
3	Delete user group	Existing user group should be deleted if needed.	Succeeded
4	Users should be able to search / view user group details in the list view of user groups	User group details should be displayed with a link for viewing more details	Succeeded
5	View list of user groups	All the user groups should be listed with their group id, name, and description	Succeeded
6	Assign access rights and modules for a user group	System should be able to assign modules and their access rights for user groups	Succeeded
7	Reset access rights of a user group	All the assigned rights should be removed	Succeeded
8	Change access rights of a user group	Existing access rights should be changed	Succeeded
9	Create user account	New user account should be added to the PIMS	Succeeded
10	Edit user account details	Existing user account details should be edited	Succeeded
11	Delete user account	Existing user account should be deleted if needed.	Succeeded
12	Users should be able to search / view user account details in the list view of user account	User account details should be displayed with a link for viewing more details	Succeeded
13	View list of user accounts	All the user account should be listed with their user id, name,	Succeeded

		and status	
14	Administrator should be able to change the password of users.	Password should be changed and email of new credentials should be sent to the user by PIMS	Succeeded
15	Administrator should be able to disable an user account.	Particular user should not allow to login to the system and display the error message	Succeeded

Table 5.3 Test Cases for Consultancy Module

Test	Test Description	Expected Result	Status
1	Create input category	Input category details should be added to the system	Succeeded
2	Edit an existing input category	Existing input category details should be changed.	Succeeded
3	Delete input category	Existing input category should be deleted if needed.	Succeeded
4	Users should be able to search / view input category details in the list view	Input category details should be displayed with a link for viewing more details	Succeeded
5	View list of input categories	All the input categories should be listed with their id, name	Succeeded
6	Create input question	Input question details should be added to the system	Succeeded
7	Edit an existing input question	Existing input question details should be changed.	Succeeded
8	Delete input question	Existing input question should be deleted if needed.	Succeeded
9	Users should be able to search / view input question details in the list view	Input question details should be displayed with a link for viewing more details	Succeeded

10	View list of input questions	All the input questions should be listed with their id, name	Succeeded
11	Assign results for the input question	Multiple results should be assigned to a input question	Succeeded
12	Delete result from an input question	Assigned results should be removed from an assigned question	Succeeded
13	Edit result of an input question	Result of an input question should be edited	Succeeded
14	Generate STD episode care form for entering consultancy details of patient	Episode care form should be created dynamically with the created input questions and their results	Succeeded
15	Generate option buttons for input results which has one answer	Generated results of a question should have option buttons if the question has one answer.	Succeeded
16	Generate check boxes for input results which has one or more answers	Generated results of a question should have check boxes if the question has one or more answers.	Succeeded
17	Update STD episode care from	Consultancy details of the patient should be updated in the STD episode care form	Succeeded
18	View STD episodes of a patient	All the episodes should be displayed with episode no, start date, end date, status and a link for viewing the history of episode .	Succeeded
19	View consultancy details of episode	Consultancy details of a episode should be displayed	Succeeded
20	Closed episode details could not be updated	Consultancy details of a closed episode should not be updated	Succeeded

21	Cancel episode	Episode should be canceled if needed	Succeeded
22	Canceled episode details should not be displayed	All the episode details should be displayed except canceled ones	Succeeded

Table 5.4 Test Cases for Registration Module

Test	Test Description	Expected Result	Status
1	Generate patient no	Patient no should be generated according to the gender of the patient and the year registered	Succeeded
2	Register new patient	Patient details should be validated and save to the database	Succeeded
3	Print new patient details	Patient details should be exported as PDF and print	Succeeded
4	Edit an existing patient details	Existing patient details should be edited	Succeeded
5	Users should be able to search a patient by their name or no	Patient details should be displayed with a link for viewing more details	Succeeded
6	Encrypt patient's first name, last name, telephone numbers, mobile numbers and contact addresses details	All the mentioned fields should be encrypted and save to the database	Succeeded
7	Decrypt patient's first name, last name, telephone numbers, mobile numbers and contact addresses details when displayed on the page.	Decrypted value should be displayed	Succeeded
8	View list of registered patients	All the patients should be listed with their patient no, first name. last name. And year registered.	Succeeded
9	Sort the list of registered patients .	Patient details should be sorted their list columns.	Succeeded
10	Create new patient category.	New patient category should be added to the PIMS	Succeeded
11	Edit an existing patient category.	Existing patient category details should be edited	Succeeded
12	Users should be able to search / view patient category details in the list view of categories.	Patient category details should be displayed with a link for viewing more details	Succeeded
13	Delete patient category.	Existing patient category should	Succeeded

		be deleted if needed.	
14	Create a episode for newly registered patient .	New episode should be created automatically when the patient is registered to the system	Succeeded
15	Create a episode for an existing patient.	Relevant user should create the new episode for an existing patient	Succeeded
16	View appointment list of patients .	Appointment list of patients should be generated according to the given date and gender	Succeeded
17	Export / print the appointment list of patients .	System should be able to export the generated appointment list of patients.	Succeeded
18	Mark patient's visited date of the clinic.	Visited date should be updated to the system when patient came to the clinic.	Succeeded
19	Create next appointment date.	Create a new appointment date for the next clinic visit.	Succeeded
20	Enter past appointment or visited date for a current appointment or visited date	Show display the error message	Succeeded
21	Enter visited date as appointment date	Visited date and appointment date can be same	Succeeded

Table 5.5 Data type validation

Test	Test Description	Expected Result	Status
1	Invalid date values are not allowed to enter for date fields .	Should display an error message.	Succeeded
2	Invalid data for number fields .	Should display an error message.	Succeeded
3	Invalid date for string fields.	Should display an error message.	Succeeded

5.5 Overall Project Evaluation

After completing the testing process, evaluation of the system was conducted to decide the successfulness of the project. The evaluation of the system against the set of functional requirements is a useful way of accessing the amount of progress that has been made during the project. Almost all the functional requirements mentioned in the project proposal were found to meet in the final product. Furthermore questionnaire was given to the client to obtain feedback in relation to the PIMS developed for the organization and feedback was taken to evaluate system by potential users. According to their feedback, the evaluation phase was successfully completed. The given questionnaire is attached in Appendix E

5.6 Error Detection and Error Handling

Mainly the error handling functionalities were required in the user authentication and the data entry parts of the system. Duplicate data should be checked before saving to the database, by the system. Also various runtime exceptions should be properly handled by the system.

5.6.1 User Authentication Errors

Figure 5.2 Blank username and password entered by the user

Figure 5.3 Server side authentications for invalid login.

5.6.2Data Redundancy Errors

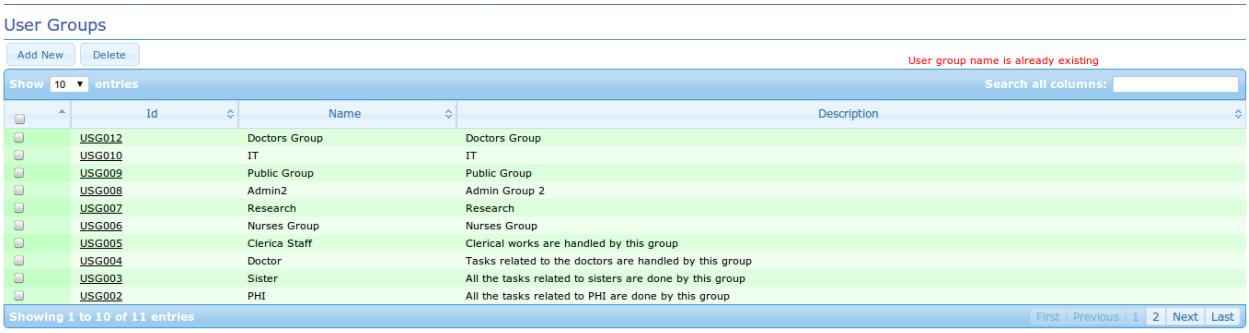

Figure 5.4 Trying to add duplicate user group name.

5.6.3 Data Type and Invalid Data Errors

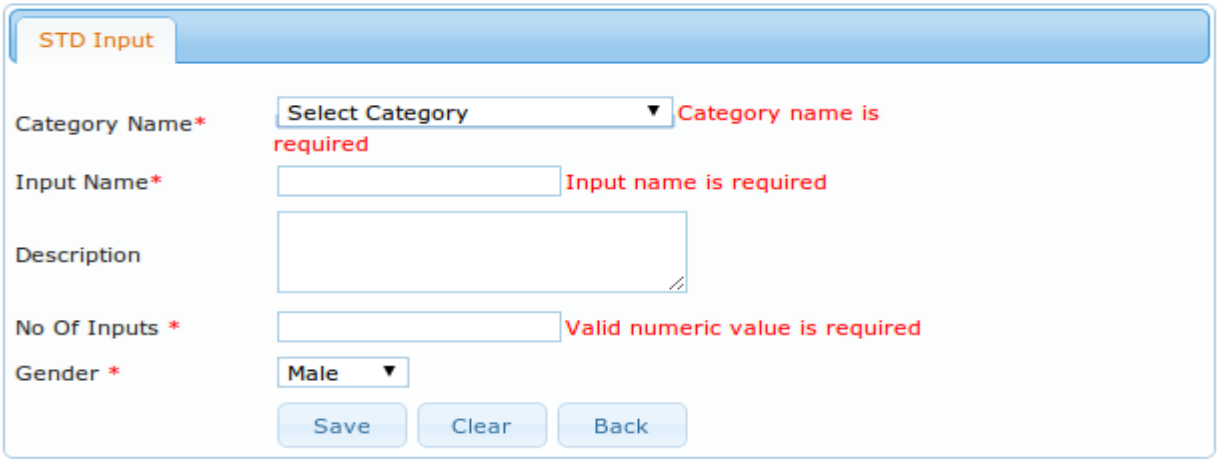

Figure 5.5 Data type and invalid data errors

Chapter 6 - Conclusion

6 Conclusion

This Patient Information Management System was developed for the use of general clinic at the National STD/AIDS Control program in Colombo. Mainly the system will be used by the doctors, PHIs, Sisters and the clerical staff of the clinic . The system was fully implemented, covered and tested all functional and non functional requirements which were proposed by the client . That has been fully completed and all the functionalities are working properly with user friendly manner. The main objective of the author is, developing this system successfully. This chapter is based on to find out achievements that have been accomplished, problems encountered and how they were solved, mistakes made and lesson learnt.

6.1 Problems Encountered

Some problems were encountered when the interviews were conducted with some of the users in the National STD /AIDS Control Program. Those were happened due to lack of their knowledge about the information technology. However after explaining about the objectives of the project , those problems were resolved.

6.2 General Achievements

The Patient Information Management System was undertaken to accomplish an effective and an efficient solution for the existing manual system . Evaluation is clearly defined whether predefined requirements are met at the end of the project. When the requirements changed though changing business circumstances, the system must evolve with change.

The Patient Information Management System provides functionalities for the existing manual system to enhance productivity and efficiency . Also it will reduce the time wastage of manual day to day activities. The system will increase the security of the patients details .Overall target of this project is to fulfill all functional requirements and reaches the quality goals within the defined scope.

User Acceptance Testing (UAT) has been performed in the client side at the target environment with the real data. Set of users there were selected from an organization for performing acceptance testing. They have been examined to find some defects of the system. They provided comments for some defects. It is mostly helpful for enhance the system functionalities and provide high quality software output.

During the project work mapping some manual parts to online system was most challenge to the developer. Also successful completion of this project within limited time period was based on project schedule provided by the PEB of USCS from the beginning. Author gain essential experience and knowledge about subject domain and that will helpful on future.

6.3 Suggestions for Future Encasements

The following improvements or enhancements are possible for this system to provide more facilitates to users and increase the scope of the system.

- All the other STD clinics of hospitals in Sri Lanaka will be connected to the system.
- Implement a new module called " Lab" in future for handling the transactions in labs at National STD/AIDS Control Program.

6.4 **Mistakes Made**

Not documenting some parts thinking that I can remember them. But when the time passed I had to re gather that information because of not remembering them.

6.5 **Lessons Learnt**

Developing a completed software within allocated time schedule with covering all user requirements, managing the works in such situation was important experience gained from the project. Learning about various software engineering concepts and using them practically in a project are good lessons learnt during this project .

References

References

[Terry Quatrani 2002] Terry Quatrani , Visual Modeling with Rational Rose 2002 and UML.

[WWW1] http://www.aidscontrol.gov.lk/me/PMIS/ [03.01.2010]
[WWW2] http://www.buzzle.com/editorials/1-5-2005-63768.asp [24.01.2010]

[WWW3] http://www.developers.com/ [24.01.2010]

[WWW4] http://ezinearticles.com/?Rapid-Application-Development-Model-

(RAD)&id=412312 [24.01.2010]

[WWW5] http://en.wikipedia.org/wiki/Functional_requirement [24.01.2010]
[WWW6] http://www.freetutes.com/systemanalysis/sa2-prototyping-model.html
[WWW7] http://www.augustana.ab.ca/~mohrj/courses/2000.winter/ [25.01.2010]
[WWW8] http://en.wikipedia.org/wiki/Model%E2%80%93View%E2%80%93Controller
[01.03.2010]
[WWW9] http://en.wikipedia.org/wiki/Systems_design [05.03.2010]
[WWW10] http://en.wikipedia.org/wiki/Entity-relationship_model [15.03.2010]
[WWW11] http://www.symfony [21.03.2010]
[WWW12] http://www.symfony-project.org/doctrine/1_2/en/01-Getting-Started
[28.03.2010]
[WWW13] http://jquery.com/[28.04.2010]
[WWW14]
http://www.filamentgroup.com/lab/http://www.filamentgroup.com/lab/update_to_jquery_visualize_accessible_charts_with_html5_from_designing_with/ [24.07.2010]

Appendix A : System Documentation

Appendix A : System Documentation

Pre-requirements of the System

The required software and hardware requirements for running the system are listed in the following two tables.

Hardware requirements

Table A.1 – hardware requirements

	Recommended Hardware Requirements
Processor	Intel Pentium IV or higher processor
Memory	512 RAM
Hard disk	20GB or more Hard disk space
Screen resolution	1280 * 1024

Software requirements

Table A.2 – hardware requirements

	Recommended Software Requirements
Operating system	Windows, Linux
Web server	Apache 2.2
Database server	MySQL 5.0 or later
Web browser	Internet Explorer 6+, Mozilla Firefox
Other	XAMPP

Installation of the System

Installation process consists the following steps.

1. Web and database servers installation
2. Application installation
3. System configuration

1. Web and Database Servers Installation

XAMPP has been selected for installing the Apache, MySQL and PHP packages. XAMPP is bundled with Apache, MySQL, and PHP packages which are used to run the application.

Install XAMPP in Windows

1. Insert Patient Information Management System (PIMS) CD into the CD ROM.
2. Open the directory called "Install".
3. Run the file "xampp-win32-1.7.3.exe" to install the web, database servers and the PHP .
4. Open the XAMPP control panel which is in the installed directory of the XAMPP. (i:e C:/ xampp / xampp-control)
5. If the Apache and MySQL servers are not started , click on the "Start" button to start those services . Following figures show the "Start" and "Stop" statuses of the Apache and MySQL servers.

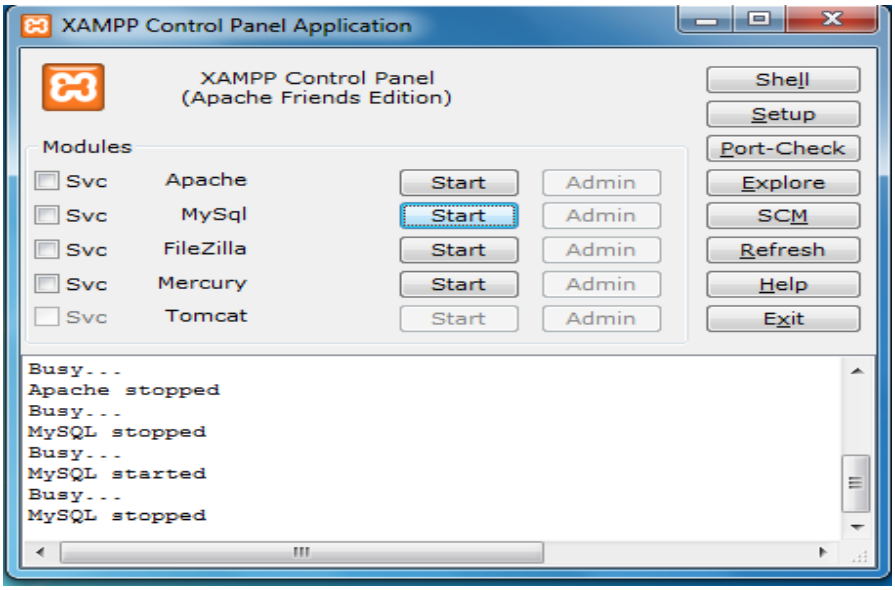

Figure A.1 Apache and MySQL servers are not running.

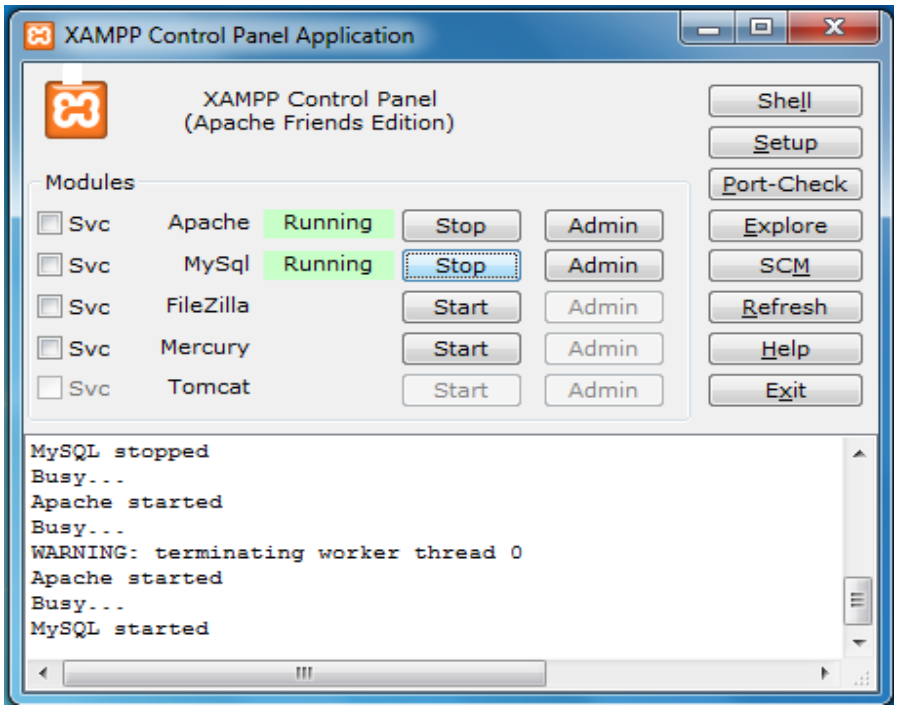

Figure A.2 Apache and MySQL servers are running.

Install LAMPP in Linux

1. Insert Patient Information Management System (PIMS) CD into the CD ROM.
2. Open the "Install" directory and copy the xampp-linux-1.7.3a.tar to a location in the file system. (Note: Copying location should be have Read, Write, Executable permissions)
3. Extract the "xampp-linux-1.7.3a.tar"
4. The web directory "htdocs" is included in the /lampp directory.
5. Start the services by using the following command.
   /opt/lampp/lampp start (Note: Here opt is the directory where lampp is located)

   If the services got started , following messages are displayed

   ```
   Starting XAMPP 1.7.3a...
   LAMPP: Starting Apache...
   LAMPP: Starting MySQL...
   LAMPP started.
   Ready. Apache and MySQL are running.
   ```

2. Application Instillation

1. Open the directory called “System” which is in the CD.
2. Copy the pims.zip or pims.tar.gz file into the web directory of the XAMPP. The web directory “htdocs” is located in the /xampp/ directory. (Note: In Linux /lamp)
3. Extract the copied file.
4. If the application environment is Linux , necessary read write and executable permissions should be set to the application directory . For example,

   chown –R apache.apache pims
   chmod –R 755 pims

3. System Configuration

1. Log in to the MySQL server and create a database with the name “pims” .
2. Select the created database and import the dbscript.sql file which is in the “Data” directory of the CD.
3. Login to the system by using default admin account’s credentials.
4. Following figures explain the database system configuration process.

Appendix B : Design Documentation

Appendix B: Design Documentation

Activity Diagrams

1. Following activity diagram shows the flow of events for saving the STD episode care details

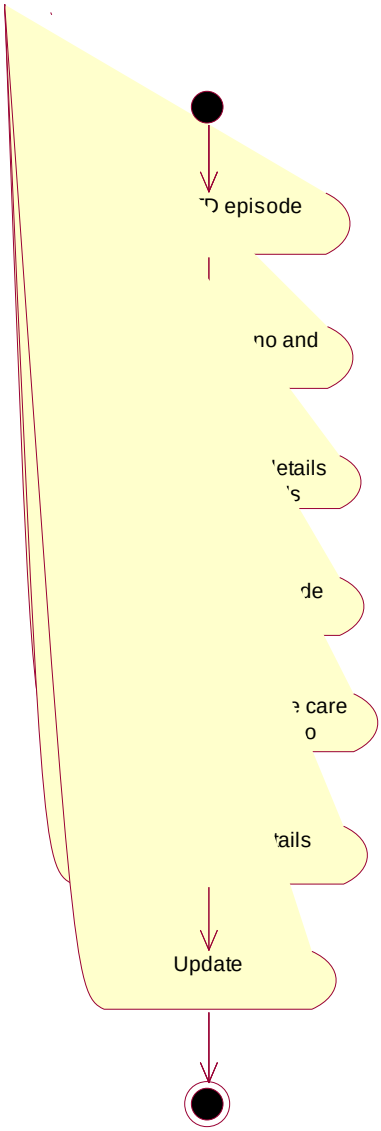

Figure B.1 Activity diagram for saving STD episode care details

2. Following activity diagram shows the flow of events for setting the access rights for a user group.

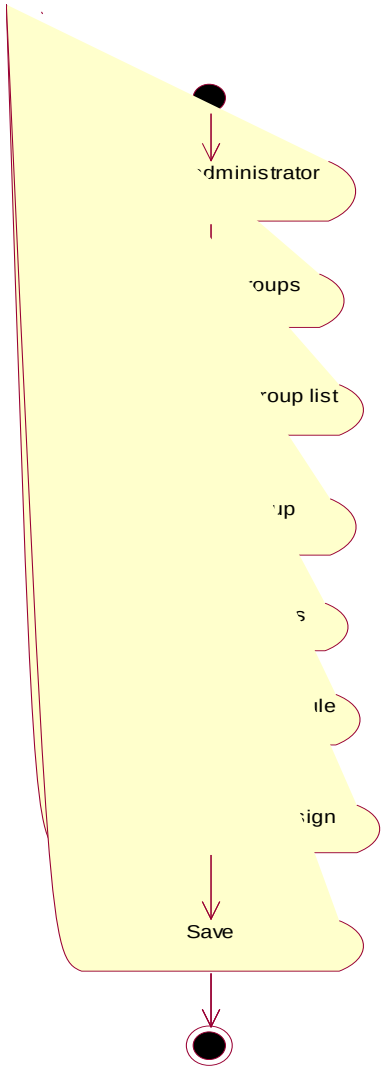

Figure B.2 Activity diagram for setting access rights for a user group.

Sequence Diagrams

This diagram shows the interactions, when the appointment list is displayed.

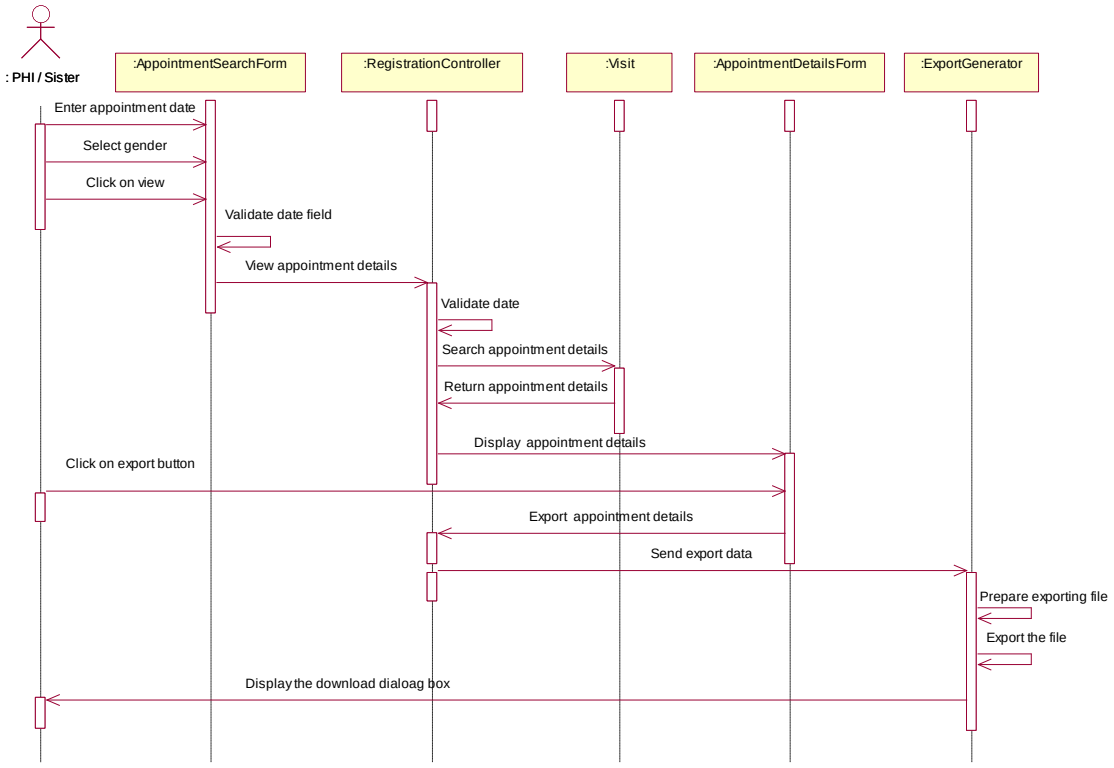

Figure B.3 Sequence diagram for display appointment list.

Appendix C - User Documentation

Appendix C - User Documentation

User Documentation

Following user interfaces explain the functionalities of the system which are carried out by different uses of different user groups.

Login Module

Login

1. Enter the user name and password in relevant fields. (Figure C.1)
2. Click on the “Login” button.
3. If the user name and password are correct , display the main menu according to the

assigned rights of the user.

Figure C.1 Login form

Figure C.2 Login form with wrong password

Figure C.3 Successful login to the system.

Admin Module

1. Create User Groups
    - Select “Users->User Groups” menu option in the “Admin” menu.
    - User group creation page will be displayed once clicked on the “Add” button in user groups list page. (Figure C.4)
    - Enter the group name and description. (Figure C.5)
    - Click on the “Save “ button to save the data .
    - Click on the “Back” button to go to the user groups list page

Figure C.4 User group list

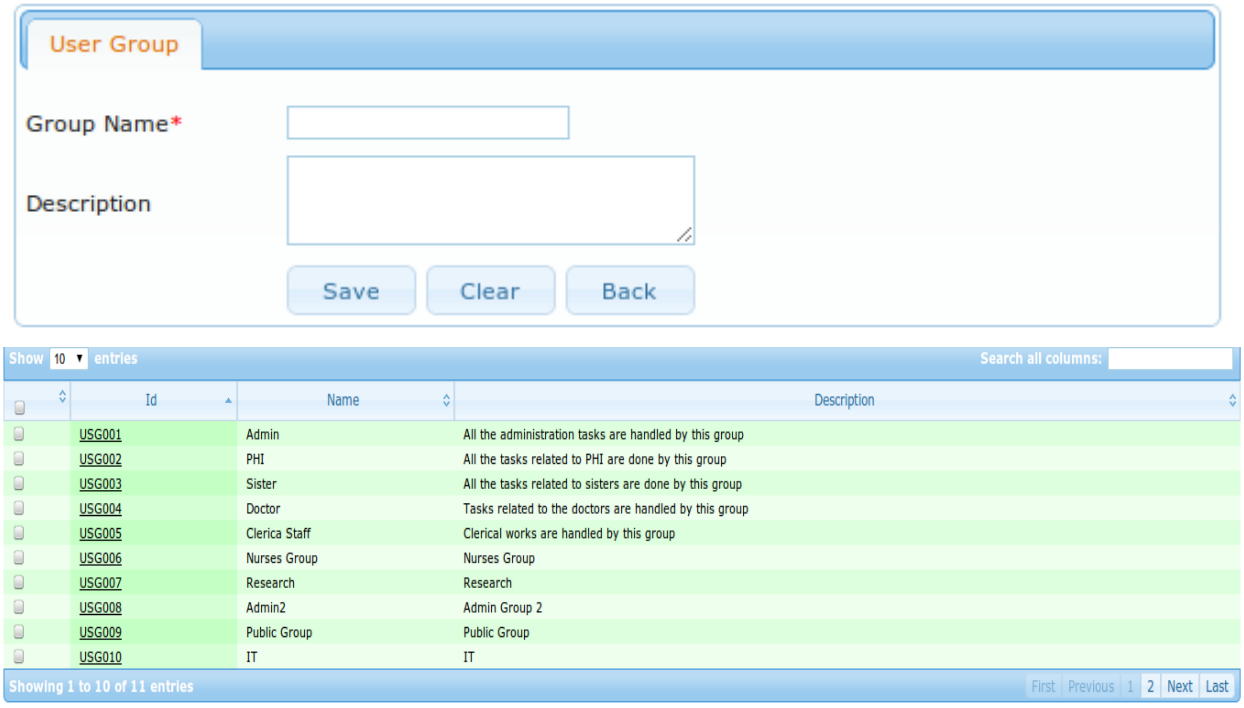

Figure C.5 User group creation page.

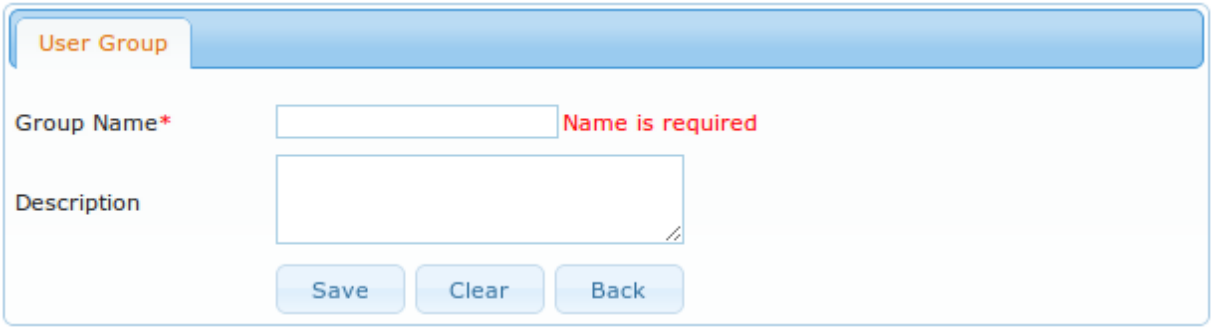

Figure C.6 Empty group name

2. Edit User Groups

- User groups list will be displayed once selected the "Users->User Groups" menu option in the "Admin" menu.
- Edit view is displayed after clicking on the relevant user group id link in the list. (Figure C.7)
- Change the required data.
- Click on the "Save" button to save the changes.

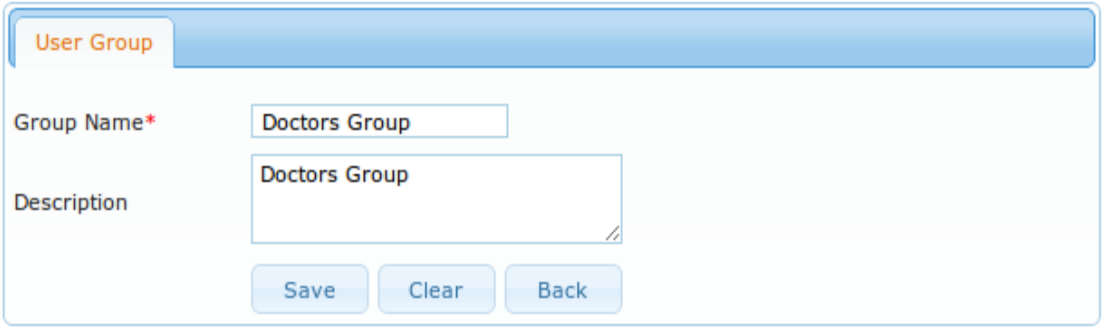

Figure C.7 Edit view of a user group

3. Delete User Group

- User groups list will be displayed once selected the "Users->User Groups" menu option in the "Admin" menu.
- Select the check box/s of user group/s which need to be deleted.
- Click on the "Delete" button.
- Click on the "Ok" button in the confirmation dialog box for deleting the record/s.

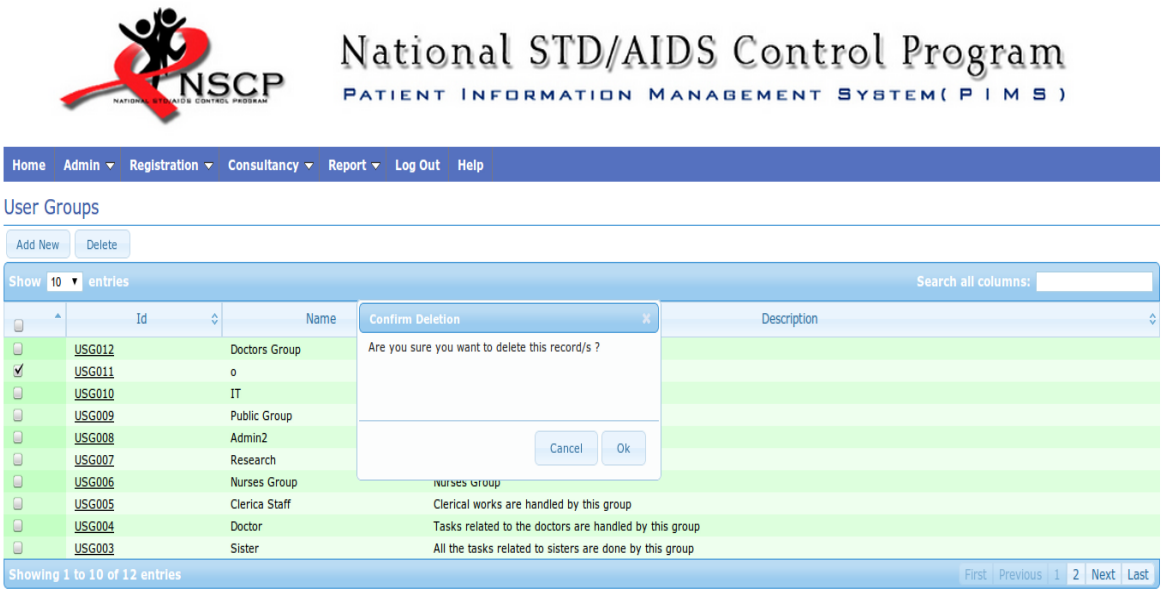

Figure C.8 Delete an user group

4. Search User Group

- User groups list will be displayed once selected the “Users->User Groups” menu option in the “Admin” menu.
- Type search value in the “Search all columns” text box.

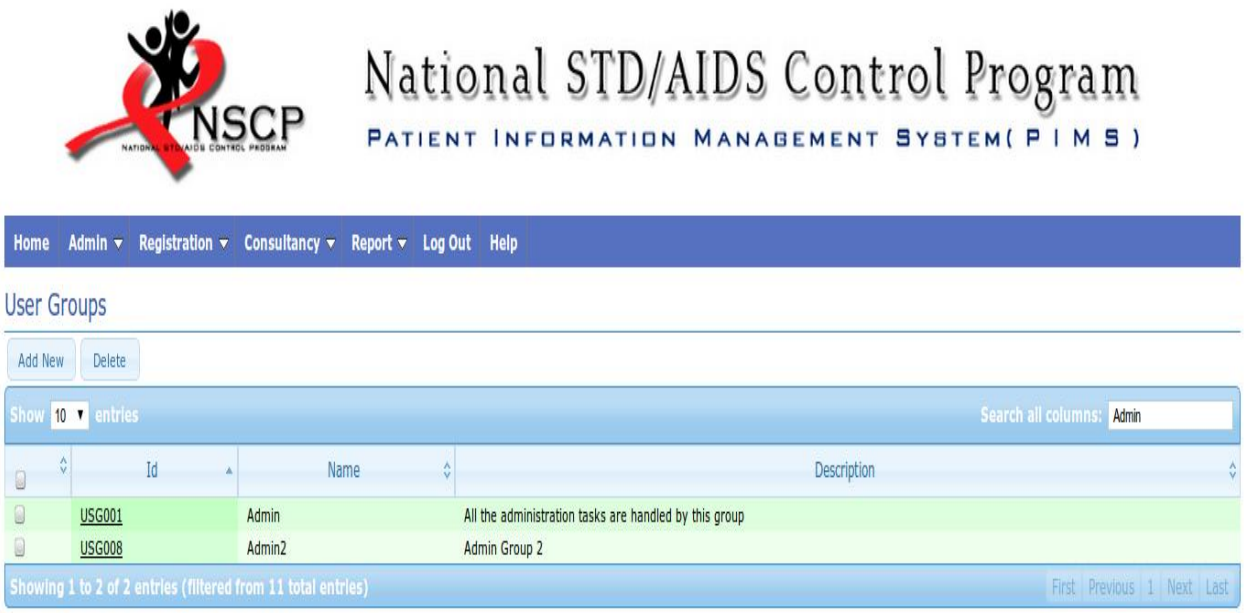

Figure C.9 Search user group

5. Create User Accounts

- Select "Users->User Accounts" menu option in the "Admin" menu.
- User account creation page will be displayed once clicked on the "Add" button in the user accounts list page (Figure C.10)
- Enter the required data.
- Click on the "Save "button to save the data.
- Click on the "Back" button to go to the user accounts list page.

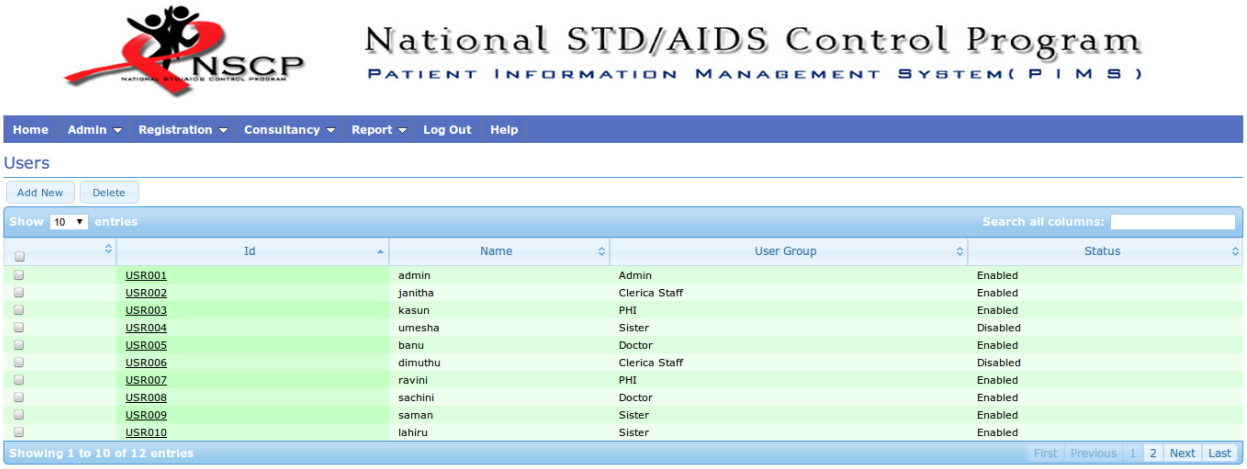

Figure C.10 Users List

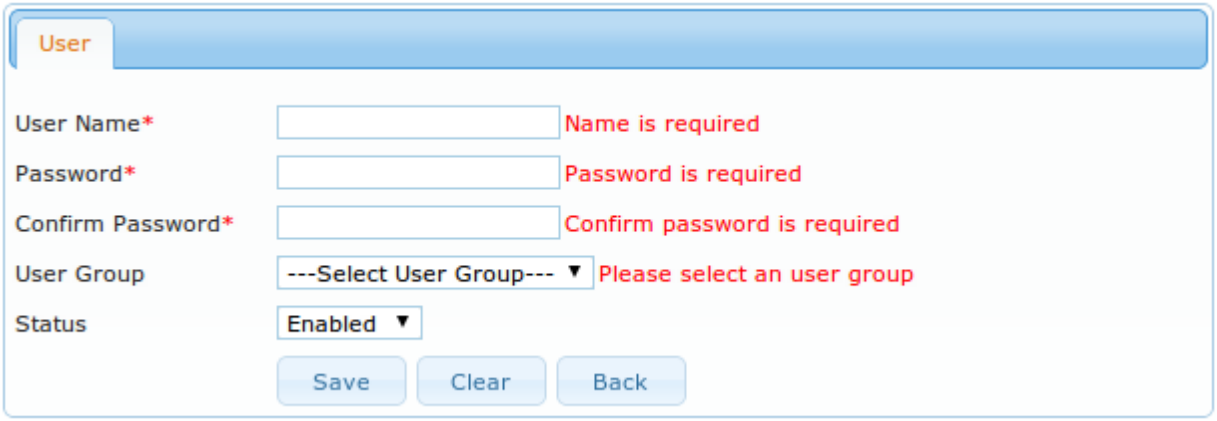

Figure C.11 Invalid data entered for the fields.

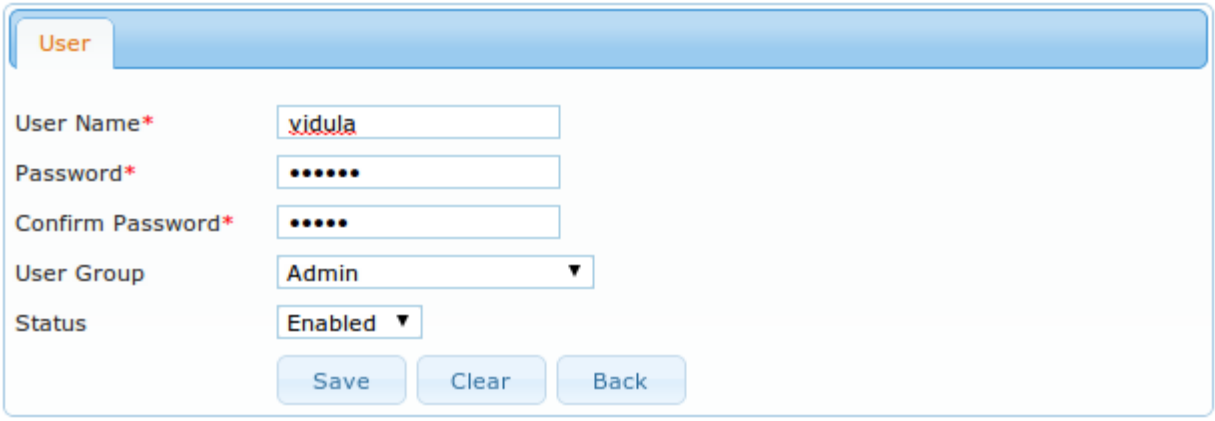

Figure C.12 Fields with proper data.

6. Assign Rights for User Groups

- Select “Users->User Groups” to open the user groups list (Figure C.4).
- Click on the user group id link which has to be assigned rights.
- Click on the “Assign Rights” button .
- Select the module from the “Module” drop down list.
- Select the required options from “Add, Edit, Delete, View” to assign (“View” is a must for any module).
- Click on the “Save” button to save the data.

Figure C.13 Click on the assign button to add user rights.

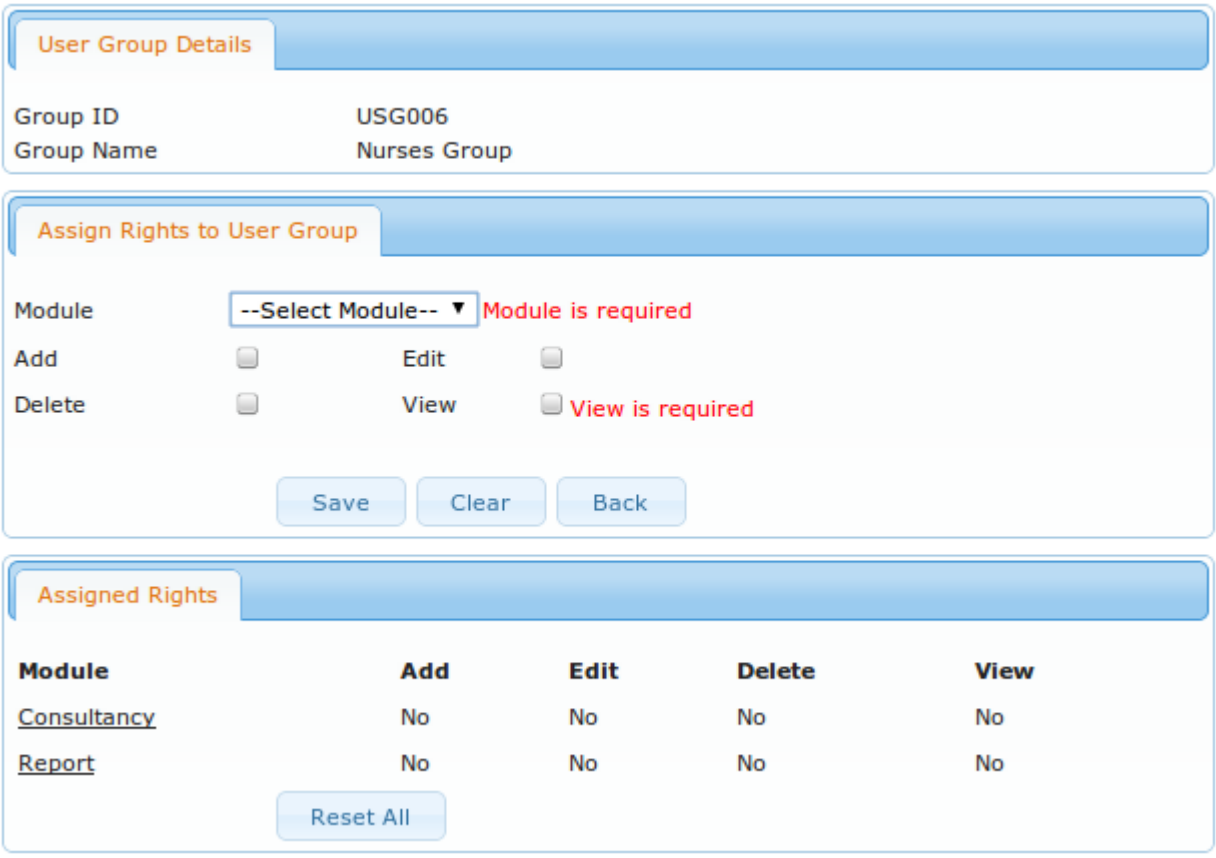

Figure C.14 Compulsory field's validation.

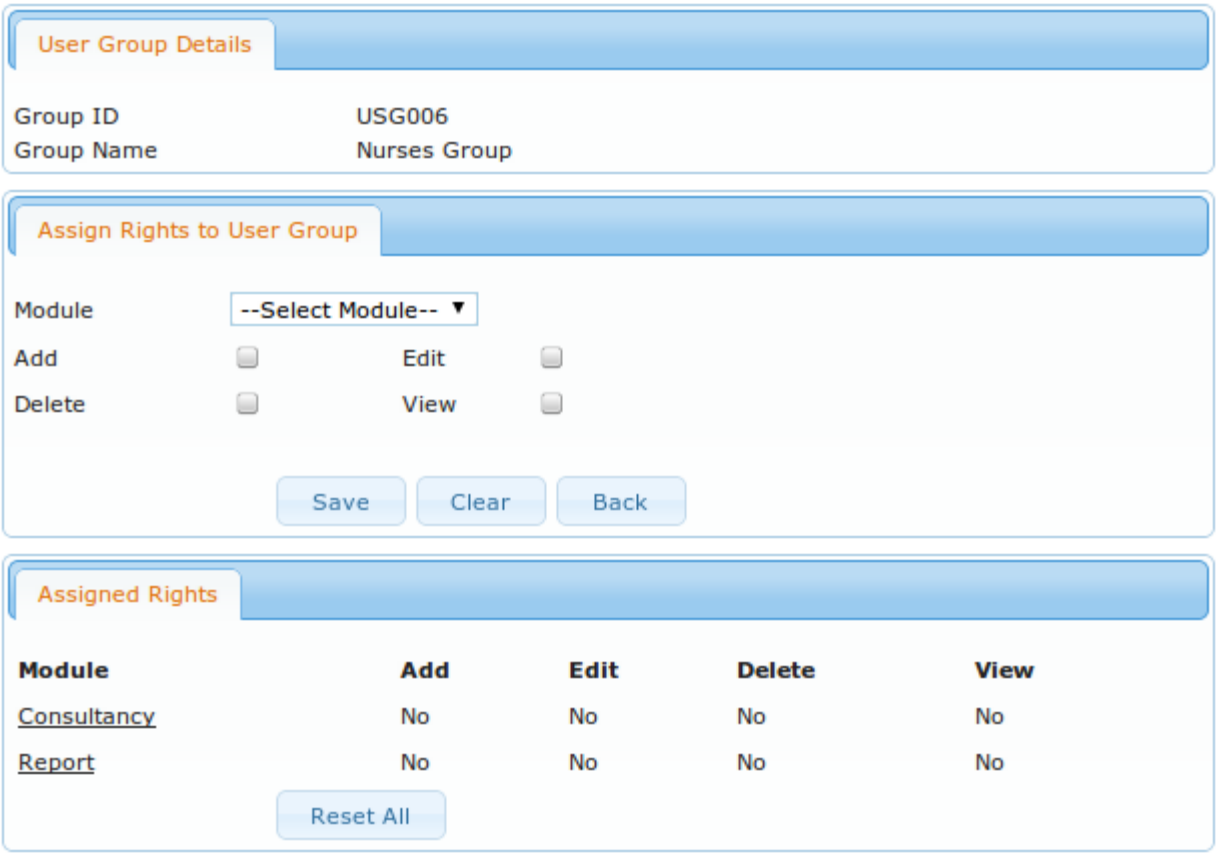

Figure C.15 Before assigning a module.

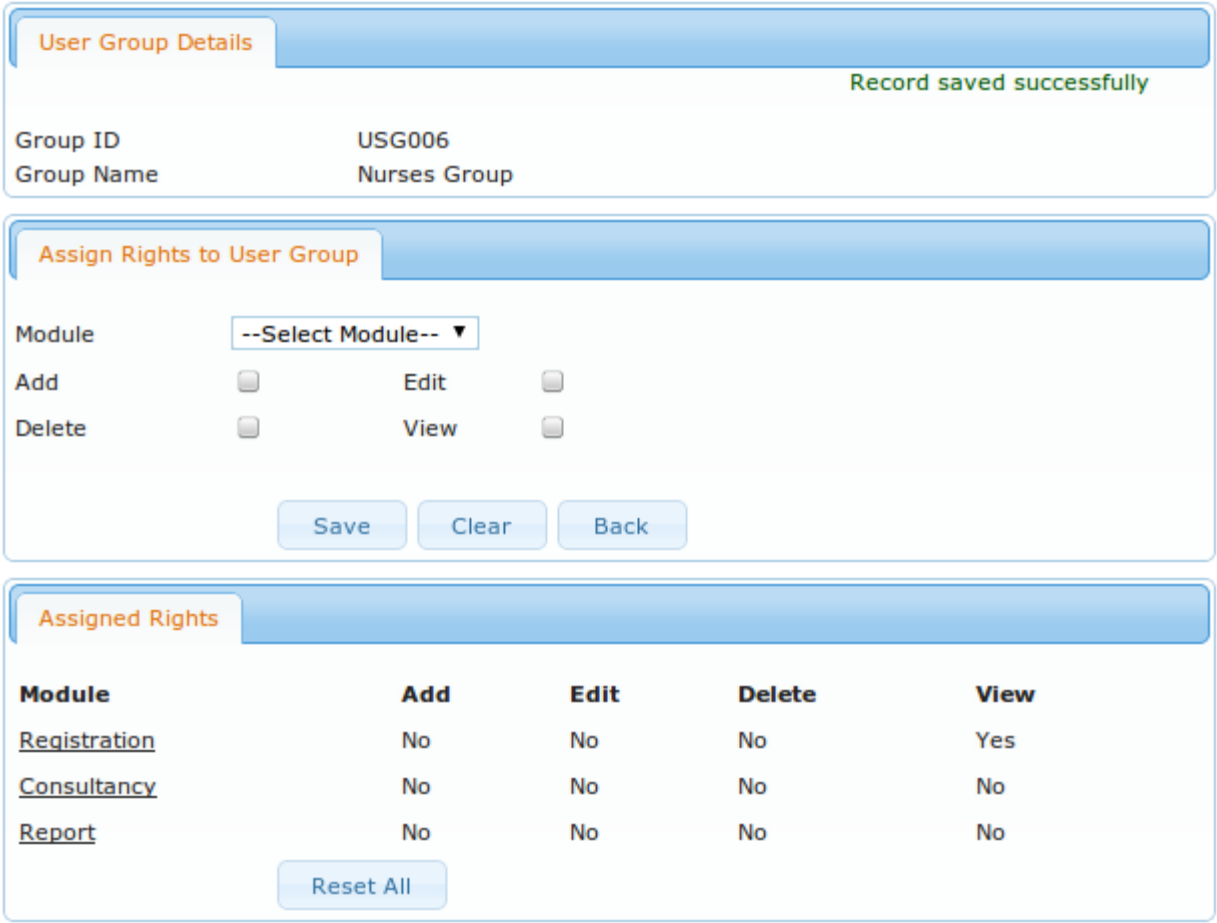

Figure C.16 After assigning a module.

7. Reset All Assigned Rights
   - Click on the “Reset All” button in the “Assigned Rights” area.
   - Click on “Ok” button in the confirmation dialog box for resetting all the rights.

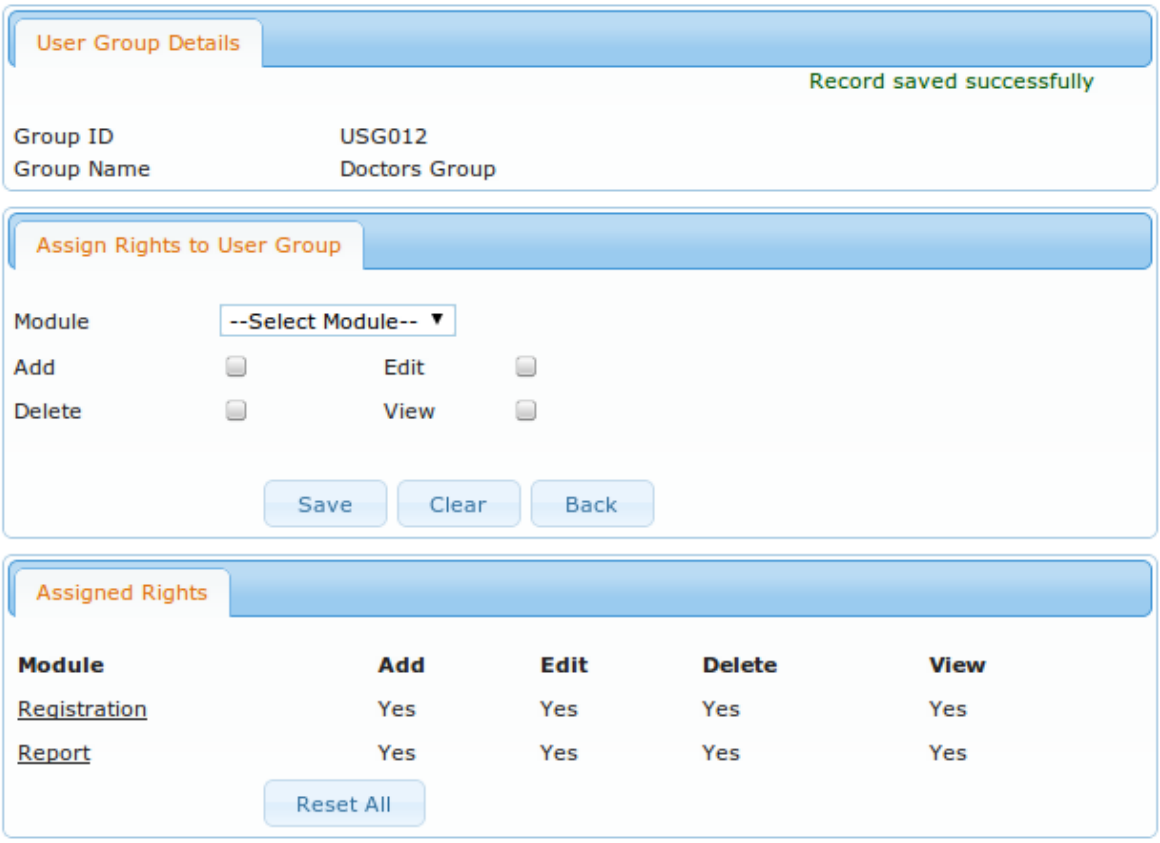

Figure C.17 Before resetting the rights.

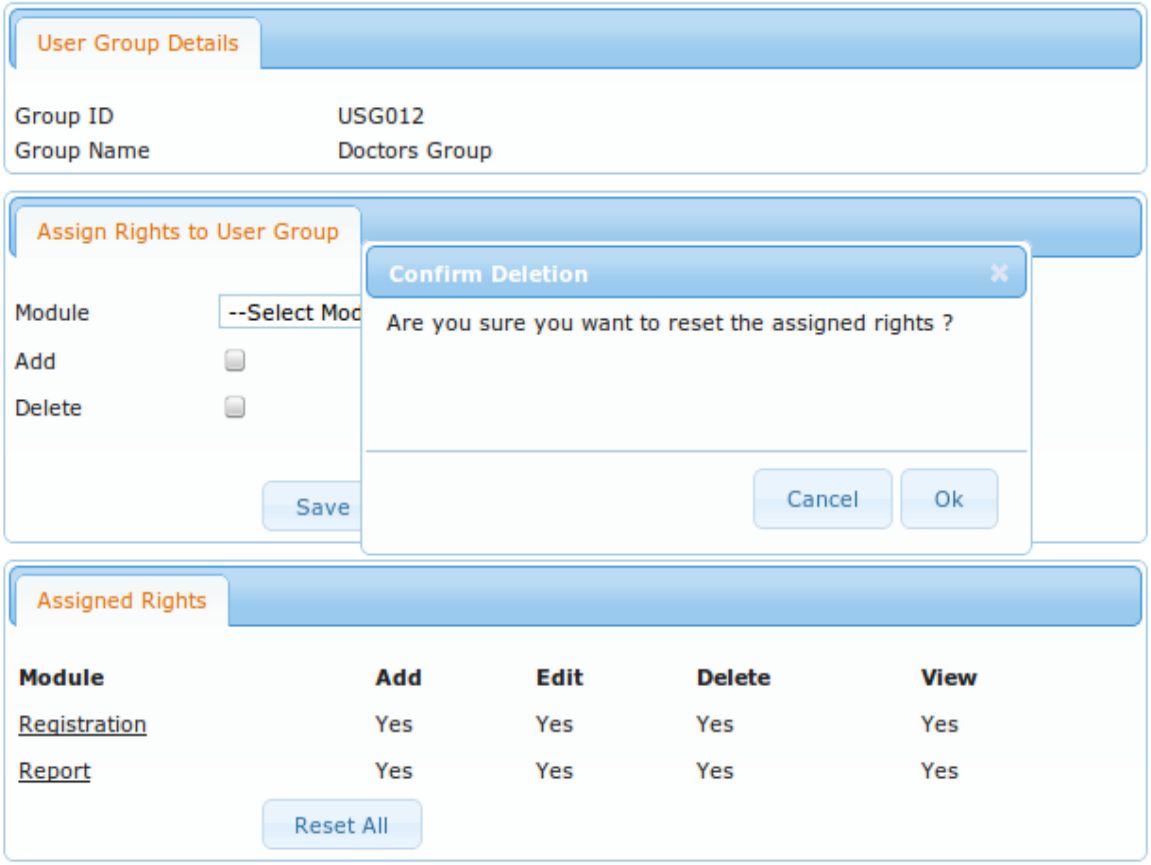

Figure C.18 Get the confirmation before resetting the rights.

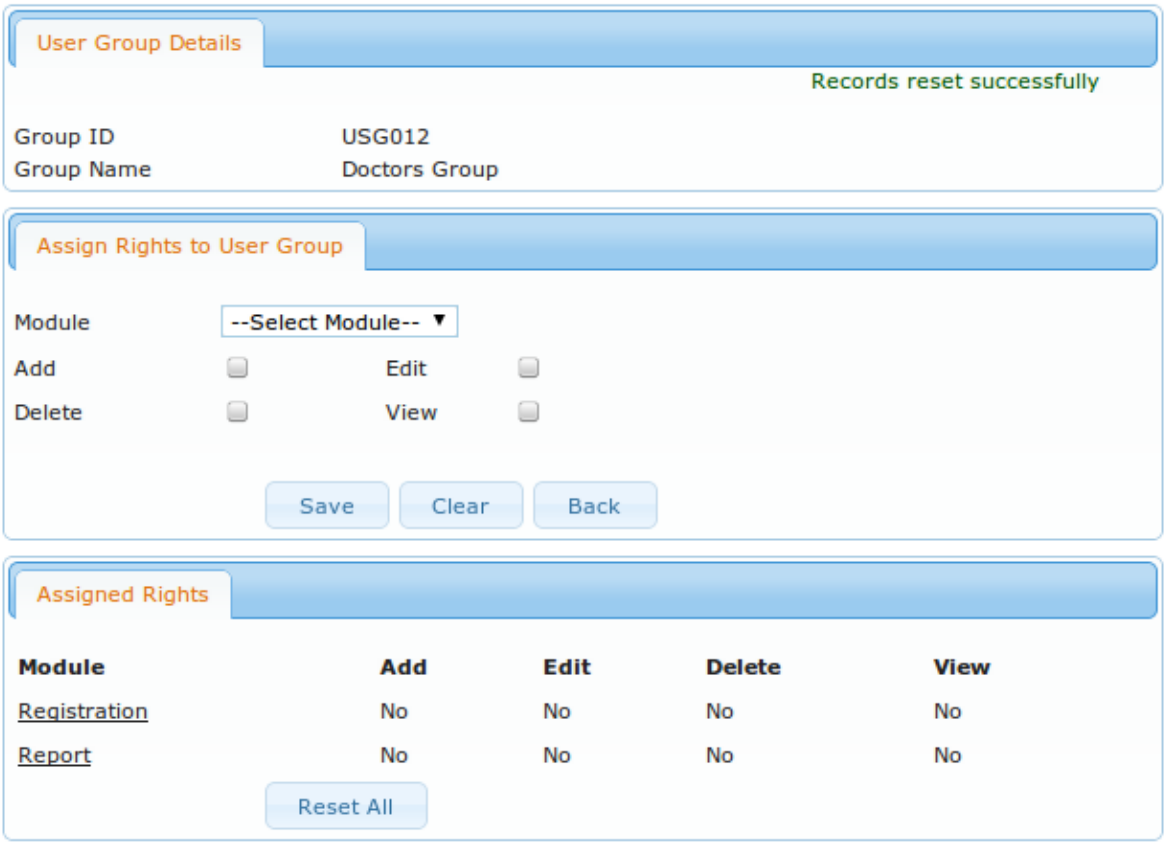

Figure C.19 After resetting the rights.

Registration Module

1. Patient Registration

   - Click on the “Patient Registration” in “Registration” menu
   - Click on “New Patient “button in the patient registration page.
   - Click on the “Personal Information Tab” for entering personal data.
   - Click on the “Contact Information Tab” for entering contact details.
   - Click on the “Reason for Attendance Information Tab” for entering attendance reasons.
   - Click on the “Save” button to save data.

Figure C.20 Select "New Patient" for patient registration form.

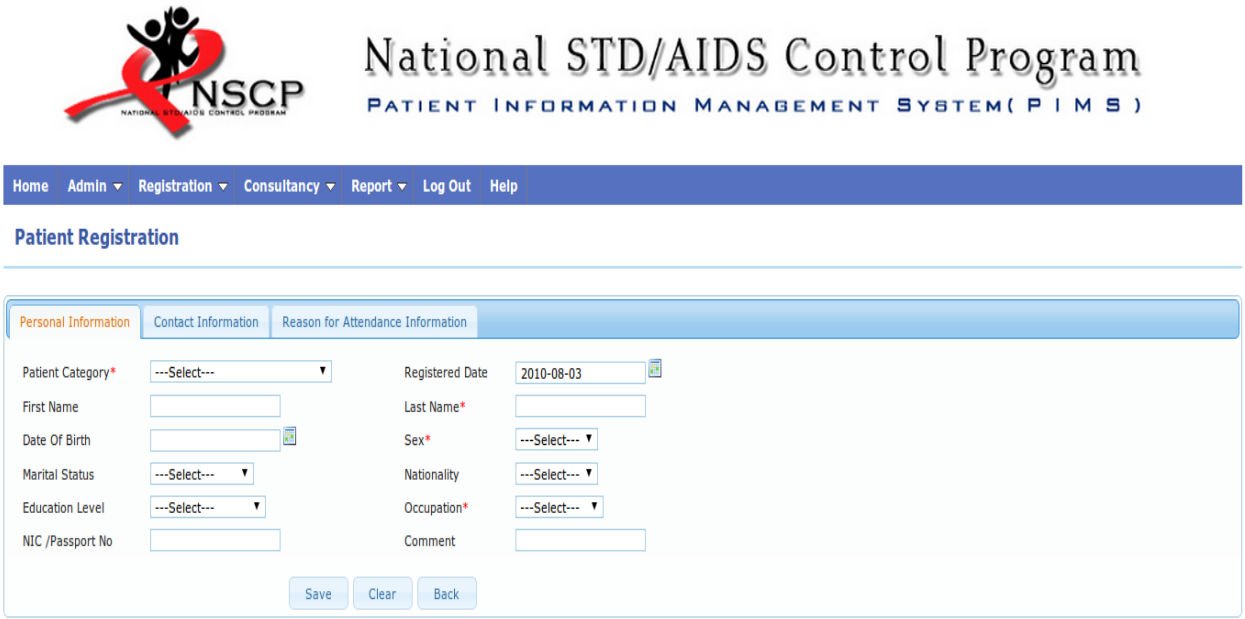

Figure C.21 Main registration form.

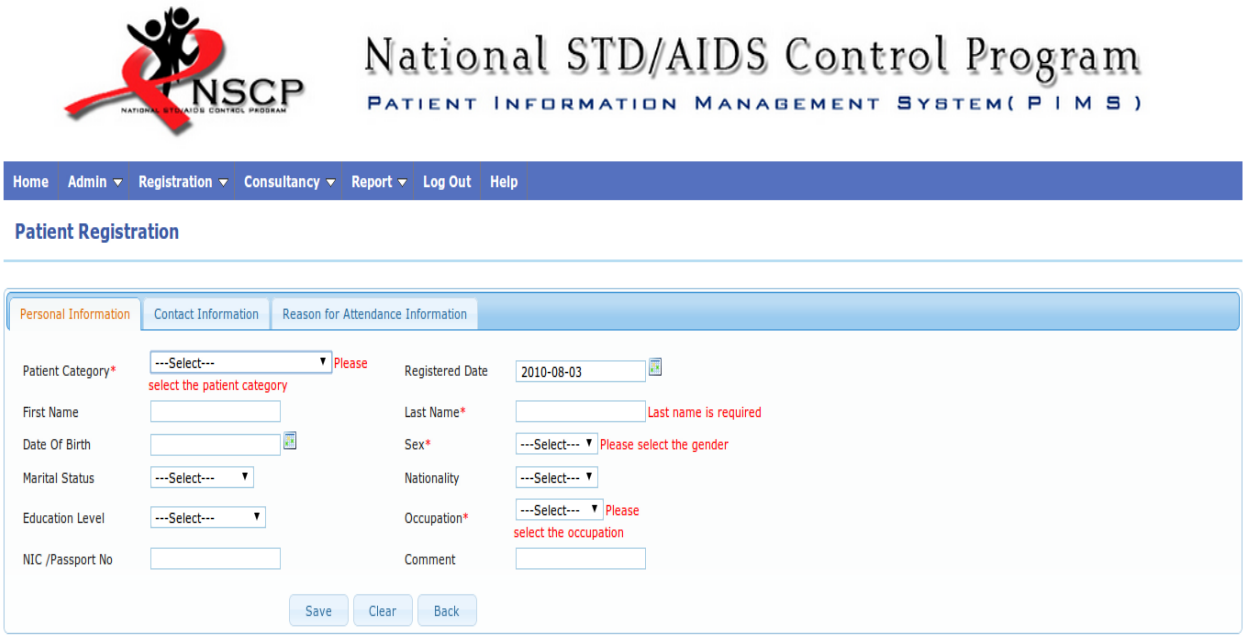

Figure C.22 Validation of registration from.

Figure C.23 Registration form with data.

2. Edit Patient Information

- Click on the "Patient Registration" in "Registration" menu.
- Enter the patient no or name to search in the patient registration page.
- Click on the "Search" button.
- In the edit view, select the relevant tab for editing.
- Click on the "Save" button to save the data.

Figure C.24 Search a patient.

Figure C.25 Edit view of a patient.

Figure C.26 Confirmation message of edited record.

8. Mark And Create Visit

- Select "Registration->Mark / Create Visit" menu option in the "Registration" menu.
- Enter the patient no in the search view.
- Click on the "Search" button.
- Click on the "Show Visits" link of an episode in episode history page.
- Enter appointment or visited date .
- Click on the "Save" button.

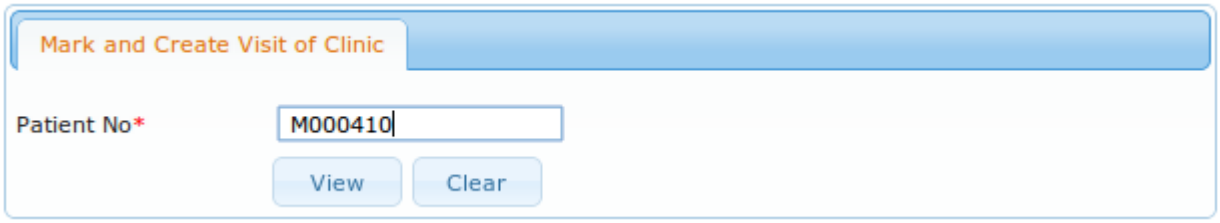

Figure C.27 Search a patient for updating the clinic attendance records.

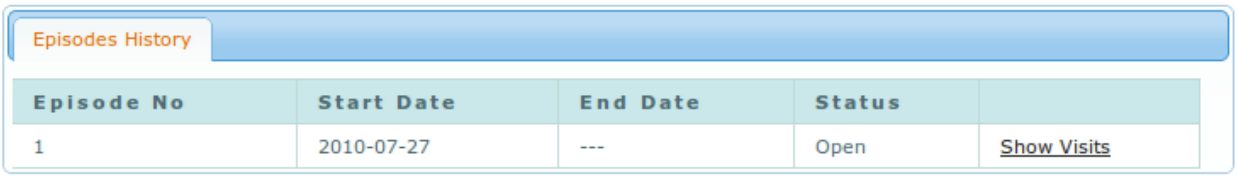

Figure C.28 Clinic visit records of episodes .

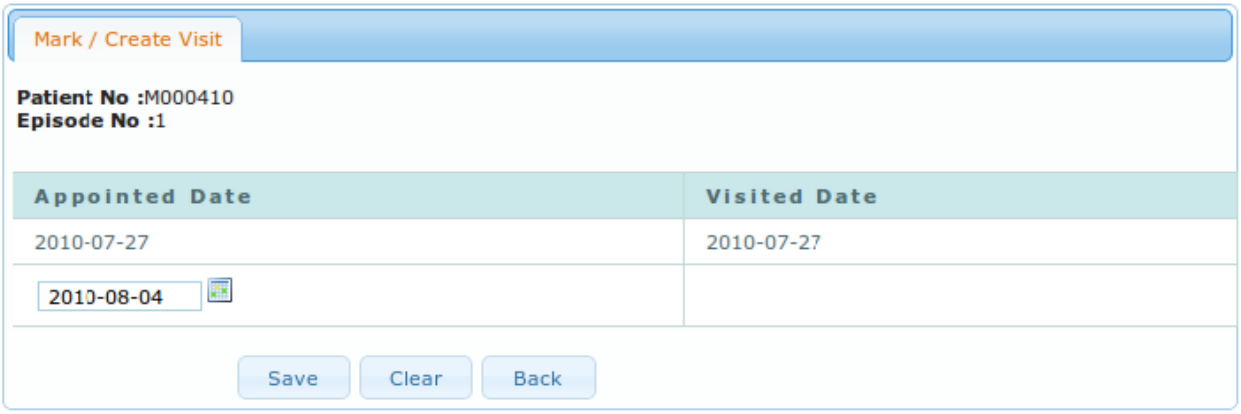

Figure C.29 Update the clinic visit record .

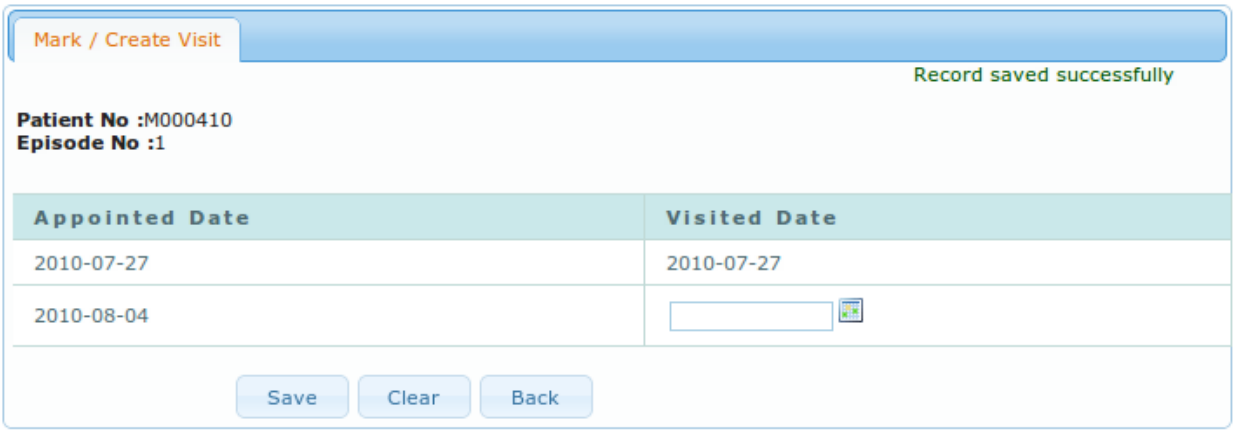

Figure C.30 Updated the clinic visit record.

9. View Appointment List

- Select “Registration->Appointment List” menu option in the “Registration” menu.
- In the clinic appointment searching area , select date an the gender to be searched.
- Click on the “View” button.
- Click on the “Export” button to export the report.

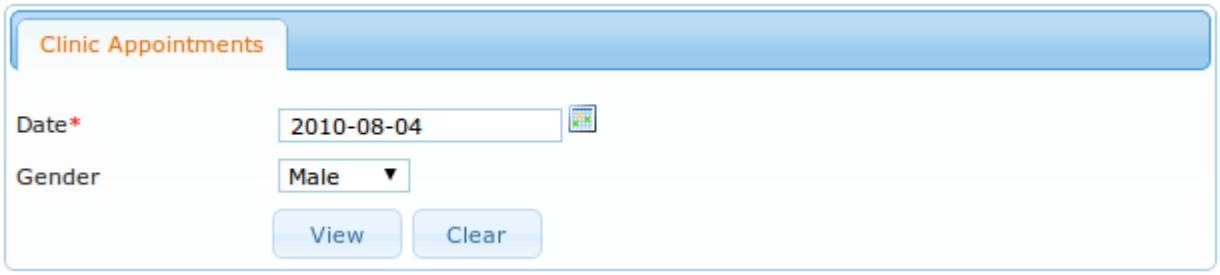

Figure C.31 Search appointment list of a particular day.

Appointments Details of Male Patients on 2010-08-04

Patient No	Patient Name	Visit No
M000410	Gayan Sudesh	2
M000510	Thilina Niranjan	2
M000610	Saman Perera	2
M000710	Kamal Gunathilaka	2
M000810	Lal Kumara	2
M001009	Chamara Jayasooriya	3

Print Export Back

Figure C.32 Appointment list.

10. Create Patient Category

- Existing category list will be displayed once selected the "Registration->Patient Category" menu option in the "Registration" menu (Figure C.38).
- Click on the "Add" button to open the category creation page.
- Enter the category name and description.
- Click on the "Save" button.

National STD/AIDS Control Program

PATIENT INFORMATION MANAGEMENT SYSTEM(P I M S)

Home Admin Registration Consultancy Report Log Out Help

Patient Category

Add New Delete

Record Saved Successfully

Show 10 entries Search all columns:

	Id	Name	Description
	PAC001	STD	STD
	PAC002	HIV	HIV
	PAC003	Completion of Episode of Care	Completion of Episode of Care
	PAC004	Examination	Examination
	PAC005	History	History
	PAC006	Summary Of Sexual History	Summary Of Sexual History
	PAC007	yy	yyyyyy
	PAC008	STD 2	STD 2
	PAC009	Clinic	Clinic
	PAC010	General	General

Showing 1 to 10 of 10 entries First Previous 1 Next Last

Figure C.33 Category list of patients..

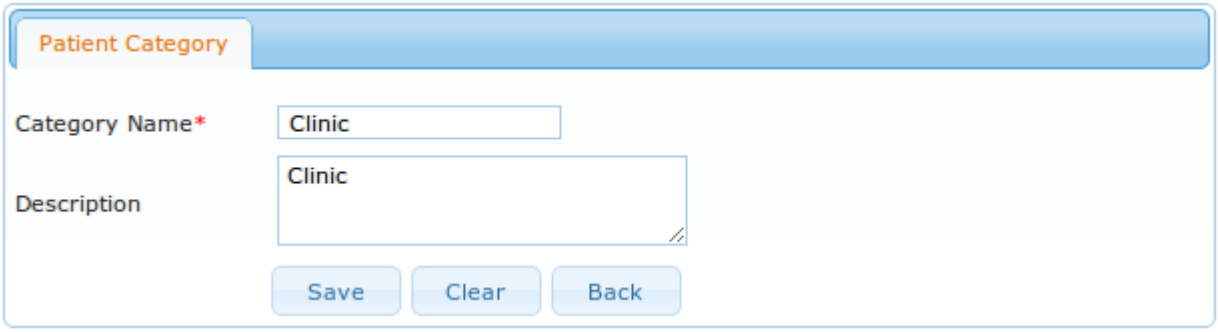

Figure C.34 Add new patient category.

Consultancy Module

1. Add STD Episode Care

- Select "Consultancy->STD Episode " in "Consultancy" modul**e.**
- Enter the patient no
- Click on the "Search" button.
- Click on the "Show Episode Care" link in episode details area.
- Select the relevant tab to add clinical records in the STD episode of care area.
- Add the clinical data.
- Click on the "Save" button to save data.

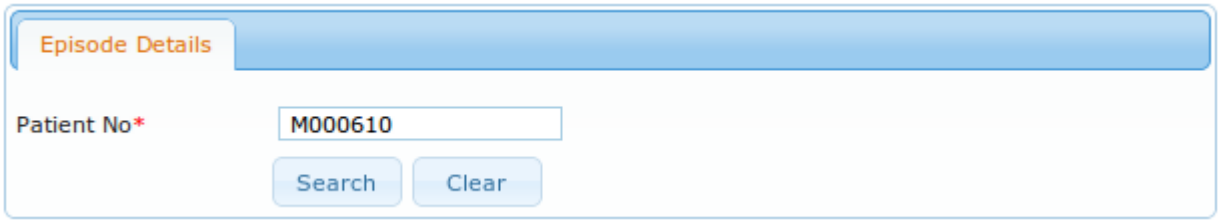

Figure C.35 Search episode details of a patient.

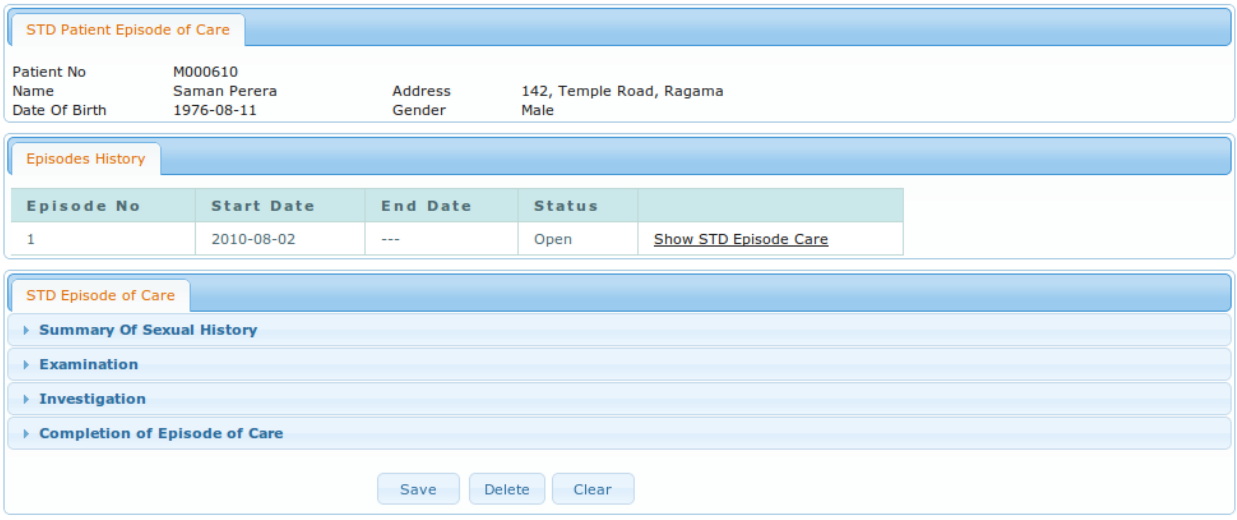

Figure C.36 Episode details of a patient.

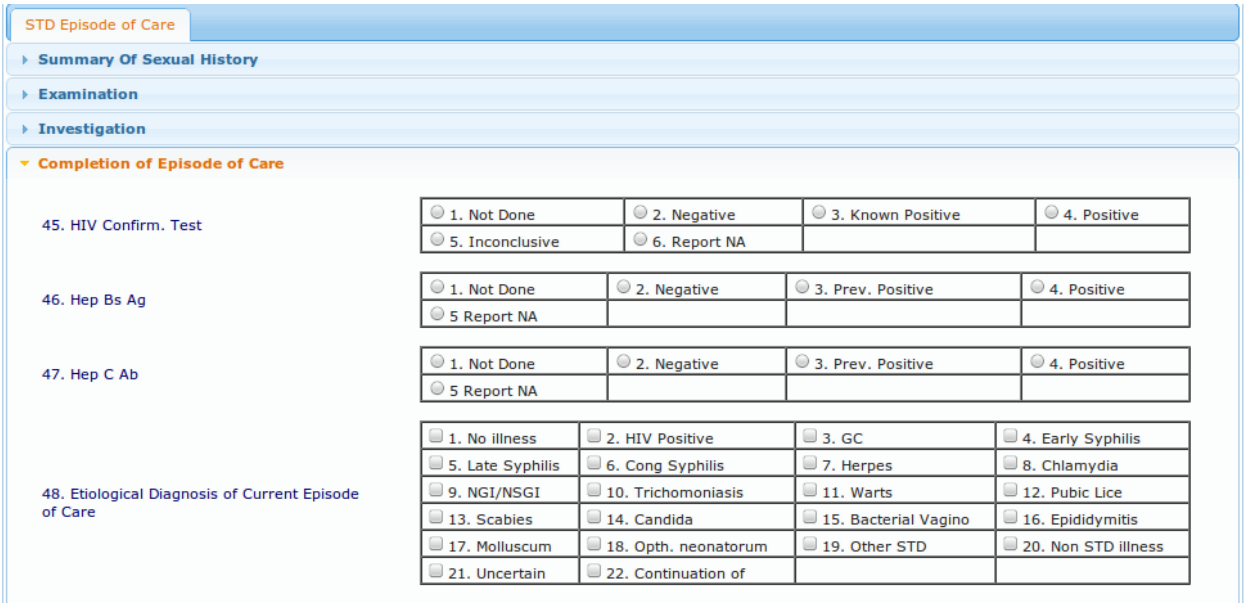

Figure C.37 STD episode care details.

Completion of Episode of Care

45. HIV Confirm. Test	◉ 1. Not Done	○ 2. Negative	○ 3. Known Positive	○ 4. Positive
	○ 5. Inconclusive	○ 6. Report NA		
46. Hep Bs Ag	◉ 1. Not Done	○ 2. Negative	○ 3. Prev. Positive	○ 4. Positive
	○ 5 Report NA			
47. Hep C Ab	○ 1. Not Done	○ 2. Negative	○ 3. Prev. Positive	○ 4. Positive
	○ 5 Report NA			
48. Etiological Diagnosis of Current Episode of Care	☐ 1. No illness	☐ 2. HIV Positive	☐ 3. GC	☐ 4. Early Syphilis
	☑ 5. Late Syphilis	☑ 6. Cong Syphilis	☐ 7. Herpes	☐ 8. Chlamydia
	☐ 9. NGI/NSGI	☐ 10. Trichomoniasis	☐ 11. Warts	☐ 12. Pubic Lice
	☐ 13. Scabies	☐ 14. Candida	☐ 15. Bacterial Vagino	☐ 16. Epididymitis
	☐ 17. Molluscum	☐ 18. Opth. neonatorum	☐ 19. Other STD	☐ 20. Non STD illness
	☐ 21. Uncertain	☐ 22. Continuation of		
49. Syndrome	☐ 1. NA	☐ 2. GUD - non vesicul	☐ 3. GUD - vesicular	☐ 4. Opth. neonatorum
	☐ 5. Vaginal discharge	☐ 6. Lower abd. pain	☐ 7. Urethral discharg	☐ 8. Scrotal swelling
	☐ 9. Other			
50. Treatment	☐ 1. None	☐ 2. Penicillin	☐ 3. Doxycycline	☐ 4. Cryotherapy
	☐ 5. Podophyllin	☐ 6. TCA - Trichloroac	☐ 7. Metranidazole	☐ 8. Scabicides
	☐ 9. Macrolides	☐ 10. Cephalosporins	☐ 11. Quinolones	☐ 12. Antifungals
	☐ 13. Aciclovir	☐ 14. Cotrimoxazole	☐ 15. Others	
51. Status of Episode	○ 1. Completed	○ 2. Referred	○ 3. Defaulted	◉ 4. Episode to be con
	○ 5. Other			
52. No of Visits	○ 1. One	◉ 2. Two	○ 3. Three	○ 4. Four
	○ 5. Five	○ 6. Six	○ 7. Seven	

Figure C.38 Updated STD episode care details.

Appendix D - Management Reports

Management Reports

Various reports of the patients were taken by quarterly and yearly. Those reports are used to make various decisions in terms of STD patients. Following are the main reports generated from the system.

To generate the graphs , Jquery visualize plug-in was used. [WWW9]

National STD/AIDS Control Program
Central STD Clinic
Colombo 10

New Episodes of STD Recorded From 2010-07-01 to 2010-09-30 in Quarter 3

	0-14		15-19		20-24		25-29		30-34		35-39		40-44		45-49		50+		Total		Grand Total
	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	
No illness										1										1	1
HIV Positive										1										1	1
GC										1										1	1
Early Syphilis													1	1					1	1	2
Late Syphilis													1	1					1	1	2
Cong Syphilis										1			1	1					1	2	3
Herpes								1												1	1
Chlamydia																					
NGI/NSGI																					
Trichomoniasis										1										1	1
Warts																					
Pubic Lice																					
Scabies																					
Candida										1										1	1
Bacterial Vagino																					
Epididymitis																					
Molluscum																					
Opth. neonatorum										1										1	1
Other STD																					
Uncertain																					

Figure D.1 New episodes of STD patients

National STD/AIDS Control Programme
Central STD Clinic
Colombo 10

New Episodes of STD Recorded Among Commercial Sex Workers For The Quarter 3 in 2010

	0-14		15-19		20-24		25-29		30-34		35-39		40-44		45-49		50+		Total		Grand Total
	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	
No illness										1										1	1
HIV Positive										1										1	1
GC										1										1	1
Early Syphilis														1						1	1
Late Syphilis														1						1	1
Cong Syphilis										1				1						2	2
Herpes																					
Chlamydia																					
NGI/NSGI																					
Trichomoniasis										1										1	1
Warts																					
Pubic Lice																					
Scabies																					
Candida										1										1	1
Bacterial Vagino																					
Epididymitis																					
Molluscum																					
Opth. neonatorum										1										1	1
Other STD																					
Uncertain																					

Figure D.2 New episodes of STD among commercial sex workers.

National STD/AIDS Control Program
Central STD Clinic
Colombo 10

Marital Status of STD Clinic Attendees - Quarter 3 in 2010

	Males	Females	Total
Single	3	2	5
Married	1	2	3
Sep / Divo		1	1
Widowed		1	1
Living Together	1		1
Not Known			
Total	5	6	11

Figure D.3 Marital status of STD clinic attendees.

National STD/AIDS Control Program
Central STD Clinic
Colombo 10

Occupations of STD Clinic Attendees Quarter 3 in 2010

	Males	Females	Total
Unemployed	1	1	2
Student	2	1	3
CSW		3	3
Retired	1		1
Other	3		3
Not Known			
Total	**7**	**5**	**12**

Figure D.4 Occupations of STD clinic attendees.

National STD/AIDS Control Program
Central STD Clinic
Colombo 10

Reasons for Clinic Attendance - Quarter 3 in 2010

	Males	Females	Total
Contact of Patient	1		1
Voluntary	2	1	3
Referral from magistrate / Court		2	2
Blood Bank	1		1
Other	2	3	5
Total	**6**	**6**	**12**

Figure D.5 Reasons for clinic attendance.

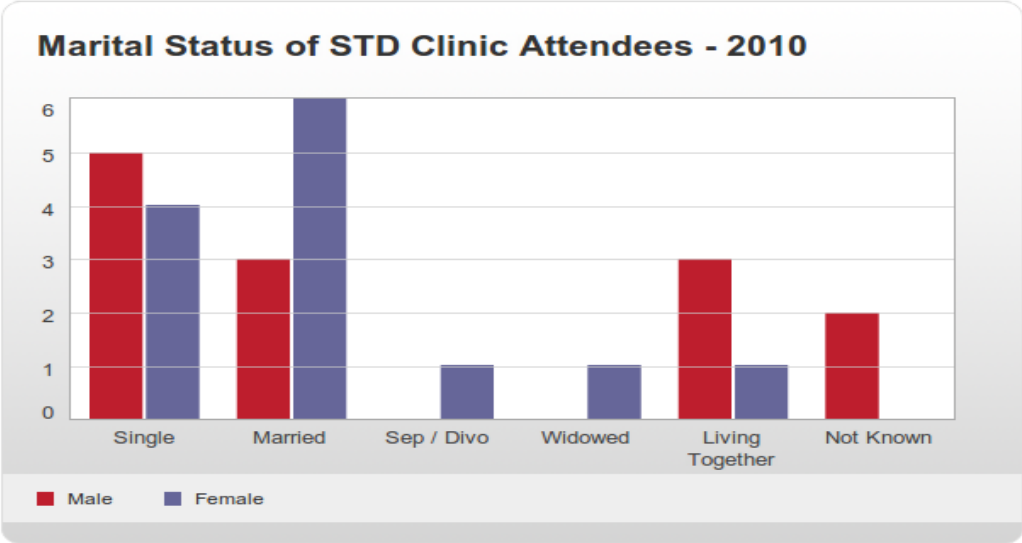

Figure D.6 Marital status of STD clinic attendees by yearly.

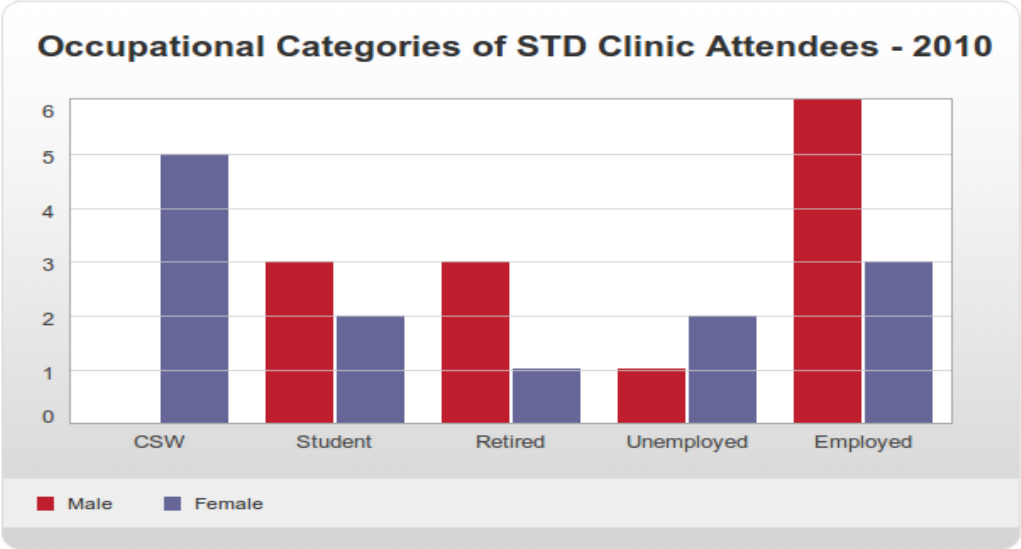

Figure D.7 Occupational categories of STD clinic attendees by yearly

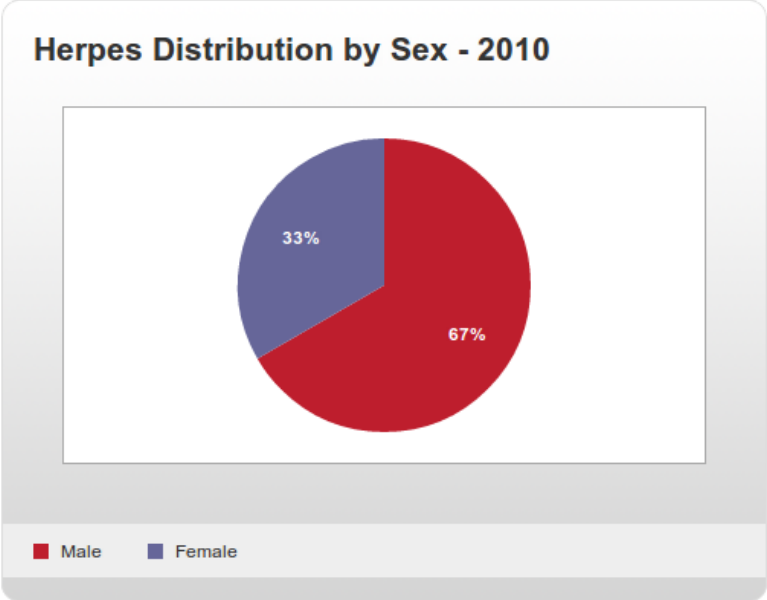

Figure D.8 STD distribution by sex.

Appendix E – Test Results

Appendix E – Test Results

Table E.1 Test cases for Admin Module

Test	Test Description	Expected Result	Status
1	Create user group	New user group should be added to the PIMS	Succeeded
2	Edit user group details	Existing user group details should be edited	Succeeded
3	Delete user group	Existing user group should be deleted if needed.	Succeeded
4	Users should be able to search / view user group details in the list view of user groups	User group details should be displayed with a link for viewing more details	Succeeded
5	View list of user groups	All the user groups should be listed with their group id, name, and description	Succeeded
6	Assign access rights and modules for a user group	System should be able to assign modules and their access rights for user groups	Succeeded
7	Reset access rights of a user group	All the assigned rights should be removed	Succeeded
8	Change access rights of a user group	Existing access rights should be changed	Succeeded
9	Create user account	New user account should be added to the PIMS	Succeeded
10	Edit user account details	Existing user account details should be edited	Succeeded
11	Delete user account	Existing user account should be deleted if needed.	Succeeded
12	Users should be able to search / view user account details in the list view of user account	User account details should be displayed with a link for viewing more details	Succeeded

13	View list of user accounts	All the user account should be listed with their user id, name, and status	Succeeded
14	Administrator should be able to change the password of users.	Password should be changed and email of new credentials should be sent to the user by PIMS	Succeeded
15	Administrator should be able to disable an user account.	Particular user should not allow to login to the system and display the error message	Succeeded

Table E.2 Test cases for Consultancy Module

Test	Test Description	Expected Result	Status
1	Create input category	Input category details should be added to the system	Succeeded
2	Edit an existing input category	Existing input category details should be changed.	Succeeded
3	Delete input category	Existing input category should be deleted if needed.	Succeeded
4	Users should be able to search / view input category details in the list view	Input category details should be displayed with a link for viewing more details	Succeeded
5	View list of input categories	All the input categories should be listed with their id, name	Succeeded
6	Create input question	Input question details should be added to the system	Succeeded
7	Edit an existing input question	Existing input question details should be changed.	Succeeded
8	Delete input question	Existing input question should be deleted if needed.	Succeeded

9	Users should be able to search / view input question details in the list view	Input question details should be displayed with a link for viewing more details	Succeeded
10	View list of input questions	All the input questions should be listed with their id, name	Succeeded
11	Assign results for the input question	Multiple results should be assigned to a input question	Succeeded
12	Delete result from an input question	Assigned results should be removed from an assigned question	Succeeded
13	Edit result of an input question	Result of an input question should be edited	Succeeded
14	Generate STD episode care form for entering consultancy details of patient	Episode care form should be created dynamically with the created input questions and their results	Succeeded
15	Generate option buttons for input results which has one answer	Generated results of a question should have option buttons if the question has one answer.	Succeeded
16	Generate check boxes for input results which has one or more answers	Generated results of a question should have check boxes if the question has one or more answers.	Succeeded
17	Update STD episode care from	Consultancy details of the patient should be updated in the STD episode care form	Succeeded
18	View STD episodes of a patient	All the episodes should be displayed with episode no, start date, end date, status and a link for viewing the history of episode .	Succeeded
19	View consultancy details of episode	Consultancy details of a episode should be displayed	Succeeded

20	Closed episode details could not be updated	Consultancy details of a closed episode should not be updated	Succeeded
21	Cancel episode	Episode should be canceled if needed	Succeeded
22	Canceled episode details should not be displayed	All the episode details should be displayed except canceled ones	Succeeded

Table E.3 Test cases for common functionalities

Test	**Test Description**	**Expected Result**	**Status**
1	Enter empty data for mandatory fields .	Relevant messages should be displayed	Succeeded
2	Display authorized menus	System should display the assigned menu of relevant modules	Succeeded
3	Duplicate records are not allowed to save	Display an error message	Succeeded
4	Logout	Redirect to the login page	Succeeded

User Evaluation

A questionnaire was given to the user for gathering the feedback about the system. The table E.4 shows the questionnaire that was used to get user feedbacks. Tick your choice for the following tasks in the Patients Information Management System.

Table E.4 Questionnaire

Index	Task	(1 –Poor, 2 – Average, 3 – Good, 4 – Very Good, 5 – Excellent)				
1	System login and the security	১	২	৩	৪	৫
2	User interfaces of the system.	১	২	৩	৪	৫
3	Menu structure	১	২	৩	৪	৫
4	Generated messages from the system	১	২	৩	৪	৫
5	Registration page of patients	১	২	৩	৪	৫
6	Search patients information	১	২	৩	৪	৫
7	Search appointment details of patients	১	২	৩	৪	৫
8	Export appointment list	১	২	৩	৪	৫
9	Mark and create visit of patients	১	২	৩	৪	৫
10	Patient category creation page	১	২	৩	৪	৫
11	Search episode details of patients	১	২	৩	৪	৫
12	Episode details page	১	২	৩	৪	৫
13	STD episode care form	১	২	৩	৪	৫
14	Update STD episode care form	১	২	৩	৪	৫
15	Navigate through STD episode care form	১	২	৩	৪	৫
16	System administration functions	১	২	৩	৪	৫
17	Reports are meaning full and accurate	১	২	৩	৪	৫
18	Export reports	১	২	৩	৪	৫

All the users were questioned regarding the user friendliness of the application by giving questionnaire. Certain new suggestions with regard to functionality were expressed. The directors were highly satisfied with the systems achievements and the new management reports that the system was able to produce.

Appendix F - Code Listing

Code Listing

Code segments of Consultancy module

All the functions of Consultancy module are handled by consultancyActions class

```
class consultancyActions extends sfActions {
   /**
    * Save the STD episode details
    * @param sfWebRequest $request
    * @return unknown_type
    */
   public function executeSaveEpisodeDetails(sfWebRequest $request) {

      $stdInput = new StdInput();
      $inputs = $stdInput->getInputAndResults(); //Get all the STD inputs and their results

      $patientNo = $request->getParameter('txtPatientId');
      $episodeNo = $request->getParameter('txtEpisodeId');

      $arrCheckBoxes = array(); // This array keeps all the status of the check boxes

      foreach ($inputs as $input) {
         $resultIndex = "res".$input['input_code'];
         $results = $request->getParameter($resultIndex); //Get the selected STD results when the
user submitted the form
         $inputCode = $input['input_code'];
         $inputCategoryCode = $input['input_category_code'];

         /*
          * Check the type of selected field in the form
          * If the number of inputs for a input is 1, it should be an option button
          * If it is > 1 it should be a check box
          *
         */
         if($input['no_of_input'] == 1) {
            if(!empty($results)) {
               $objStdResult = new StdResult();
               $objStdResult->setPatientNo($patientNo);
               $objStdResult->setEpisodeNo($episodeNo);
               $objStdResult->setInputCode($input['input_code']);
               $tempObjStdResult = $objStdResult-
>getStdEpisodeResultObjectWithoutResultCode();

               //Update an existing option button
               if(count($tempObjStdResult) >0) {
                  $tempObjStdResult[0]->setResultCode($results[0]);
                  $tempObjStdResult[0]->save();
               }else {
                  //Add a new option button
                  $objStdResult->setResultCode($results[0]);
                  $objStdResult->setInputCategoryCode($input['input_category_code']);
```

```
                $objStdResult->save();
            }
        }

    }else {
        $arrTemp = array();
        if(!empty($results)) {
            $arrTemp[0] = $results;
        }else {
            $arrTemp[0]= array();
        }
                $arrTemp[1] = array($inputCategoryCode);
                $arrCheckBoxes[$inputCode] = $arrTemp;
    }
}

//Saving the check boxes
foreach ($arrCheckBoxes as $key=>$chekbox) {
    $inputCode = $key;
    $objStdResultCheckbox = new StdResult();

    if(!empty($chekbox)) {
        //If the user has selected the check boxes
        $resultCheckboxes = $chekbox[0];
        $inputCategoryCode = $chekbox[1][0];

        //Check the exisiting selected check boxes of a input in particular episode
        $arrCheckedInputs = $objStdResultCheckbox-
>getStdEpisodeResulArraytWithoutResultCode($patientNo, $episodeNo, $inputCode);

        if(count($arrCheckedInputs )> 0) {
            //To update an existing record, delet the curerent record
            //Save it as a new record
            $objStdResultCheckbox->deleteStdEpisodeResult($patientNo, $episodeNo,
$inputCode);

        }
        //Save the record
        foreach ($resultCheckboxes as $checkResult) {
            $objStdResult = new StdResult();
            $objStdResult->setPatientNo($patientNo);
            $objStdResult->setEpisodeNo($episodeNo);
            $objStdResult->setInputCode($inputCode);
            $objStdResult->setResultCode($checkResult);
            $objStdResult->setInputCategoryCode($inputCategoryCode);
            $objStdResult->save();
        }

    }else {
        // If the user has not selected any check box for a particular input
```

```
            // All the previous selected items must be removed
            $objStdResultCheckbox->deleteStdEpisodeResult($patientNo, $episodeNo,
$inputCode);
        }

    }
    $this->getUser()->setFlash('notice', 'Record Updated Successfully');
    $this-
>redirect('consultancy/viewEpisodeHistory?mode=view&eid='.$episodeNo."&pid=".$patientNo);
  }

  /**
   * Delete STD input object
   * @param sfWebRequest $request
   * @return unknown_type
   */
  public function executeDeleteStdInput(sfWebRequest $request) {
    if (count($request->getParameter('chkLocID')) >0) {
      $stdInput = new StdInput();
      $stdInput->deleteStdInput($request->getParameter('chkLocID'));
      $this->getUser()->setFlash('notice', 'Successfully Deleted');
    } else {
      $this->getUser()->setFlash('notice', 'Select at Least One Record to Delete');
    }
    $this->redirect('consultancy/showStdInputList');
  }
```

//More functions

```
}
```

Code segments of Registration module

Handling clinic appointments and visit details in Registration module.

```
class registrationActions extends sfActions {
  //Display the current visits of an episode
   public function executeShowCurrentVisit(sfWebRequest $request) {
      $visit = new Visit();
      $patientNo = $request->getParameter('pid');
      $episodeNo = $request->getParameter('eid');
      $visit->setPatientNo($patientNo);
      $visit->setEpisodeNo($episodeNo);
      $visits = $visit->getCurrentEpisodeVisitsOfPatient(); //Get the current visits
      $this->currentVisits = $visits;
   }
   //Mark a clinic visit of an episode
   public function executeMarkVisits(sfWebRequest $request) {
      $patientNo = $request->getParameter('txtPatientNo');
      $episodeNo = $request->getParameter('txtEpisodeNo');
      $visitNo = $request->getParameter('txtVisitNo');
      //Get all the visited dates for validation
      $allVisits = Visit::getAllVisitsOfEpisode($patientNo, $episodeNo);
      if($request->getParameter('txtAppointedDate') != " ") {
         $appointedDate = $request->getParameter('txtAppointedDate');
         $visit = new Visit();
         $visit->setPatientNo($patientNo);
         $visit->setEpisodeNo($episodeNo);
         $visit->setVisitNo($visitNo);
         $currentVisit = $visit->getCurrentEpisodeObject();
          //Validate the appointment date
         if($this->_validateAppointedDate($allVisits, $appointedDate)) {
            $this->getUser()->setFlash('error', 'Appointed date is already visited');
             $this->redirect('registration/showCurrentVisit?pid='.$patientNo."&eid=".$episodeNo);
         }else {
             $currentVisit->setNextVisitDate( $appointedDate);
```

```
            $currentVisit->save();
          }
        //Add a new visit
        $newVisit = new Visit();
        $newVisit->setPatientNo($patientNo);
        $newVisit->setEpisodeNo($episodeNo);
        $newVisit->setVisitNo($visitNo+1);
        $newVisit->setAppointedDate($request->getParameter('txtAppointedDate'));
        $newVisit->save();
      }else {
        //Update the visited dates
        $visit = new Visit();
        $visit->setPatientNo($patientNo);
        $visit->setEpisodeNo($episodeNo);
        $visit->setVisitNo($visitNo);
         //Get the current object for update
        $currentVisit = $visit->getCurrentEpisodeObject();
        $currentVisit->setVisitedDate($request->getParameter('txtVisitedDate'));
         //Validate the visited date
        if($this->_validateVisitedDate($allVisits, $request->getParameter('txtVisitedDate'))) {
            $this->getUser()->setFlash('error', 'Appointed date is already visited');
             $this->redirect('registration/showCurrentVisit?pid='.$patientNo."&eid=".$episodeNo);
        } else {
            $currentVisit->save();
        }
      }
      $this->getUser()->setFlash('notice', 'Record saved successfully');
      $this->redirect('registration/showCurrentVisit?pid='.$patientNo."&eid=".$episodeNo);
    }}
```

Code segments of Report module

All the reports of the system are generated by the `reportActions class`

```
class reportActions extends sfActions {

  //Generate the marital status report of STD patients
   public function executeViewMaritalStatusReport(sfWebRequest $request) {

      $quarter = $request->getParameter('cmbQuarter');
      $date = $request->getParameter('txtDate');
      $year = date("Y", strtotime($date));
      $reportData = $this->_getPatientReportData($request);
      //Sex 1 = Male
      //Sex 2 = Female
     //$maritalStatus keeps the marital status count of each patient
      $maritalStatus['single'] = array(0 => 0, 1 => 0, 2 => 0);
      $maritalStatus['married'] = array(0 => 0, 1 => 0, 2 => 0);
      $maritalStatus['sep'] = array(0 => 0, 1 => 0, 2 => 0);
      $maritalStatus['widowed'] = array(0 => 0, 1 => 0, 2 => 0);
      $maritalStatus['lt'] = array(0 => 0, 1 => 0, 2 => 0);
      $maritalStatus['non'] = array(0 => 0, 1 => 0, 2 => 0);

      foreach ($reportData as $record) {
         if ($record['sex'] == "1") {
            //Male patients
            switch ($record['marital_status']) {

               case 1:
                  $maritalStatus['married'][0] = $maritalStatus['married'][0] + 1;
                  break;

                     case 2:
                  $maritalStatus['single'][0] = $maritalStatus['single'][0] + 1;
                  break;

               case 3:
                  $maritalStatus['sep'][0] = $maritalStatus['sep'][0] + 1;
                  break;

               case 4:
                  $maritalStatus['widowed'][0] = $maritalStatus['widowed'][0] + 1;
                  break;

               case 5:
                  $maritalStatus['lt'][0] = $maritalStatus['lt'][0] + 1;
                  break;
               case 6:
                  $maritalStatus['non'][0] = $maritalStatus['non'][0] + 1;
                  break;

               default:
```

```
                break;
            }
        } else {
            //Female patients
            switch ($record['marital_status']) {
                case 2:
                    $maritalStatus['single'][1] = $maritalStatus['single'][1] + 1;
                    break;

                case 1:
                    $maritalStatus['married'][1] = $maritalStatus['married'][1] + 1;
                    break;

                case 3:
                    $maritalStatus['sep'][1] = $maritalStatus['sep'][1] + 1;
                    break;

                case 4:
                    $maritalStatus['widowed'][1] = $maritalStatus['widowed'][1] + 1;
                    break;

                case 5:
                    $maritalStatus['lt'][1] = $maritalStatus['lt'][1] + 1;
                    break;
                case 6:
                    $maritalStatus['non'][1] = $maritalStatus['non'][1] + 1;
                    break;

                default:
                    break;
            }
        }
    }
    $maritalStatus['single'][2] = $maritalStatus['single'][0] + $maritalStatus['single'][1];
    $maritalStatus['married'][2] = $maritalStatus['married'][0] + $maritalStatus['married'][1];
    $maritalStatus['sep'][2] = $maritalStatus['sep'][0] + $maritalStatus['sep'][1];
    $maritalStatus['widowed'][2] = $maritalStatus['widowed'][0] + $maritalStatus['widowed'][1];
    $maritalStatus['lt'][2] = $maritalStatus['lt'][0] + $maritalStatus['lt'][1];
    $maritalStatus['non'][2] = $maritalStatus['non'][0] + $maritalStatus['non'][1];

    $this->maritalStatuses = $maritalStatus;
    $this->totalMales = $maritalStatus['single'][0] + $maritalStatus['married'][0] +
$maritalStatus['sep'][0] + $maritalStatus['widowed'][0] + $maritalStatus['lt'][0] +
$maritalStatus['non'][0];
    $this->totalFemales = $maritalStatus['single'][1] + $maritalStatus['married'][1] +
$maritalStatus['sep'][1] + $maritalStatus['widowed'][1] + $maritalStatus['lt'][1] +
$maritalStatus['non'][1];
    $this->total = $this->totalMales + $this->totalFemales;        $this->year = $year;
    $this->quarter = $quarter;
}//More functions}
```

Appendix G - Client Certificate

දුරකථන අංක:
தொலைபேசி இல
Tel. Nos.

අධ්‍යක්ෂ
இயக்குனர்
Director 2692456

කාර්යාලය
காரியாலயம்
Office 2666342
2696533
2662970

ෆැක්ස්
பெக்ஸ்
Fax 5337368

E-mail: **dir@eureka.com**

මගේඅංකය.
எமது இல.
My No.

ලිංගාශ්‍රිත රෝග/ඒඩ්ස් මර්දන වැඩසටහන
பாலுறவுநோய்/எய்ட்ஸ் தடுப்பு வேலைத்திட்டம்
STD/AIDS CONTROL PROGRAMME

ද සේරම් පෙදෙස/டி சேறம் பிளேஸ் /De Saram Place
තැ.පෙ./த.பெ.இல./P.O. Box 567
කොළඹ 10 (ශ්‍රී ලංකාව)
கொழும்பு 10 (இலங்கை)
Colombo 10 (Sri Lanka)

ඔබේ අංකය:
உமது இல.
Your No.

දිනය:
திகதி
Date: 02.08.2010

Project Examination Board,

University of Colombo School of Computing,

Colombo 07.

Dear Sir,

Client Certification

Mr. W.V.D.D.S. Warakagoda has successfully completed the Patient Information Management System for our institution as his project work for the Bachelor of Information Technology (BIT) external degree program collaborated by University Of Colombo. I wish to declare that the software which is developed by her is suitable for day to day operation of our institution. I wish him success.

DIRECTOR
NATIONAL STD/AIDS CONTROL PROGRAMME

DIRECTOR
STD/AIDS CONTROL PROGRAMME
P. O. BOX 567, COLOMBO 10.