



**Web Based Cardiothoracic Anaesthesia
and Critical Care Unit Management
System
for
Lady Ridgeway Hospital**

D. M. U. I. Dissanayake

BIT registration number – R141193

Index Number – 1411934

Supervisor – Dr. Viraj Pinto Jayawardena

Academic Year – 2017 November



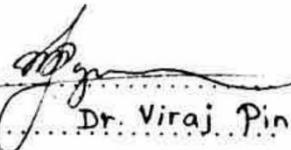
**This dissertation is submitted in partial fulfilment of the requirement of the
Degree of Bachelor of Information Technology (external) of the
University of Colombo School of Computing**

Declaration

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.

Signature of Candidate: Udan Date: 2017/11/06
Name of Candidate: D. M. U. I. Dissanayake

Signature of Supervisor(s)/Advisor(s):  Date: 2017/11/06
Name(s) of Supervisor(s)/Advisor(s): Dr. Viraj Pinto Jayawardena

Abstract

Lady Ridgeway Hospital, Colombo is now known as the largest public hospital for children in the world. It provides treatments for children under 12 years old and serves as the national pediatric care center for the whole country and as a local hospital for children in Colombo. LRH provides a wide range of 24 hours' services for completely free of charge. The hospital is equipped with well-trained and qualified staff and excellent facilities. Cardiothoracic Anaesthesia and Critical Care unit of LRH is a specialized unit which provides critical and surgical care for children struggling with heart diseases.

Currently, the unit is using several applications such as spreadsheets and word documents for capturing, storing, and processing information. However, mainly information handling is done manually. The current system is quite complex, and difficult to manage, due to the use of papers and spreadsheets. Data retrieval is time-consuming and needs a lot of effort to track the patient history.

The main area where an Information Technology (IT) solution was needed is for handling follow-up records of inpatient patients and retrieving discharged patient history. The system facilitates computer-based storage and quick retrieval of data of inpatient and discharged patients. Also, the system can track particular patient information and schedule of the relevant patient's surgery plan. The system will provide shift allocations for medical officers, which is currently done using a manual system. The solution will address problems such as unfairness and invisibility of the process, which can be raised when shift allocation is done manually. The solution will facilitate unit administration to analyze information of the patients by the reports generated through the system, such as Mortality reports, Discharge summaries. Intensive care provided by the unit will be enhanced by linking information and keeping track of them.

The system is web-based and the server used is Apache. It is based on a client-server architecture and developed in accordance with object-oriented principles; therefore, it can be easily extended in the future. The system is developed using languages such as PHP, HTML, and JavaScript. Unified Modeling Language (UML) is used for modeling. The database used is a MySQL database. The Bootstrap framework has been used, therefore, the system is accessible through several devices such as Laptops, Tablets, and Mobile phones.

Acknowledgement

First of all, I would like to express my special thanks of gratitude to BIT coordinator, academic staff of University of Colombo School of Computing (UCSC) and Project Examination Board (PEB) for their effort and providing us a great opportunity to follow a valuable degree program and gain the practical knowledge through this project.

A very special gratitude to my project supervisor Dr. Viraj Pinto Jayawardena. This project would not be possible without kind support and guidance given by him. And I'm really grateful to the lecturer panel at Matrix Institute of Information and Technology for the knowledge and wisdom given throughout these years.

I would like to offer my deep gratitude to Mr. Sarath Wijesinghe for the patient guidance, support and encouragement he has provided to make this a success.

I wish to express my sincere gratitude to Dr. L. A. P. Perera who requested the system, her support and cooperation despite of her busy schedule cannot be forgotten. Her interest in introducing an IT solution for the unit was truly a motivation to carry out the project. Special thanks goes to Dr. M. T. D. Munasinghe for the support offered by him in requirement gathering, to gain background knowledge and for much patience in clarifying the functionalities and procedures in the unit.

I kindly recall my family who motivated me and always believed in me. Special thanks go to my sister who encouraged me to do BIT and for all the guidance she has provided.

Finally, I would like to thank everyone who helped me from the beginning by giving me all the support and assistance.

Contents

Declaration	ii
Abstract	iii
Acknowledgement	iv
Contents	v
List of Figures	ix
List of Tables	xi
List of Acronyms	xii
Chapter 1 – Introduction	1
1.1 Introduction to the Cardiothoracic Anaesthesia and Critical Care Unit of Lady Ridgeway Hospital Colombo	1
1.2 Motivation for the Project	1
1.3 Objectives of the project	2
1.4 Scope of the project.....	2
1.5 Dissertation Overview	2
1.5.1 Chapter 2 - Analysis.....	3
1.5.2 Chapter 3 - Design	3
1.5.3 Chapter 4 - Implementation	3
1.5.4 Chapter 5 - Evaluation	3
1.5.5 Chapter 6 - Conclusion	3
1.5.6 References.....	3
1.5.7 Appendices.....	4
1.5.8 Glossary and Index	4
Chapter 2 – Analysis	5
2.1 Initial structure and the work flow of the existing system	5
2.2 Some downsides of the existing system.....	6

2.3	High level Use case diagram for the current system	7
2.4	Similar Systems	7
2.4.1	Simple EMR.....	7
2.4.2	Hospital Health Information Management System (HHIMS)	8
2.5	Similar systems and proposed system features comparison.....	9
2.6	Requirement gathering techniques	10
2.6.1	Interviews.....	10
2.6.2	On site observations	10
2.6.3	Forms and Reports from the Unit	10
2.6.4	Questionnaires.....	11
2.6.5	Research	12
2.7	Requirement analysis	12
2.7.1	Functional requirements.....	12
2.7.2	Nonfunctional requirements.....	14
2.7.3	Relationship between Functional and Nonfunctional requirements	15
Chapter 3 – Design.....		16
3.1	Alternative solutions	16
3.2	Methodology	16
3.2.1	RUP with UML.....	17
3.3	Use case diagram for the proposed system	18
3.4	Sequence Diagrams	19
3.5	Entity Relationship Diagram (ERD)	20
3.6	Database Normalization	20
3.7	Database Design	21
3.8	Class Diagram	22
3.9	Interface design	23
3.9.1	Login Page	23

3.9.2	Dashboard	24
3.9.3	User Management	24
3.9.4	Roles Management.....	25
3.9.5	New Patient Registration form.....	26
3.9.6	Patient Management.....	27
3.9.7	Patient Administration	30
3.9.8	Medication Management	31
3.9.9	Utility Management	32
3.9.10	Roster Management for Medical officers	33
3.9.11	Report Module	34
Chapter 4 – Implementation.....		35
4.1	Introduction to Implementation.....	35
4.2	Implementation Environment.....	35
4.2.1	Software Environment	35
4.2.2	Hardware Environment.....	36
4.3	Actual Implementation.....	37
4.3.1	Database server	38
4.3.2	Web server	38
4.4	File structure of the developed system.....	38
4.5	High-level Module structure	39
4.6	Mapping between Modules and Functional Requirements	39
4.7	Major code segments.....	40
4.7	Re-usable components.....	43
Chapter 5 – Evaluation.....		44
5.1	Introduction	44
5.2	System Testing	44
5.3	Test Plan.....	44

5.4	System Test Cases	45
5.5	Test Data and Test results	47
5.6	User Acceptance Test.....	47
Chapter 6 – Conclusion.....		50
6.1	Introduction	50
6.2	Overview of developed system	50
6.3	Lessons learnt.....	51
6.4	Future Enhancements	51
References.....		52
Appendix A - System Documentation		53
Appendix B – Design Documentation		55
Appendix C – User Documentation.....		62
Appendix D - Management Reports		72
Appendix E – Test Results.....		74
Appendix F - Code Listing		82
Appendix G - Client Certificate.....		94
Glossary		95
Index		96

List of Figures

FIGURE 2.1 HIGH LEVEL USE CASE DIAGRAM FOR THE CURRENT SYSTEM.....	7
FIGURE 2.2 INTERFACE OF SIMPLE EMR.....	8
FIGURE 2.3 INTERFACE OF HHIMS	9
FIGURE 2.4 PRESCRIPTION CHART OF THE UNIT.....	11
FIGURE 3.1 HIGH LEVEL USE CASE DIAGRAM FOR THE PROPOSED SYSTEM.....	18
FIGURE 3.2 SEQUENCE DIAGRAM FOR THE DISCHARGE SUMMERY GENERATION.....	19
FIGURE 3.3 ER DIAGRAM FOR THE SYSTEM.....	20
FIGURE 3.4 DATABASE DESIGN.....	21
FIGURE 3.5 CLASS DIAGRAM FOR THE SYSTEM.....	22
FIGURE 3.6 SYSTEM LOGIN PAGE	23
FIGURE 3.7 DASHBOARD.....	24
FIGURE 3.8 USER MANAGEMENT PAGE	24
FIGURE 3.9 NEW USER REGISTRATION FORM.....	25
FIGURE 3.10 ROLES MANAGEMENT PAGE.....	25
FIGURE 3.11 PATIENT REGISTRATION FORM.....	26
FIGURE 3.12 NAVIGATION PANEL OF THE PATIENT MANAGEMENT MODULE.....	27
FIGURE 3.13 ADMISSION REQUESTS TO THE UNIT.....	27
FIGURE 3.14 DETAILS OF PATIENTS WHO ARE ADMITTED TO THE ICU	28
FIGURE 3.15 DETAILS OF SURGERIES IN A SPECIFIC OT	29
FIGURE 3.16 INTERFACE OF THE PATIENT ADMINISTRATION	30
FIGURE 3.17 MEDICATION MANAGEMENT PAGE	31
FIGURE 3.18 INVENTORY CONTROL PAGE.....	31
FIGURE 3.19 NEW MEDICATION REGISTRATION FORM.....	32
FIGURE 3.20 MAIN NAVIGATION OF THE UTILITY MANAGEMENT MODULE.....	32
FIGURE 3.21 SCHEDULE PLAN	33
FIGURE 3.22 REPORT GENERATION PAGE	34
FIGURE 4.1 MVC ARCHITECTURE REPRESENTATION	37
FIGURE 4.2 FILE STRUCTURE OF THE DEVELOPED SYSTEM	38
FIGURE 5.1 USER ACCEPTANCE TESTING EVALUATION FORM.....	48
FIGURE 5.2 USER ACCEPTANCE TEST FEEDBACK FORM.....	49
FIGURE 5.3 USER ACCEPTANCE TEST RESULTS	49

FIGURE B.1 USE CASE DIAGRAM – USER MANAGEMENT.....	55
FIGURE B.2 USE CASE DIAGRAM – PATIENT MANAGEMENT.....	56
FIGURE B.3 USE CASE DIAGRAM – MEDICATION MANAGEMENT.....	57
FIGURE B.4 USE CASE DIAGRAM – REPORT GENERATION MODULE.....	58
FIGURE B.5 ACTIVITY DIAGRAM – LOGIN	59
FIGURE B.6 ACTIVITY DIAGRAM – CREATE SCHEDULE PLAN.....	60
FIGURE B.7 SEQUENCE DIAGRAM – PATIENT REGISTRATION	61
FIGURE C.1 LOGIN PAGE.....	62
FIGURE C.2 DASHBOARD	63
FIGURE C.3 NAVIGATION BAR	63
FIGURE C.4 SIDE MENU	63
FIGURE C.5 USER MANAGEMENT SCREEN	64
FIGURE C.6 USER REGISTRATION FORM	64
FIGURE C.7 USER-ROLE MAINTENANCE SCREEN	65
FIGURE C.8 ASSIGN ROLES FORM.....	65
FIGURE C.9 PATIENT REGISTRATION FORM.....	66
FIGURE C.10 NAVIGATION PANEL THROUGH SECTIONS.....	66
FIGURE C.11 ADMISSION REQUESTS	67
FIGURE C.12 ADMISSION TO ICU	67
FIGURE C.13 ADMISSION TO OT	67
FIGURE C.14 PATIENT DETAILS INSIDE AN ICU.....	68
FIGURE C.15 FULL PROFILE OF THE PATIENT	68
FIGURE C.16 DETAILS INSIDE A SPECIFIC OT	69
FIGURE C.17 PATIENT ADMINISTRATION SCREEN	69
FIGURE C.18 MEDICATION MANAGEMENT SCREEN	70
FIGURE C.19 INVENTORY OF MEDICATION	70
FIGURE C.20 UTILITY MANAGEMENT ITEM NAVIGATION.....	70
FIGURE C.21 SCHEDULE GENERATION FORM	71
FIGURE C.22 INVENTORY OF MEDICATION	71
FIGURE D.1 ALL ADMISSIONS TO THE DIFFERENT SECTIONS.....	72
FIGURE D.2 ALL ADMISSIONS TO THE DIFFERENT SECTIONS WITHIN A GIVEN TIME PERIOD	72
FIGURE D.3 ALL ADMISSIONS TO A SPECIFIC SECTION WITHIN A GIVEN TIME PERIOD.....	73

List of Tables

TABLE 2.1 SIMILAR SYSTEM AND PROPOSED SYSTEM FEATURES COMPARISON.....	9
TABLE 2.2 DETAILS OF THE INTERVIEW WITH A STAFF MEMBER	10
TABLE 2.3 RELATIONSHIP BETWEEN FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS	15
TABLE 4.1 MAPPING BETWEEN MODULES AND FUNCTIONAL REQUIREMENTS	39
TABLE 5.1 SYSTEM TEST CASES	46
TABLE B.1 USE CASE DESCRIPTION – USER MANAGEMENT	55
TABLE B.2 USE CASE DESCRIPTION – PATIENT MANAGEMENT	56
TABLE B.3 USE CASE DESCRIPTION – MEDICATION MANAGEMENT.....	57
TABLE B.4 USE CASE DESCRIPTION – REPORT GENERATION.....	58
TABLE E.1 USER AUTHENTICATION TEST RESULTS	74
TABLE E.2 USER MANAGEMENT TEST RESULTS	76
TABLE E.3 PATIENT MANAGEMENT TEST RESULTS	78
TABLE E.4 PATIENT ADMINISTRATION TEST RESULTS	79
TABLE E.5 MEDICATION MANAGEMENT TEST RESULTS.....	80
TABLE E.6 SCHEDULE PLAN TEST RESULTS	81

List of Acronyms

BHT- Bed Head Ticket

BIT – Bachelor of Information and Technology

CATH Lab - Cardiac Catheterization Laboratory

CBS – Comfort Behavior Scale

EMR – Electronic Medical Records

ERD - Entity Relationship Diagram

FR – Functional Requirement

ICU- Intensive Care Unit

IT – Information Technology

LRH – Lady Ridgeway Hospital

NFR – Nonfunctional Requirement

OOAD – Object Oriented Analysis and Design

OT- Operation Theater

RAD – Rapid Application Development

RUP – Rational Unified Process

UML – Unified Modeling Language

Chapter 1 – Introduction

1.1 Introduction to the Cardiothoracic Anaesthesia and Critical Care Unit of Lady Ridgeway Hospital Colombo

Cardiothoracic Anaesthesia and Critical Care Unit of LRH is a specialized and advanced unit which provides intensive and surgical care for children with heart diseases. Most of the patients treated or operated here are newborns whom usually diagnosed with congenital heart disease, which is considered as a defect in the heart.

The treatments provided by this unit is difficult, exceptional and highly demanding which is performed by highly skilled and experienced team of Cardiologists, Anaesthesiologists and Cardiac surgeons. Some of these professionals have international experience in their particular specialty.

The remarkable service they provide with excellent results can even be comparable to the best medical services in the world. This is the only dedicated center which provides timely and proper treatments for absolutely free which gives a hope of life to the children with particular heart diseases.

1.2 Motivation for the Project

Final year project is an important and key element of the BIT degree program of University of Colombo School of Computing which gives us the opportunity to deepen our practical knowledge and learn how to use our skills.

Cardiothoracic Anaesthesia and Critical Care unit of LRH, Colombo was interested in introducing a computer based unit management system. Client was looking for a solution which can electronically store and process their patient information, surgery details as well as roster management of Medical officers. IT solution was required for shift allocation (roster) of Medical officers.

Huge support was provided by the in-charge doctor and the staff of the unit. Permission was given to visit and observe the work flow of the unit several times.

New system will provide efficiency and accuracy in patient history, surgery details and follow-up records. Better assistance with the generation of shift rotation schedules will be provided. System will generate useful reports to analyze information which will be helpful to the unit administration.

1.3 Objectives of the project

- Replacing the current system with an automated system.
- Reduce work load and improve efficiency and productivity of the users.
- Keep track of all the activities collaboratively.
- Patient history management.
- Provide reports for administrative and research purpose.
- Quality maintenance and reduce errors in the records.
- Provide user friendly system to the unit.
- Enhance privacy and security of the medical records.
- Practically use the knowledge and skills gained throughout this degree program.

1.4 Scope of the project

- Keep electronic medical records.
- Manage Admission requests, surgery plans staff information and roster of the Medical officers.
- Manage all the medical records and follow-up records of the inward patients.
- Keep the records of discharged patients.
- Generate Mortality reports and Discharge summaries.
- User input validation and provide views to update information of the unit.
- Provide graphs to show drug dosage over time.
- Users are provided with usernames and passwords to access the system.

1.5 Dissertation Overview

The main six chapters of the Dissertation including the Introduction chapter describes the outline flow of the software development. This provides the better understanding of the system.

1.5.1 Chapter 2 - Analysis

This chapter includes the description about process of requirement identification and finalizing, requirement that have to be obtained in the new system such as user requirements, functional and nonfunctional requirements, techniques used for requirement gathering.

1.5.2 Chapter 3 - Design

Design chapter includes the Database design, Interface and System design. Use case diagrams, ER diagrams are used to further understanding of the system. Designing wireframe of the user interfaces are described up to some extent.

1.5.3 Chapter 4 - Implementation

Describes the techniques of the implementation process. Coding standards, Tools which are used and the reason to choose them are explained with some examples.

1.5.4 Chapter 5 - Evaluation

Describe how the testing is done using some of actual data and dummy data. Test cases, Test reports and Error handling techniques are discussed with the provided samples.

1.5.5 Chapter 6 - Conclusion

This chapter summarize the final outcome of the system, how can future extend the system and achievements gained throughout the project.

1.5.6 References

Details of the references and acknowledgement of the any work of others that adapted to the system.

1.5.7 Appendices

Appendix A - System Documentation – Technical Documentations which provides configurations, illustrations and hardware and software requirements are included.

Appendix B - Design Documentation – Entity relationship, use case diagrams and UML diagrams used in design stage are included here

Appendix C - User Documentation – Comprehensive documentation which describes how to use the system is included.

Appendix D - Management Reports – All reports including administrative reports with samples.

Appendix E – Test Results – Actual test results which were used to validate system are included.

Appendix F - Code Listing – Codes which were used in development process of the system.

Appendix G - Client Certificate – Client Certificate which indicates the client requirement are met and provided a satisfying system.

1.5.8 Glossary and Index

Definition of the technical terms used is with an extensive index is included.

Chapter 2 – Analysis

Requirements analysis, is also called requirements engineering, is the process of determining user expectations for a new or modified product. [1] Understanding the nature of the problem and way to fix it, recognizing background of the physical environment has to be done in this phase. Main objectives of the analysis phase are identifying the project goal and scope and finalizing the functional and nonfunctional requirements. Result of this phase builds up a common understanding between the end user and the system analyst. Therefore this phase plays a major role in system development as incomplete or incorrect specification may lead to a bad software product.

2.1 Initial structure and the work flow of the existing system

Structure of the unit – ICU 1, ICU 2 and four Operation theaters such as OT- A, OT- B, OT- C, CATH Lab

Types of admission to the unit

- Direct admission to ICU
- Admission to an Operation theater for surgery
- Admission to ICU for post-operative care (After a surgery)

Admission to the unit is always done from Cardiology ward and cardiothoracic surgical ward (Ward 15 and Ward 20 respectively). Prior treatments, diagnoses and tests are done at the wards. Paper work including patient's personal details, admission details to the ward, medical history and identified diagnoses, intervention needed, is sent to the unit when admission is made.

Work flow of the unit

ICU - Advanced treatments to the critical patients, observations, medication prescriptions, cardiac and respiratory support and monitoring in post-operative period are done. Patient is assigned to a bed and carrying out investigations and maintaining follow-up records are done in ICU for all inward patients. Follow-up record update of every patient in each shift is given to the in-charge Medical officer of the next shift. (Shift hand-over chart)

OT - CATH Lab is allocated for cardiac catheterization. Other surgical procedures (surgeries) including open heart surgeries are done in other Operation theaters. Surgical plan for a relevant patient is made by the cardiothoracic surgical ward and sent to the unit.

After the surgical procedures and treatments are done, patient is provided a discharge summary including relevant information during unit stay and discharged status, then sent back to the Cardiology or Cardiothoracic surgical ward.

Staff – Consultants and Medical officers, Nurses, In-charge doctors of the ward

Shifts

- Three shifts for ICU - Morning (7am-1pm), Evening (1pm-7pm) and Night (7pm-7am).
- Two shifts for OT – Morning, Evening.

2.2 Some downsides of the existing system

Some of identified downsides of the existing system while requirement analysis is listed below.

- Possibility of having redundant data was high.
- Patient history maintenance was low.
- Data retrieval takes lot of effort and it was time consuming.
- Maintaining surgery details, information of the patient from the ward is mainly done through paper work. Therefore, it was really hard and inefficient to maintain the collaboration, transportation of the documents.
- Report generation needed for unit administration was complicated.
- Manually creating roster for Medical officers was really difficult and caused invisibility of the process, unfairness.

2.3 High level Use case diagram for the current system

The basic work flow of the current system is shown in Figure 2.3.1

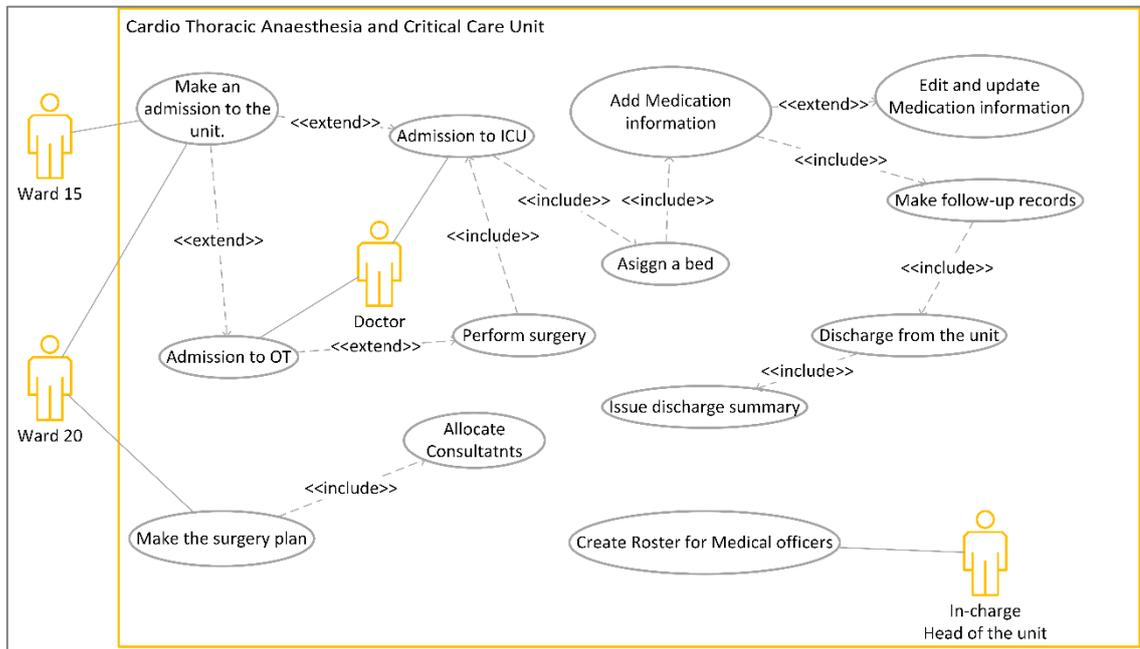


Figure 2.1 High level Use case diagram for the current system

2.4 Similar Systems

It was decided to study some similar systems and their features to further understand and learn the proposed system. Proposed system is designed to manage the information of a specialized unit of the Government Hospital. So the access to the information should only be granted to the Medical staff of the unit and Admin.

2.4.1 Simple EMR

Simple EMR is an easy to use, multi user, low cost, off-the-shelf Electronic Medical Record (EMR) / Medical Practice Management software solution for practitioners. It is also chosen by a few universities to train their medical students due its simplicity and low cost. It is, as the name implies, a simple medical software, but not in the sense that it lacks functionality and good information structure, but rather in terms of workflow simplicity. [2]

Some functionalities:

- Ability to generate, edit, format and print or prescriptions, encounter records.
- Ability to search patients by almost all types of information stored in the medical records.
- Ability to backup and restore the medical records database to prevent data loss.

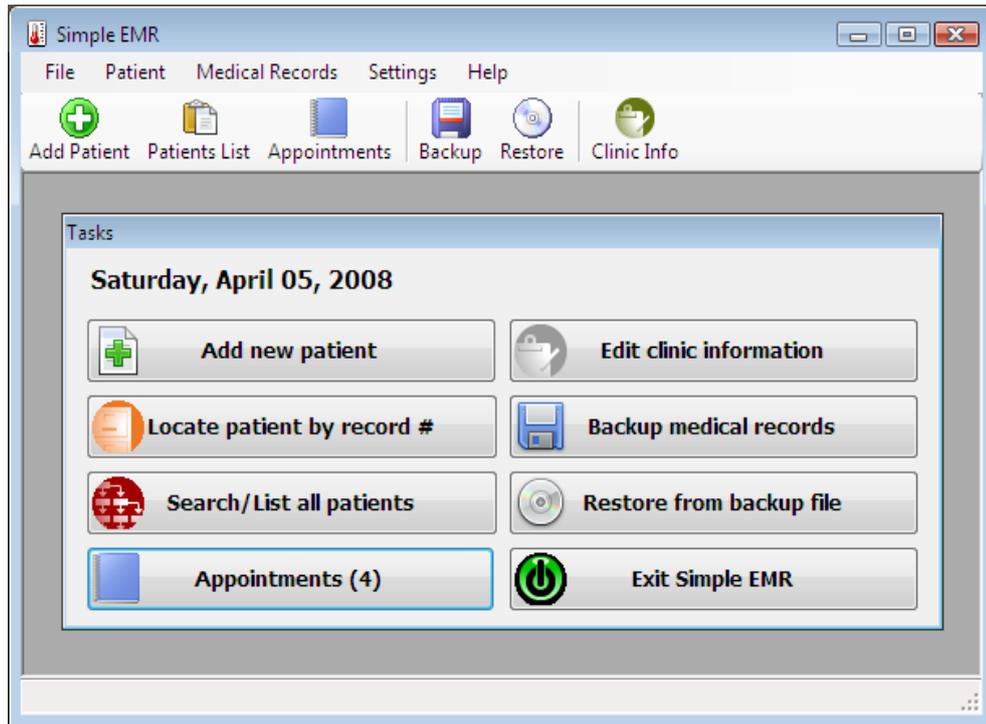


Figure 2.2 Interface of Simple EMR

2.4.2 Hospital Health Information Management System (HHIMS)

HHIMS is open-source medical record software developed for use in Sri Lankan hospitals. It stores the patient's clinical details during out-patient visits, clinic consultations or when admitted to the wards and is designed to replace paper records. Medical details can be entered directly into the database as the patient is examined or shortly afterwards. Lab-tests, prescriptions and treatments can be ordered through the computer network and carried out without the need for paper records. One single screen can display an overview of all the patient's clinical details when they return for a further visit or when they are admitted. [3]

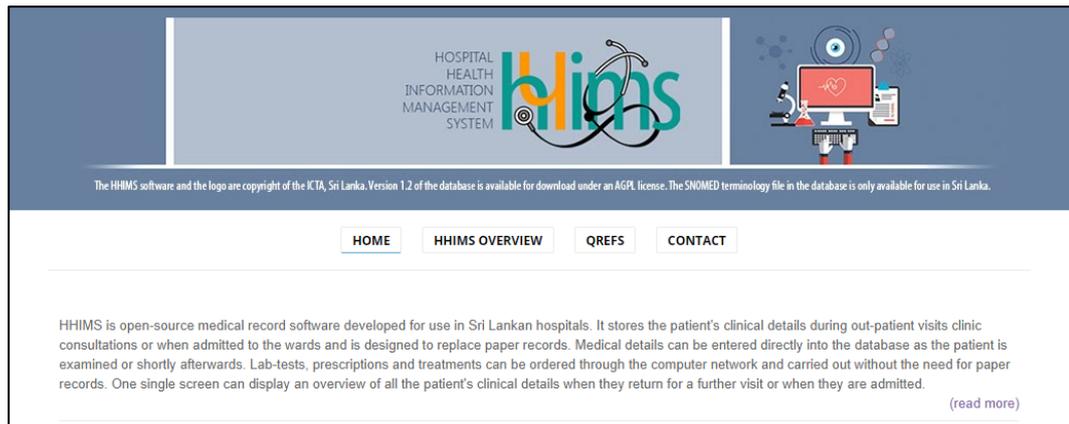


Figure 2.3 Interface of HHIMS

Some functionalities:

- Patient registration, search and overview.
- Examination and drug prescriptions.
- Clinic management, consultation and Print clinic book.

2.5 Similar systems and proposed system features comparison

Features	Simple EMR	HHIMS	Proposed System
Electronic patient and staff records management	✓	✓	✓
Lab reports management	✓	✓	✓
Manage surgery plans	✗	✗	✓
Follow-up record management	✓	✓	✓
Issue discharge summaries	✓	✓	✓
Mortality report generation	✗	✓	✓
Drug dosage graphs generation	✓	✓	✓
Medical officers' roster generation	✗	✗	✓
Patient appointments managements (Not a requirement of the proposed system)	✓	✓	✗

Table 2.1 similar system and proposed system features comparison

2.6 Requirement gathering techniques

The quality, correctness and the productivity of the system is mainly based on how well the requirements are gathered. Therefore it is important to choose the correct and useful fact gathering techniques in identifying the requirements. And to get the better and clear understanding of the system and the requirements.

2.6.1 Interviews

This is considered as a very common fact gathering technique. This system is introduced to a unique and specialized unit of the hospital. Doctors inside the unit and the wards which made the admissions to the unit were the focused end users. Therefore this is the main method used for fact gathering. Interviewing the client was done for identifying and finalizing the requirements. Interviews with the Medical officers were helpful to understand the processes inside the unit and gain the background knowledge.

Date of the Interview	05/28/2017
Duration	50 minutes
Interviewee	Dr. M. T. D. Munasinghe (Medical officer)
Purpose	Learn and understand the structure and work flow of the current system

Table 2.2 Details of the Interview with a staff member

2.6.2 On site observations

Observation is a fact gathering technique where system analyst watch how people carry out tasks and how the activities get done through site visits. This was helpful to clarify some requirements whenever the information provided by the client and Medical officers were too complex or hard to express.

2.6.3 Forms and Reports from the Unit

The existing system was mainly paper based. Knowledge gained through the forms and reports they have been used was helpful to identify some of the facts that were not possible to clarify after consulting users. This technique was useful to overcome conflicts and to provide a system which is familiar to the users.

2.6.5 Research

It was decided to do some researches to gather background knowledge and to study the operations of similar systems in the Internet. It was helpful to decide the feasibility and limitations of the system.

2.7 Requirement analysis

2.7.3 Functional requirements

FR-1. Manage Admission requests to the Unit and patient information.

This function helps to manage the Admission requests to the Unit with the personal information, admission details and medical history of the patient. And it will provide facility to insert medication information after patient is admitted to the Unit.

- New patient registration. (Admission request)
- View and edit information of patients.

FR-2. Manage staff (Medical officers, Consultants) information.

System will maintain the personal information, contact details and Rosters of the staff. This function will be helpful to find the in charge Medical officer(s) of specific time and a section (ICU 1, ICU 2 etc.) of the Unit.

- New staff member registration.
- Edit and view information of staff members by their designation (category) and registration number.
- Send email to the user about account details

FR-3. Manage surgery plans.

System will handle the information of daily surgery plans. And will keep track with the relevant patient information and schedule of the particular surgery plan. Status indications of a surgery, such as scheduled/ongoing will be given by the system.

- Enter a new surgery plan.
- Allocate Consultants (Surgeon, Anaesthesiologist) to a surgery plan.

- Make changes to the surgery plan and change status of the surgery.

FR-4. Keep track with the daily follow-up records.

System has the ability to provide suitable forms to gather follow-up records (Changes made to a patient's medication or any other taken actions) and it will be used to create the Shift hand over chart (Record of a patient made by the Doctor at the end of each shift) with patient's general information. System will always indicate the status (Stable/ Unstable/ Critical).

- Enter follow-up records and medication information for a patient
- Make changes to follow-up records and change the status of the patient
- View follow-up with submitted time and user.

FR-5. Analyze discharged patients' information.

System will keep the history of the discharged patients with their all information during Unit stay, including their diagnosis information, Follow-up records and Discharge status etc. Which will be helpful when generating Mortality Reports. And it will allow Unit administration to easily find information of any discharged patient from the Unit.

- Ability to change the filtering criteria. (Clinic number, ICU Admission number etc.)
- Filter out the patients with the discharge status, section etc.

FR-6. Issue Discharge summary.

At the time of discharging a patient, System will provide a suitable record of the patient's general information and other selected information of the patient. This will be helpful when issuing the Discharge Summary with the updated information by the Doctor at the time of discharge.

- Get a record when discharging.
- Change the discharge status, make changes to the information of the discharge summary.

FR-7. Provide views which used to update Unit information.

System will provide views for the Unit administration to update the Unit information such as new surgeries, new Diagnoses, New Drugs or Antibiotics etc.

FR-8. Mortality Reports Generation.

System will generate Mortality Reports when needed, using Information of the Discharge summaries. This function will provide the percentage of Mortality of a specific time period. Can compare the Mortality information between months or years.

FR-9. Generate Roster for Medical officers.

System can generate the monthly Roster for all the Medical officers using details such as number of shifts at a single section of the unit should covered by a Medical officer. Unit administration can make changes to the Roster if needed. System will send the roster to the all Medical officers via an email.

FR-10. Generate Drug usage graphs over time.

System will generate graphs for a given drug usage over the time in a specific section of the unit.

2.7.2 Nonfunctional requirements

NFR-1. Security

No unauthorized access to the system should be guaranteed. The records in the system are really critical and need not be modified by an unauthorized party. Logging will be done using a user name and a password. Users are added by the Admin and they will be provided with the logging details. An email will be sent to the specific user with logging details.

NFR-2. Reliability

Based on patient information, medication information in the system, further treatments to the patient and the decisions about the patient are taken. Therefore 100% accuracy is needed. Precise information which will be needed for administration and research purpose will be given by the reports.

NFR-3. Usability

The target users of the system are familiar with paper based system and some other basic computer applications. Therefore system should be easy to use even for a person who does not have much technical knowledge. Medical terms will be used in the system so that it will become familiar to the users.

NFR-4. Availability

System should always be available. This system holds critical information about patients. The records could be needed any time in the day. The system should be operational 24/7.

NFR-5. Performance

System should be fast and should complete the task in minimum amount of time. Data should be retrieved as soon as possible and will have a small response time.

NFR-6. Extendibility

System should be made open to be extended. In case of the unit is expanded and added new modules in future, system should be easily modified and possible to add new functionalities without impacting the existing system functionalities.

2.7.3 Relationship between Functional and Nonfunctional requirements

Functional Requirements	Nonfunctional Requirements
FR-1	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-2	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-3	NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-4	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-5	NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-6	NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-7	NFR-3, NFR-4, NFR-5, NFR-6
FR-8	NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-9	NFR-1, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6
FR-10	NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

Table 2.3 Relationship between Functional and Nonfunctional requirements

Chapter 3 – Design

Planning and designing a solution to the problem specified by the requirement analysis is done in Design phase. This phase is a critical phase as the quality of the final outcome highly depends on this phase.

3.1 Alternative solutions

One of the main alternative solution would be developing a stand-alone system. Stand-alone systems are very costly and installation and maintenance is harder compared to a web based application. User is restricted to use the application only on the machine it has been installed and storage capacity is restricted.

Reasons for using the web based solution

The unit had access to the Internet. One of the main requirements client had was enabling rapid access to the records anytime. Web based applications allow user to access the system from anywhere, anytime and it also enables wide range of devices to be used to access the system.

3.2 Methodology

There are several well-known methodologies such as Agile, Waterfall, Rational Unified Process (RUP), Scrum and Rapid Application Development (RAD).

It was decided to use RUP for this project. Waterfall method is the traditional method and done in to a sequential manner whereas RUP is an iterative method which supports the structure of the project. Concept of visual diagrams is used in RUP, while really heavy documentation is used in Waterfall. The requirements of the project were specific and well defined. Therefore agile method was not suited as it supports vague requirements instead of the well-defined scope and it is recommended for quick enhancements of a short term project. It is more suitable for systems with rapidly changing requirements, which does not depend on a deadline. The only formal artifact of the agile methodology is operational software. RUP is supported with rich documentation and Object Oriented Analysis and Design (OOAD) with the use of UML diagrams.

3.2.1 RUP with UML

The RUP has determined a project life-cycle consisting of four phases. These phases allow the process to be presented at a high level in a similar way to how a 'waterfall'-styled project might be presented, although in essence the key to the process lies in the iterations of development that lie within all of the phases. Also, each phase has one key objective and milestone at the end that denotes the objective being accomplished. [4] Inception, Elaboration, Construction and Transition are the four phases of RUP.

UML diagrams are used to visualize the specifying and constructing artifacts of the system. This graphical language provided with wide range of different kind of diagrams and notations to design every aspect of the system clearly.

Two types of diagrams are used in UML: Structure Diagrams and Behavior Diagrams. Behavior Diagrams represent the processes proceeding in a modeled environment. Structure Diagrams represent the elements that compose the system. [5] Some common diagrams used are Use case diagrams, Class diagrams, Sequence diagrams and Activity diagrams. Entity relationship diagrams are used to show the table structure of the database.

Reasons for using RUP

- RUP supports object oriented development.
- Facilitates the Unified Modeling Language (UML).
- Proposed system has specific requirements.
- Can adapt new ideas to the project iteratively.
- Allow to visualize the problem and the object interactions using various diagrams which shows the direct mapping of the real world objects to the system entities.
- Applications are easier to maintain therefore the development time is reduced.

3.3 Use case diagram for the proposed system

The interactions by an actor with the system and the functionalities specified for each actor can be visualized using a Use case diagrams.

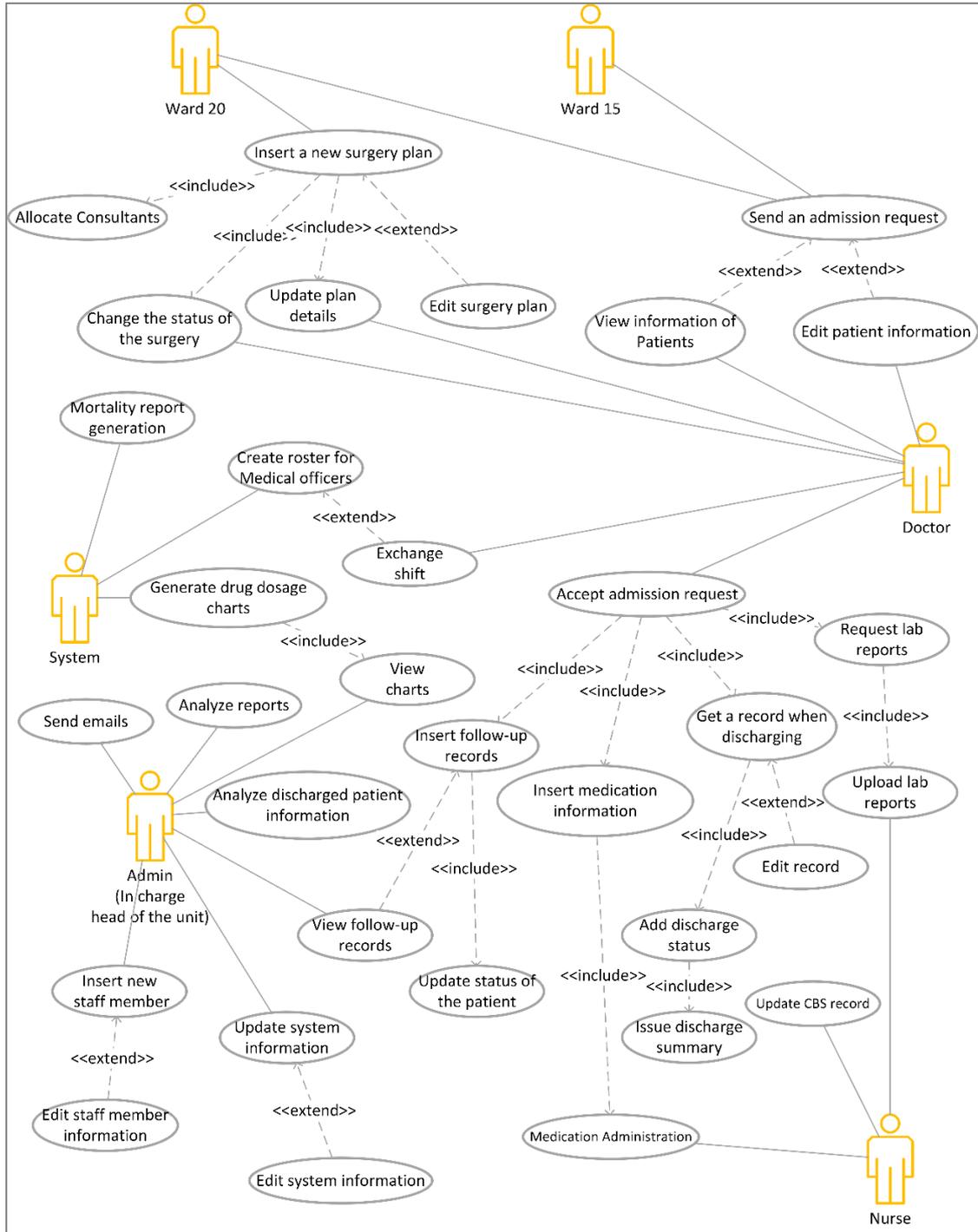


Figure 3.1 High level use case diagram for the proposed system

3.4 Sequence Diagrams

The ordered interaction of the objects during generating Discharge summary is shown in Figure 3.4.1.

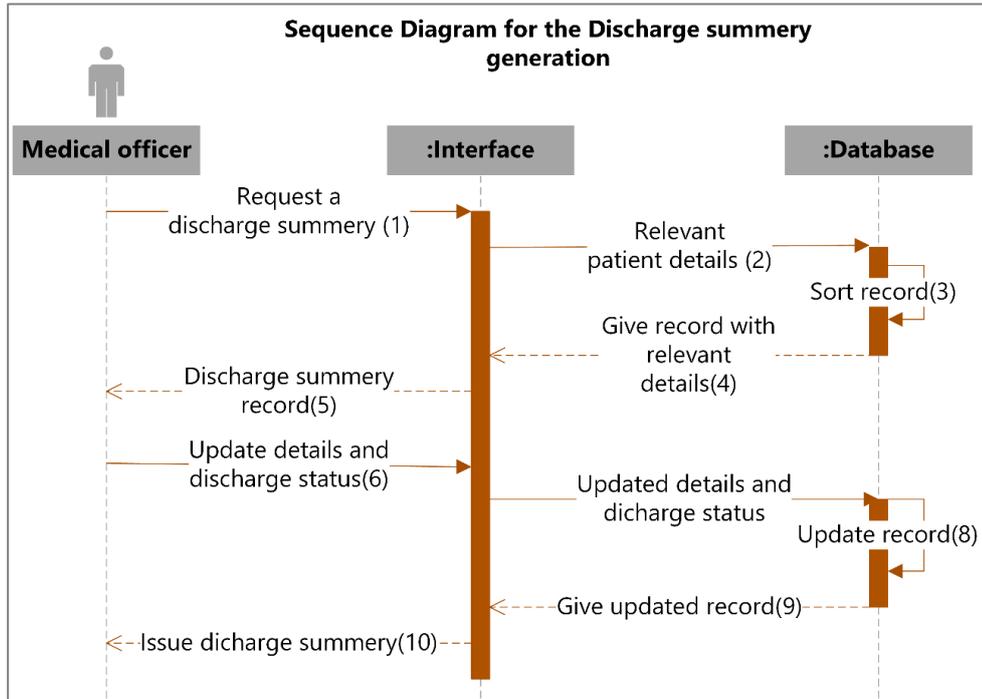


Figure 3.2 Sequence diagram for the Discharge summery generation

3.5 Entity Relationship Diagram (ERD)

Relationships among the entities of the system is shown in Figure 3.5.1.

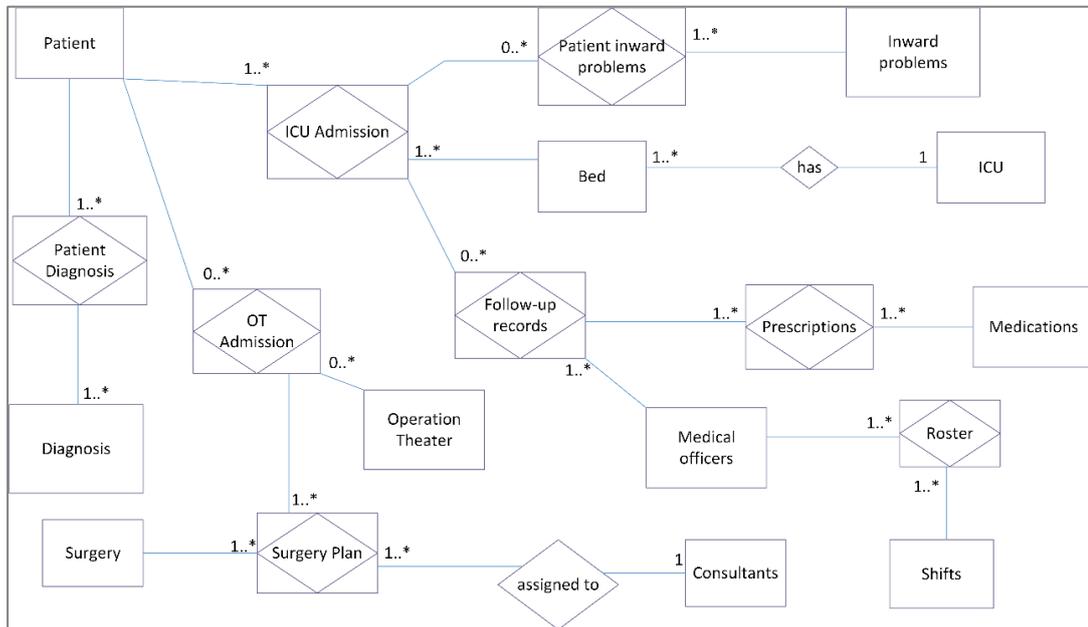


Figure 3.3 ER diagram for the system

3.6 Database Normalization

Database normalization is the process of organizing the tables and the columns of the database to reduce data redundancy and improve data integrity. Avoiding data modification issues, Simplifying data queries, Minimizing data duplication can be achieved by Database normalization.

First Normal Form

A relation in First Normal Form is the values in the relation are atomic for every attribute in the relation.

Second Normal Form

A relation in Second Normal Form is no non key attribute is functionally dependent on just a part of the key. Second Normal Form can only be violated when a key becomes a composite key.

Third Normal Form

A non-key attribute should not be functionally dependent on another non key attribute.

3.7 Database Design

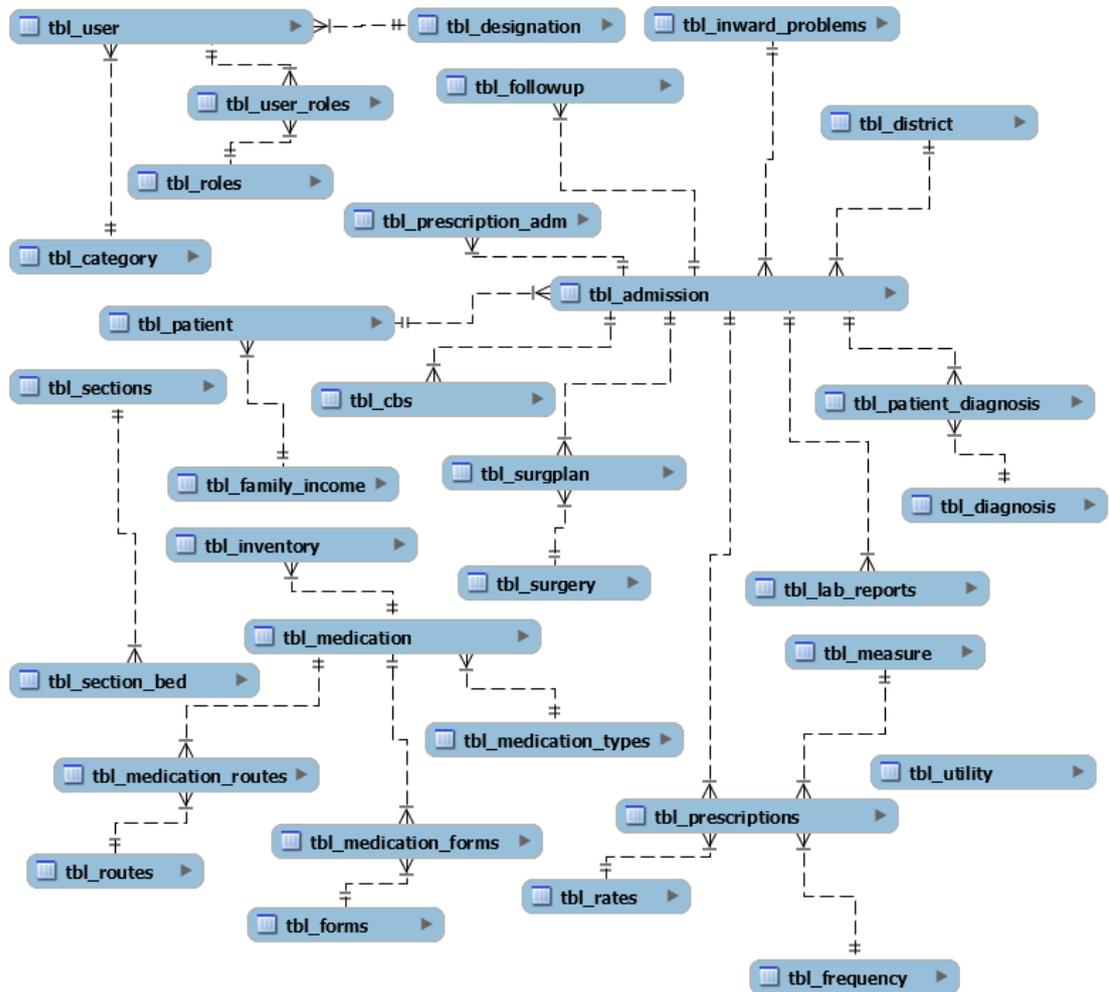


Figure 3.4 Database Design

3.8 Class Diagram

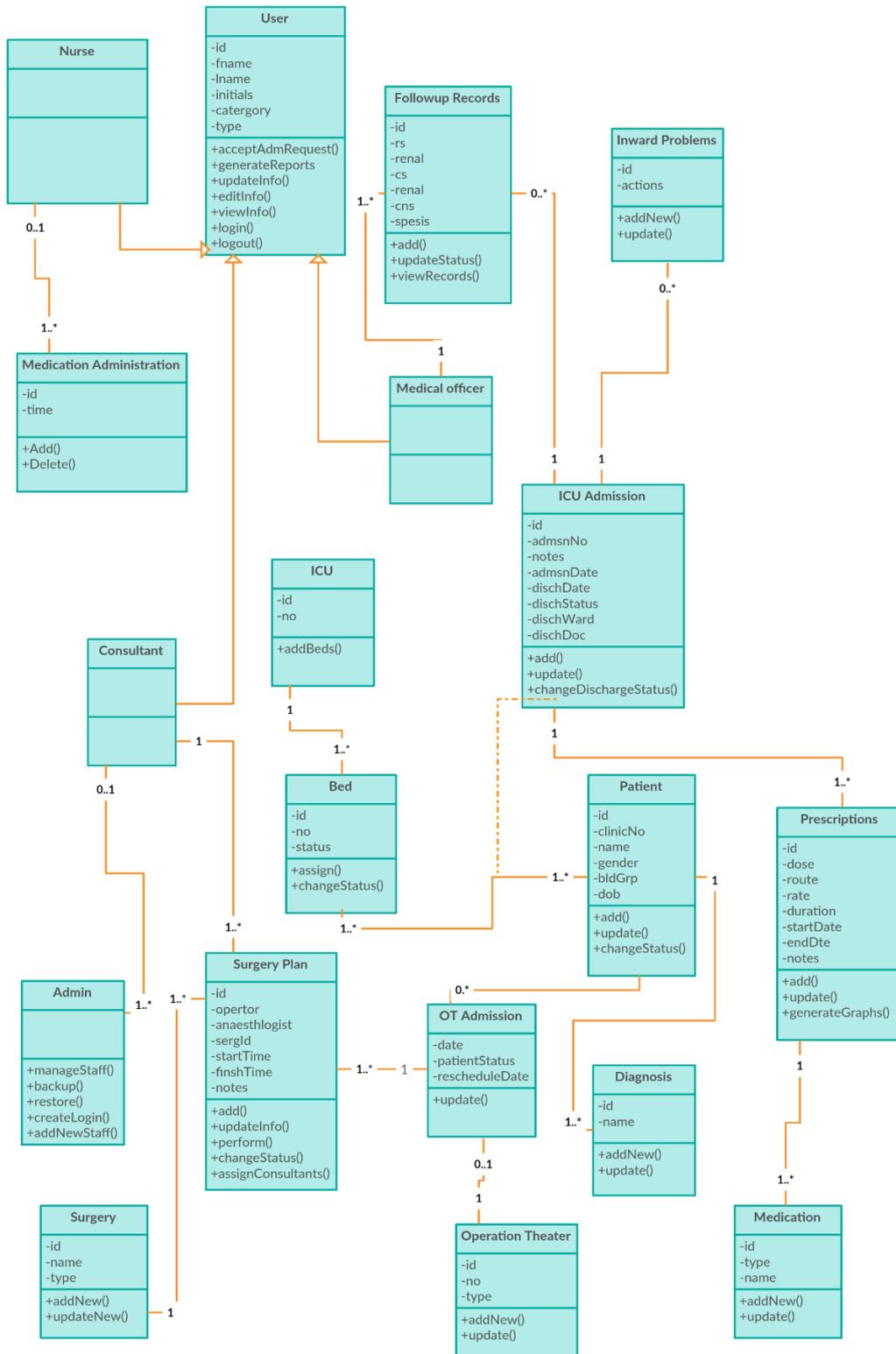


Figure 3.5 Class diagram for the system

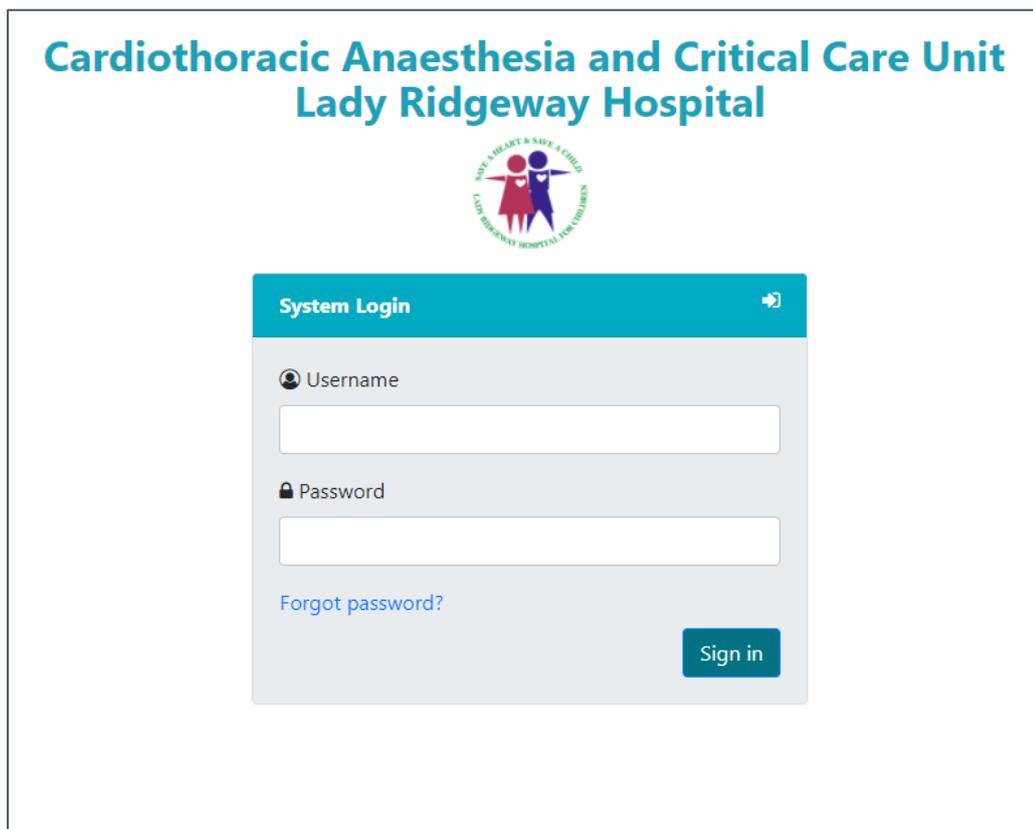
3.9 Interface design

The user interface or human-machine interface is the part of the machine that handles the human-machine interaction. [6] It is essential to design the interfaces that are easily understood by the user. Clear communication should be done with the use of terms which users are familiar with. Design should be simple and enable common tasks to be done easily.

Target users of the system are Doctors. Main priority was given to building a clear and organized application with easy navigation through the information. Medical terms were used to communicate throughout the system to set the interaction close to the typical operations, and made it easily accessible to the users with intermediate tech literacy.

3.9.1 Login Page

Shown below is the login page of the system.



Cardiothoracic Anaesthesia and Critical Care Unit
Lady Ridgeway Hospital

SAFE & HEALTHY & SAFE & SOUND
LADY RIDGEWAY HOSPITAL NEW ZEALAND

System Login

Username

Password

[Forgot password?](#)

Sign in

Figure 3.6 System Login Page

Requirements - NFR-1, NFR-2, NFR-3, NFR-5

3.9.2 Dashboard

This page shows the current state of the unit and navigation through the main modules.

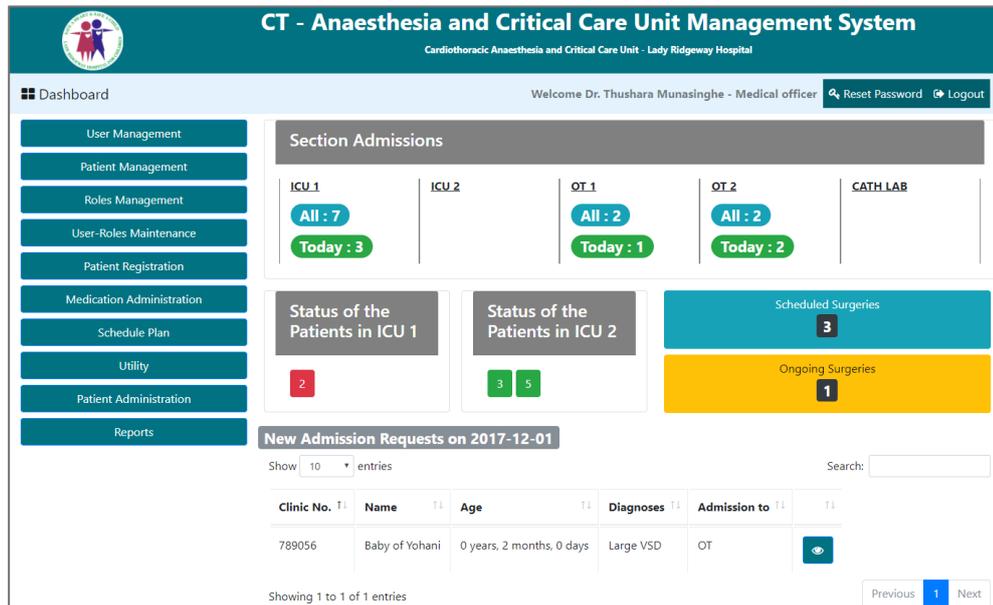


Figure 3.7 Dashboard

Requirements - FR-1, FR-3, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

3.9.3 User Management

This screen allows the Administration to add new users and manage user details.

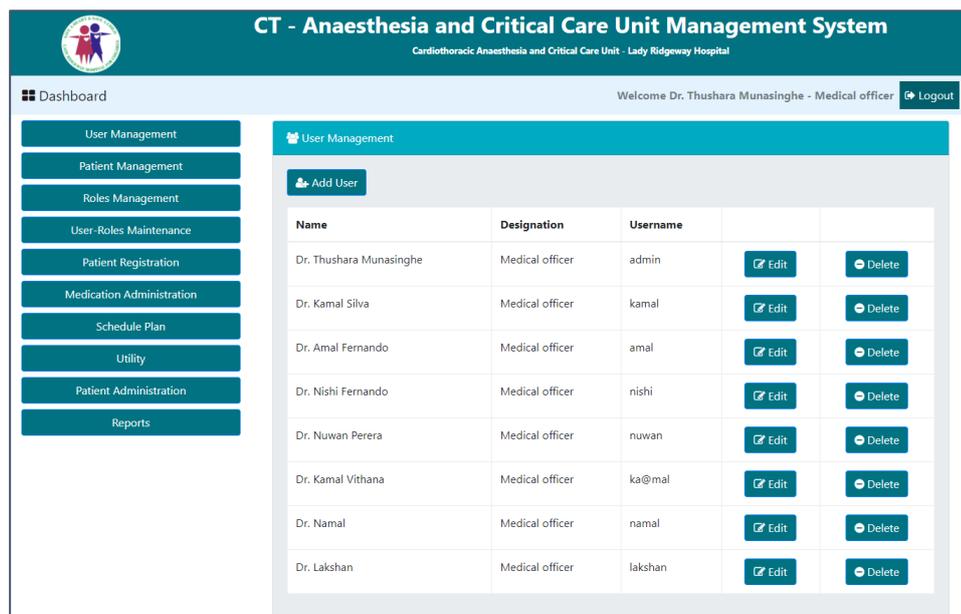


Figure 3.8 User Management Page

Requirements – FR-2, NFR-1, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

The figure below shows the Registration form of a new user.

The screenshot shows the 'User Management' section of the system. The 'View User' form contains the following fields:

- Title: Select Title (dropdown)
- Initials: Text input
- First Name: Text input
- Last Name: Text input
- Designation: Select Designation (dropdown)
- Category: Select Category (dropdown)
- Gender: Radio buttons for Male and Female
- Appointed Date: Text input
- NIC: Text input
- Registration No.: Text input
- Mobile Number: Text input
- Mobile Number (Other): Text input
- Email: Text input
- Address: Text area with 'Enter...' placeholder
- Photo: Choose File button (No file chosen)
- Username: Text input
- Password: Text input

Figure 3.9 New user Registration Form

Requirements – FR-2, NFR-1, NFR-3, NFR-5, NFR-6

3.9.4 Roles Management

Roles management screen is used to assign roles to a specific user by the administration.

The screenshot shows the 'User-Roles Maintenance' section. The 'Assign Roles' table is as follows:

User ID	Role ID	
Dr. Thushara Munasinghe	Reports, Patient Administration, Utility, Schedule Plan, Medication Administration, Patient Registration, User-Roles Maintenance, Roles Management, Patient Management, User Management	Edit
Dr. Amal Fernando	Roles Management, Patient Management	Edit
Dr. Nishi Fernando	Patient Management	Edit
Dr. Kamal Silva	Patient Management	Edit

Figure 3.10 Roles Management Page

Requirements – FR-2, NFR-1, NFR-2, NFR-3, NFR-5, NFR-6

3.9.5 New Patient Registration form

Registration form of a new patient is divided in to several sections.

The screenshot shows the 'Patient Registration' form within the 'CT - Anaesthesia and Critical Care Unit Management System' interface. The system is for 'Cardiothoracic Anaesthesia and Critical Care Unit - Lady Ridgeway Hospital'. The user is logged in as 'Dr. Thushara Munasinghe - Medical officer'. The form is divided into several sections:

- Personal Information:** Includes fields for First Name, Last Name, Date of Birth, Age (Years, Months, Days), Gender (Male/Female), Blood Group, Weight (Kg), and Height (cm).
- Contact Details:** Includes Guardian Name, Address, District, Contact Number (Land line and Mobile), and Email.
- Information for Research Purpose:** Includes Religion and Family Income (Monthly).

The form also includes a 'Submit' button at the bottom.

Figure 3.11 Patient registration form

Requirements – FR-1, FR-3, NFR-2, NFR-3, NFR-5, NFR-6

3.9.6 Patient Management

This page below shows the main navigation through Patient Management Module.

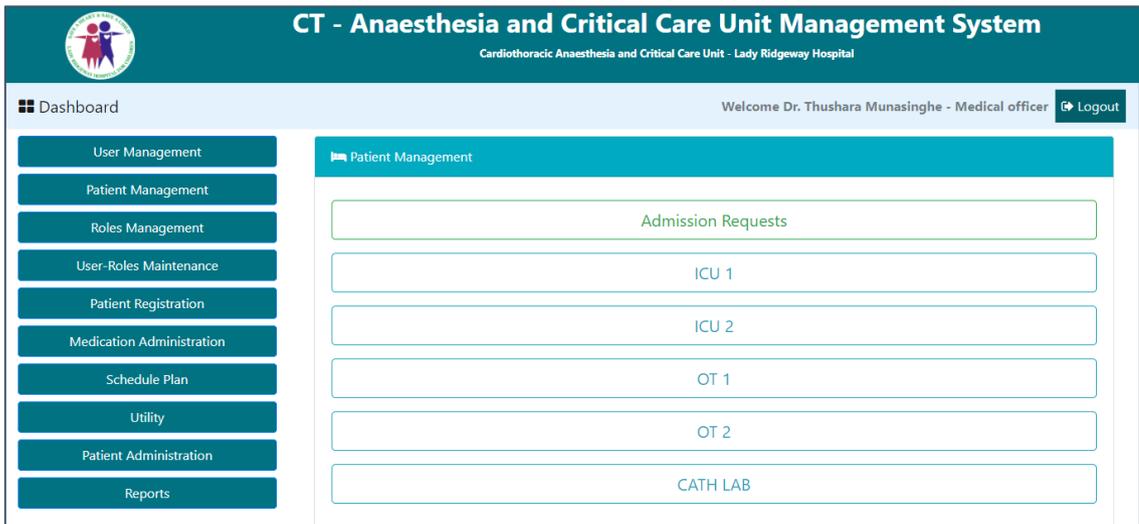


Figure 3.12 Navigation panel of the Patient Management Module

Requirements – FR-1, FR-3, FR-6, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

All new admission requests to the Unit are shown in the page below.

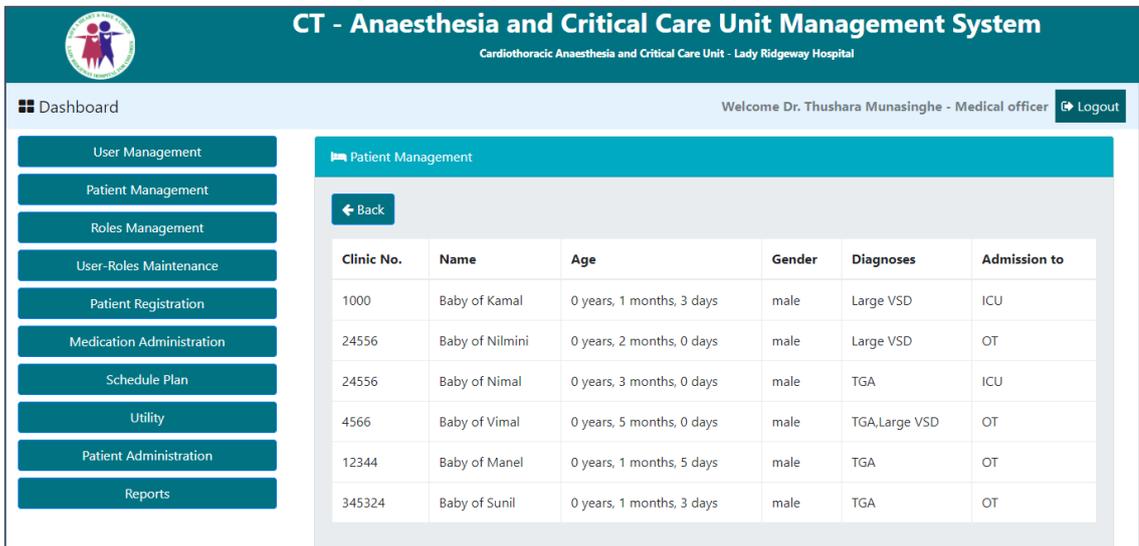


Figure 3.13 Admission Requests to the Unit

Requirements – FR-1, FR-3, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

Accordions are used to enable easy navigation through the details of a patient. Main details including the bed number of the patient is always visible above the accordion. Accordion panel contains the details such as personal, admission and surgery details, Follow-up records and Inward prescriptions.

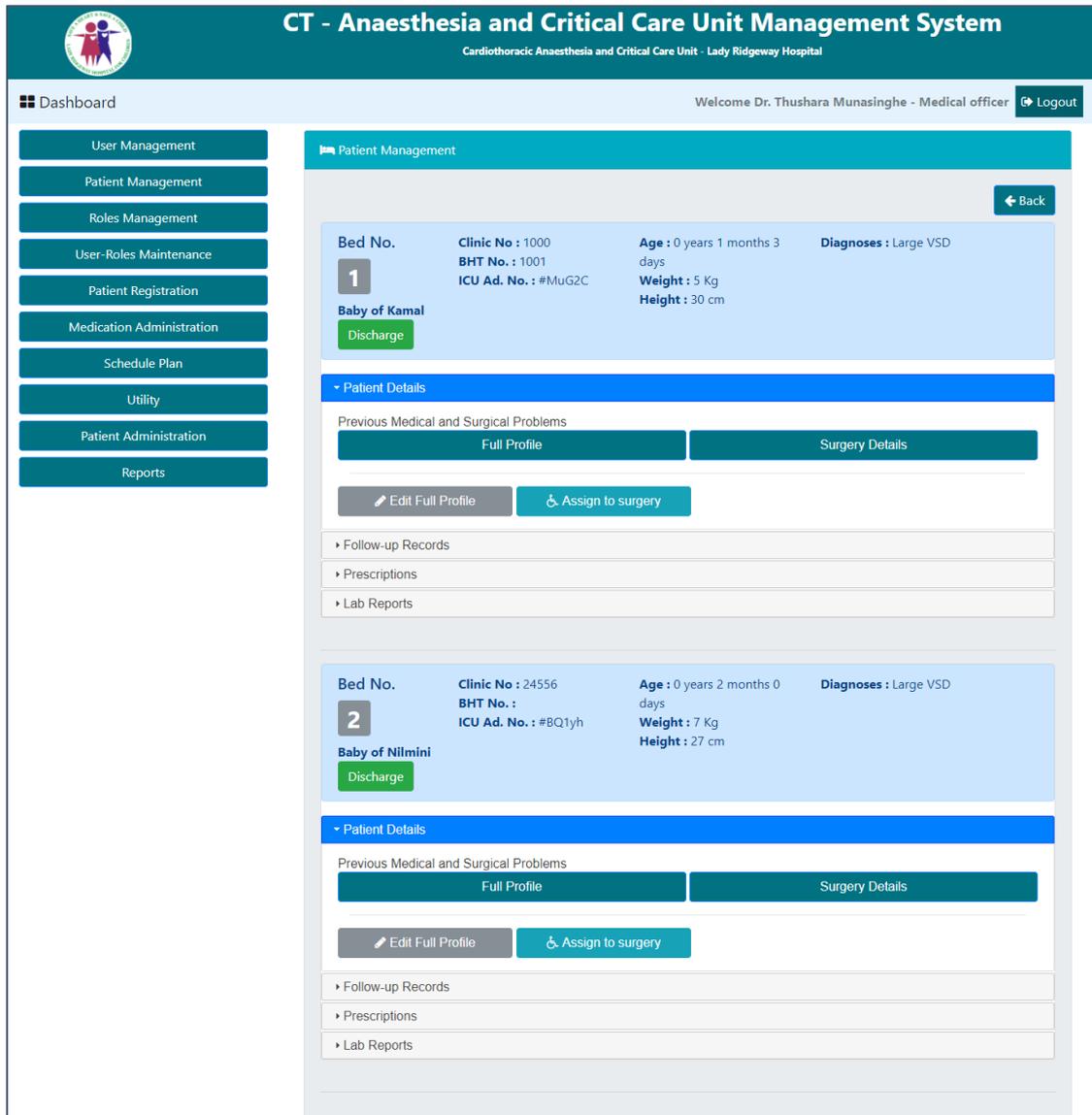


Figure 3.14 Details of patients who are admitted to the ICU

Requirements – FR-3, FR-4, FR-5, FR-6, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

Figure below shows the screen of all scheduled and on-going surgeries inside a specific Operation Theater.

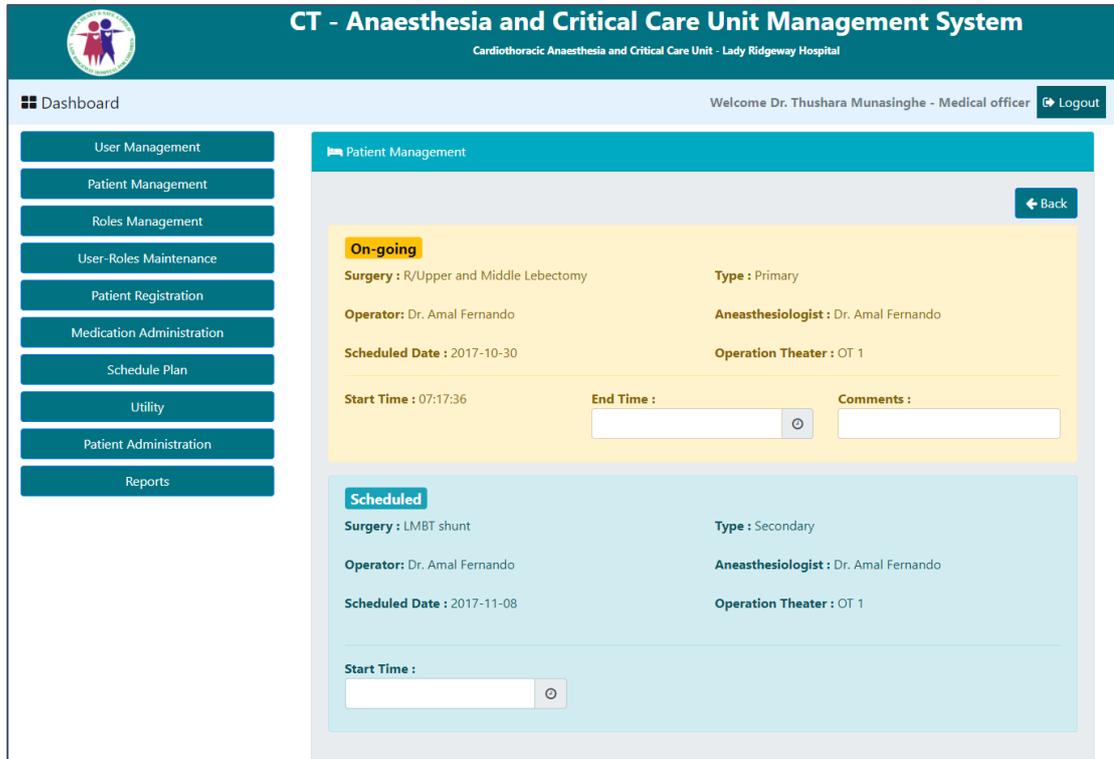


Figure 3.15 Details of Surgeries in a specific OT

Requirements – FR-3, FR-4, FR-5, FR-6, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

3.9.7 Patient Administration

Following figure shows the interface used by nurses for Patient administration.

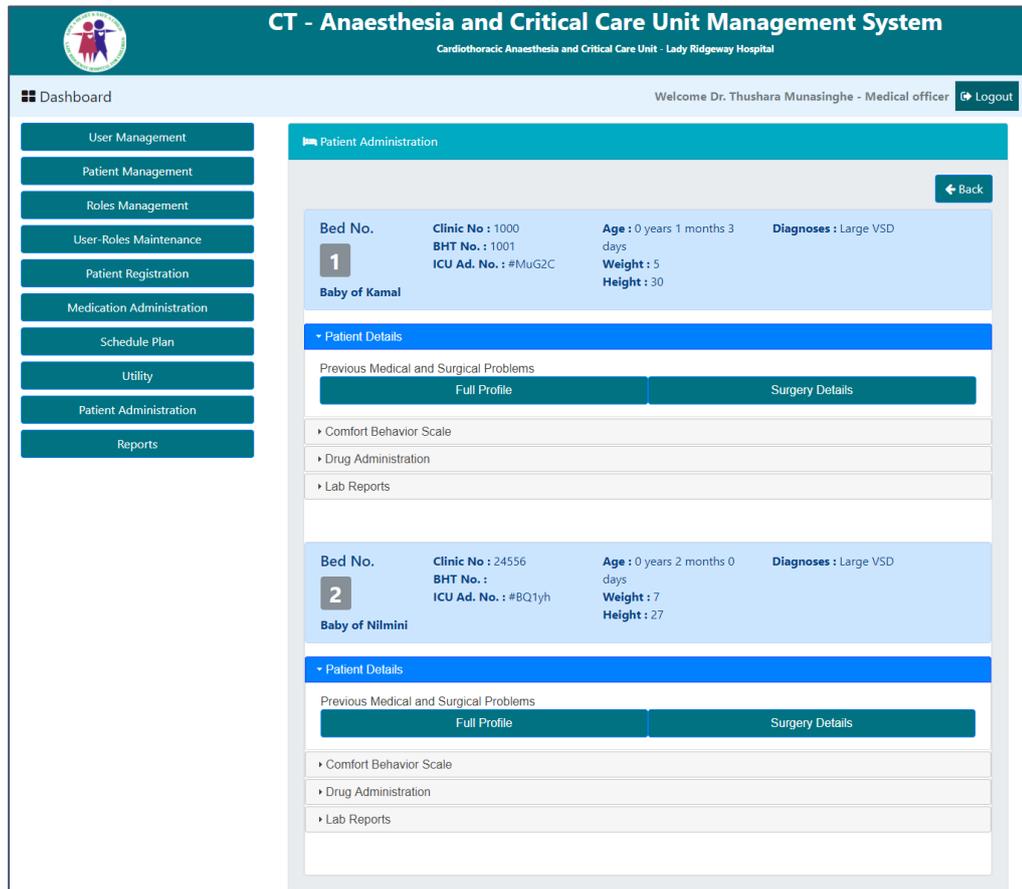


Figure 3.16 Interface of the patient administration

Requirements – FR-3, FR-4, FR-5, FR-6, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

3.9.8 Medication Management

The screen shown below is the Medication Management Page.

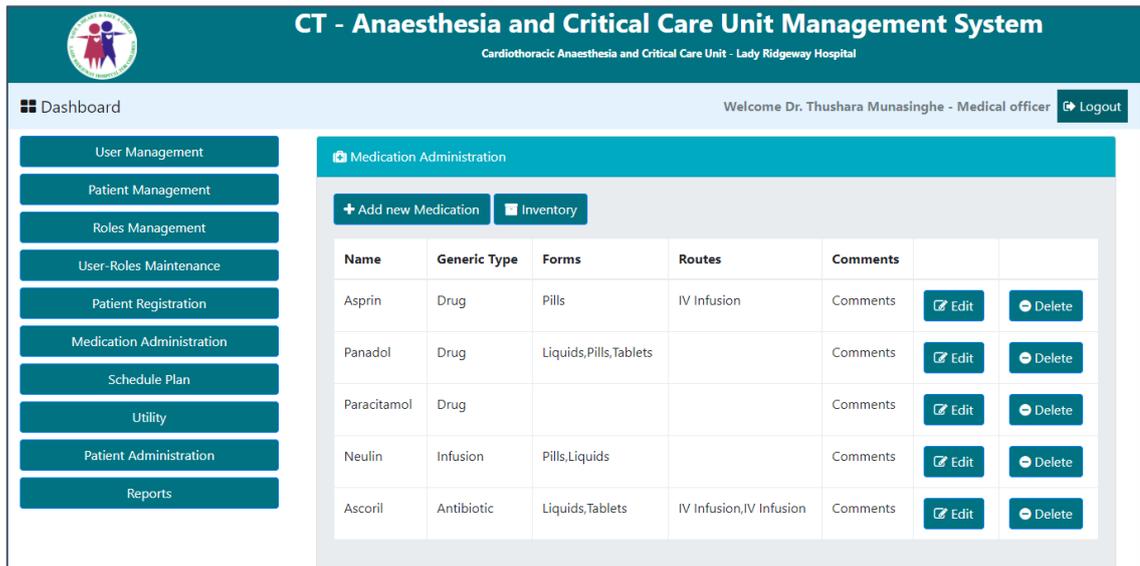


Figure 3.17 Medication Management Page

Requirements – FR-7, NFR-2, NFR-3, NFR-5, NFR-6

Inventory of the medication is controlled using the following page.

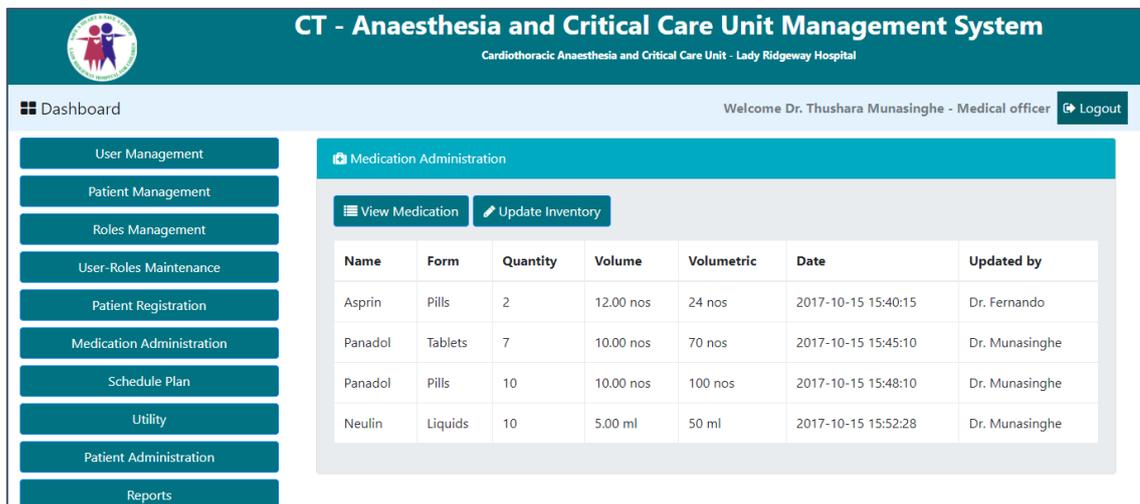


Figure 3.18 Inventory control page

Requirements – FR-7, FR-10, NFR-2, NFR-3, NFR-5, NFR-6

New Medication is registered to the system using the below form.

The screenshot shows the 'Medication Administration' section of the system. On the left is a sidebar with navigation buttons: User Management, Patient Management, Roles Management, User-Roles Maintenance, Patient Registration, Medication Administration (highlighted), Schedule Plan, Utility, Patient Administration, and Reports. The main content area is titled 'Medication Administration' and contains a 'View Medication' button. Below this are several input fields: 'Name' (text), 'Generic Type' (dropdown menu), 'Format' (checkboxes for Liquids, Pills, Tablets), 'Re-order Level' (text fields with units 'ml' and 'nos'), 'Routes' (text), and 'Comments' (text area). A 'Submit' button is located at the bottom left of the form area.

Figure 3.19 New medication registration form

Requirements – FR-7, NFR-2, NFR-3, NFR-5, NFR-6

3.9.9 Utility Management

The figure shown below is the main navigation through the utility management module. The forms and views used update system data will be provided here.

The screenshot shows the 'Utility Management' section of the system. The sidebar on the left is the same as in Figure 3.19, but the 'Utility' button is highlighted. The main content area is titled 'Utility Management' and contains a vertical list of six navigation buttons: Medication Types, Designation Categories, Surgeries, Diagnoses, Inward Problems, and Sections.

Figure 3.20 Main navigation of the Utility Management Module

Requirements – FR-7, NFR-2, NFR-3, NFR-5, NFR-6

3.9.10 Roster Management for Medical officers

This screen will allow the Admin to allocate Medical officers to a specific shift of a selected section.

The screenshot displays the 'Schedule Plan' interface within the 'CT - Anaesthesia and Critical Care Unit Management System'. The system title and logo are at the top. Below the title, it says 'Cardiothoracic Anaesthesia and Critical Care Unit - Lady Ridgeway Hospital'. The user is logged in as 'Dr. Thushara Munasinghe - Medical officer'. A sidebar on the left contains navigation buttons for various system functions. The main area shows a 'Schedule Plan' for the month of April 2017, specifically for 'ICU 1'. The plan is organized into three columns: Morning, Evening, and Night. Each day from 1 to 30 is listed in a row, with three dropdown menus under each shift column for assigning medical officers.

	Morning	Evening	Night
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Figure 3.21 Schedule plan

Requirements – FR-9, NFR-1, NFR-2, NFR-3, NFR-5, NFR-6

3.9.11 Report Module

Useful reports for the administration and management of the Unit is generated here.

The screenshot shows the 'Reports' page of the 'CT - Anaesthesia and Critical Care Unit Management System'. The system is for 'Cardiothoracic Anaesthesia and Critical Care Unit - Lady Ridgeway Hospital'. The user is logged in as 'Dr. Thushara Munasinghe - Medical officer'. The sidebar on the left contains various management options. The main content area features a search interface for reports, including fields for 'From', 'To', and 'Admission to', along with 'Print', 'Export to PDF', and 'Export to Excel' buttons. Below the search area is a table with the following data:

#	Admission Date	Admission To	Number of Admissions
1	2017-11-03	ICU	2
2	2017-11-03	OT	1
3	2017-11-04	ICU	1
4	2017-11-05	OT	2

Figure 3.22 Report Generation Page

Requirements – FR-8 FR-10, NFR-1, NFR-2, NFR-3, NFR-4, NFR-5, NFR-6

Chapter 4 – Implementation

4.1 Introduction to Implementation

The implementation is a process of defining how the project should be developed to ensure it is operational and the project goals are correctly met with the required quality. The design of the system should be implemented and converting the design of the system to machine understandable language is required. The requirements identified in the previous chapters are converted into real outcomes using computer languages and computer hardware during this phase. Codifying of the design into computer languages and getting the required output in to an attractive and user friendly way is the most important task during this phase.

4.2 Implementation Environment

The System Implementation Environment is mainly divided into two categories: Hardware Environment and Software Environment.

4.2.1 Software Environment

Codeigniter framework which is a powerful yet simple PHP framework, has been used to create the system. One advantage of using Codeigniter is that it has a very small footprint. The platform used in developing WAMP is a free and open source Windows web development environment. It includes Apache, My SQL and PHP. WAMP stands for Windows (W), Apache (A), MySQL (M) and PHP (P).

Supported Languages and Technologies

- Ajax
Ajax is a JavaScript based technology and it supports updating the system components without refreshing the whole system page.
- CSS
CSS stands for Cascading Style Sheet. This is used for the styling the HTML pages of the system.

- **HTML**
This is the basic web related language and it helps to keep the system structure clear and conscious. Stands for Hyper Text Markup Language.
- **MYSQL**
This is the Database software which was used to handle the database of the system.
- **JavaScript**
This is a Scripting language which helps for client side validation and to dynamically create the system components.
- **JQuery**
It is a JavaScript library and used as reusable component when developing.
- **PHP**
This is a server side object oriented scripting language which was used when developing the system.
- **NetBeans**

This is an Integrated Development Environment (IDE) tool that combines the basic tools developers need to develop and test software. It generally contains source code editor, build automation tool and a debugger. NetBeans is a cross platform software development platform written in Java and PHP and HTML5 are also supported among other software development tools.

Other supported software and frameworks

- **Documentation**
MS Word 2016, MS Project 2016, MS Visio
- **Report generation Tool**
mPDF
- **Image Processing**
Adobe Photoshop, Adobe Illustrator

4.2.2 Hardware Environment

This system has been developed using a computer with the following requirements.

- **Processor: 3.0GHz Intel Processor**

- Hard Disk: 512MB
- RAM: 4GB
- Operating System: Windows 7/ Windows 8/ Windows 10
- Display: 1024 x 768 resolution Monitor
- Printer: Inkjet or laser printer
- Internet: ADSL connection with minimum speed of 512kbps.

4.3 Actual Implementation

System is developed using the MVC architecture where the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

- Model

Represents data structure. This contains functions that retrieve, insert, update and delete data (CRUD operations) in the database.

- View

Represents the information that is being viewed by the user via the user interface.

- Controller

Act as intermediate between model and view. Controller process user requests.

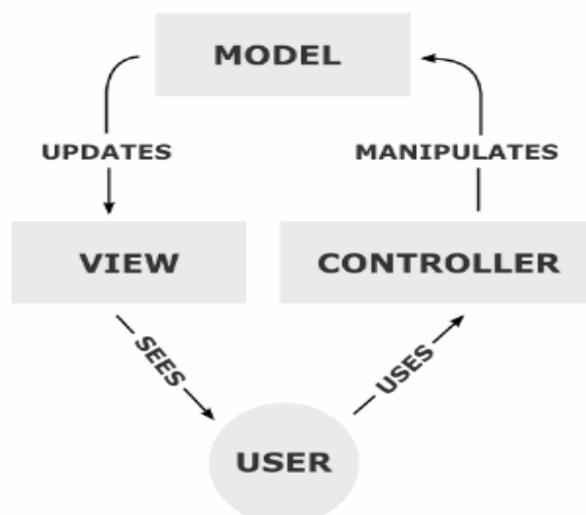


Figure 4.1 MVC Architecture representation

Database can only be accessed via the application server. These three tiers are implemented on the same computer and accessed through local-host only for the ease of development and testing purposes. This system totally used Codeigniter framework.

4.3.1 Database server

WAMP server should be installed on the computer dedicated for the database server. WAMP server enables running phpMyAdmin for more appliance. MYSQL is used as the database server which is included in the WAMP solution stack.

4.3.2 Web server

The computer used as the web server should be installed with WAMP. Therefore Apache is used as the server.

4.4 File structure of the developed system

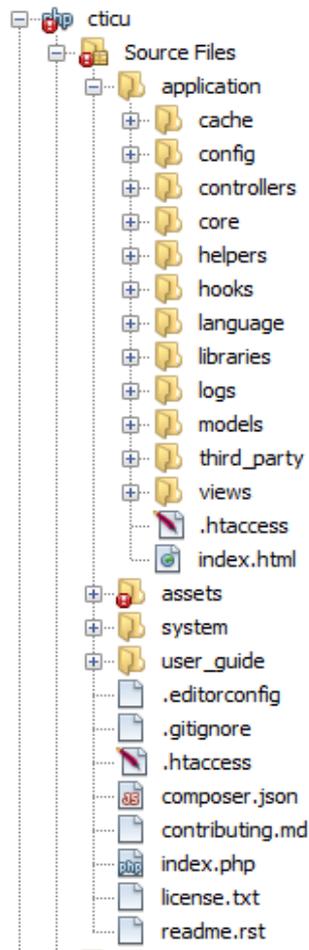


Figure 4.2 File structure of the developed system

4.5 High-level Module structure

M-1. User Management – Accessed by Administration

M-2. User-roles Maintenance – Accessed by Administration

M-3. Patient Management – Accessed by Doctors, Wards

M-4. Patient Administration – Accessed by Doctors, Nurses

M-5. Medication Management – Accessed by Nurses

M-6. Roster Management – Accessed by Administration, Doctors

M-7. Utility Maintenance – Accessed by Administration

M-8. Reports – Accessed by Administration, Doctors

4.6 Mapping between Modules and Functional Requirements

Module	Functional Requirements
M-1	FR-2
M-2	FR-2
M-3	FR-1, FR-3, FR-4, FR-6
M-4	FR-1, FR-3
M-5	FR-7
M-6	FR-9
M-7	FR-7
M-8	FR-5, FR-8, FR-10

Table 4.1 Mapping between Modules and Functional requirements

4.7 Major code segments

Actual segments of the major codes which are used in the system are included in the Appendix F.

Following are the basic pseudocode descriptions for some major codes used.

- User Login

Enter username and password

check session

if session exist

direct to dashboard

else

redirect to login page

- Validating Username and Password.

if username is not empty and password is not empty

read username and password, check against database

if they are correct

create a session with the user information

else

print an error message

- Check username existence

```
read username  
  
check against database for existence  
  
if exist  
    print "Username not available"  
else  
    print "Username available"
```

- Logout User

```
destroy session  
  
direct to the login page
```

- Generating code for ICU Admission No.

```
X = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'  
  
REPEAT  
    shuffle X  
    create random code with 5 characters  
    check against database for existence  
  
UNTIL  
    code does not exist in the database
```

Following is the actual code snippet for routing.

```
$route['default_controller'] = 'welcome';

$route['404_override'] = "";

$route['translate_uri_dashes'] = FALSE;

//Routes for User controller

$route['user/login'] = 'user/validate';

$route['user/logout'] = 'ser/logout_user';

$route['user/management'] = 'user/manage';

$route['user/add'] = 'user/add_user';

$route['user/save'] = 'user/save_user';

$route['user/edit/(:num)'] = 'user/edit_user/$1';

$route['user/update'] = 'user/update_user';

$route['user/delete'] = 'user/delete_user';

$route['user/list'] = 'user/list_user';

//Routes for Roles controller

$route['roles/management'] = 'roles/manage';

$route['roles/add'] = 'roles/add_roles';

$route['roles/save'] = 'roles/save_roles';

$route['roles/edit/(:num)'] = 'roles/edit_roles/$1';

$route['roles/update'] = 'roles/update_role';

$route['roles/delete'] = 'roles/delete_role';

$route['roles/list'] = 'roles/list_role';
```

4.7 Re-usable components

Some of the well tested re-usable components have been used when implementing the system to add more attractiveness and to maximize the efficiency and performance of the system.

- Codeigniter - Codeigniter is a powerful PHP framework that utilize the MVC architecture.
- Bootstrap v4 - This is most popular and open source CSS framework in web development.
- CSS re-usable components
Color theme has been used when styling some parts of the system..
- JQuery re-usable components

Code for the dropdown menu has been used when styling certain menu items.

Date Picker has been used when designing date selection controllers.

Time Picker has been used when designing time selection controllers.

Code for Accordion has been used when styling.

- Some re-usable JavaScript and Ajax Functions have been used to form submission, load content, report generation and to communicate data between the system and the server efficiently and quickly.

Chapter 5 – Evaluation

5.1 Introduction

This chapter discuss the Evaluation of software which is a process of testing of the system or its components with the intent of finding whether the developed software satisfies the specified requirement of the Client. It is an important phase of a successful system. After developing the whole programs of the system, a test plan should be developed and run on a given set of dummy data. The actual output of the test run should match the expected result.

5.2 System Testing

System Testing is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. [7]

The complete system is executed on the actual data in system testing. At each stage of the execution, the results of the specific execution is analyzed to find defects. During the result analysis, it may be found that the outputs are not matching with the expected output of the system. In such case, the errors in the particular programs are identified and are fixed and further tested until system gives the expected output.

All system tests are carried out according to a test plan to clarify whether all the functional requirements are properly met and user requirements are satisfied.

5.3 Test Plan

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project. [8]

Quality assurance will be used to validate the quality and the correctness of this system, this is done prior to using the system in the real environment. It also contains various resources required for the successful completion of the project.

A unit testing was carried out for the functions and procedures to ensure that they work properly and individually as intended. This is a code level testing which tests the units of codes. An integration testing has been done in order to test the modules of the system. This is essential when using different kinds of implementing technologies together. It ensures that all the integrated components are working cooperatively for the desired outcome. Then the System test is done. Through a User acceptance testing a confirmation is done by the client and actual users of the system. This verifies whether the user requirements are properly met.

5.4 System Test Cases

The most important contents of any test plan are the test cases. Generally test cases include the test case name, execution conditions, expected results, actual results and the priority. System has divided in to modules in order to reduce the complexity. Test cases were written for each module separately. This will simply decide whether the system will pass or fail the test. Test cases are the component used for quality assurance of the system.

Module Name	Related Functional Requirement	Function Name	Test Priority
User Authentication Module	FR-2	Check username	High
	FR-2	Check password	High
User Management Module	FR-2	Add Users	High
	FR-2	Edit Users	High
	FR-2	Delete Users	High
	FR-2	Search Users	Medium
	FR-2	Confirm update of Users	Law
	FR-2	Confirm deletion of Users	Law
	FR-2	Confirm deletion of Users	Law
User-roles Maintenance Module	FR-2	Assign Roles to a User	High
	FR-2	Edit assigned Roles	High
	FR-2	Confirm update of assigned Roles	Medium
Patient Management Module	FR-1	Patient Registration	High
	FR-1	Accept Admission Requests	High
	FR-1	Edit Patient Details	High

	FR-3	Assign patient to a surgery	High
	FR-3	Change Surgery status	High
	FR-4	Add follow-up records	High
	FR-4	Add prescriptions	High
	FR-4	Request lab reports	Medium
	FR-4	Delete prescriptions	Medium
	FR-4	Delete lab report requests	Medium
	FR-6	Issue Discharge summary	High
Patient Administration Module	FR-1	Add CBS record	High
	FR-1	Administer drugs	High
	FR-1	Upload Lab reports	Medium
	FR-1	Delete lab reports	Medium
	FR-1	Delete administered medication	Medium
Medication Management Module	FR-7	Add Medication	High
	FR-7	Edit Medication	High
	FR-7	Delete Medication	High
	FR-7	Update Inventory	High
Roster Management Module	FR-9	Create Monthly Roster	High
	FR-9	Update Roster	High
	FR-9	Email Monthly Roster	Medium
	FR-9	Exchange Shifts	High
Utility Maintenance Module	FR-7	Add Utility Items	High
	FR-7	Edit Utility Items	Medium
	FR-7	Delete Utility Items	Medium
Reports Module	FR-5	Daily Admission to the Unit	High
	FR-5	Monthly Admission to the Unit	High
	FR-8	Mortality Reports	High
	FR-10	Drug usage Reports	High

Table 5.1 System test cases

5.5 Test Data and Test results

Actual tests for the proposed system are mentioned with the related screenshots of the test result in Appendix E. Further the actual test results mentioned as pass or fail.

5.6 User Acceptance Test

After implementing the system, it was tested in the real environment using the actual data before the final product is released to the client. This is hold to be the final step of testing phase. The system was tested by the client to identify the functionality provided by the system, whether it can satisfy the operational needs. By introducing the system and the system functionalities briefly, the client could understand how the system functionalities work. User Acceptance Testing was started by feeding the actual data to the system. There were some minor modifications required for the system, some were requested by the client and actual users.

User Acceptance testing was done through four different user levels.

- Administration
- Doctors (Consultants and Medical officers)
- Nurses
- Ward (In-charge Doctors)

User feedback form for the system

User Acceptance Test form of Cardiothoracic Anaesthesia and Critical Care

Unit Management System

To provide a better solution, please provide your correct feedback about the system.

Name of the User:

Role of the User:

No	Evaluation Item	Rating				
		Very Good	Good	Average	Bad	Very Bad
01	User Friendliness of the system					
02	Interface design of the system					
03	Easy navigation through the functionalities of the system					
04	Form validation					
05	Proper, meaningful and understandable error messages generated by the system					
06	Ease of generating reports					
07	Response time of the system					

Signature

Date

Thank you for your feedback.

Figure 5.1 User Acceptance Testing evaluation form

**User Acceptance Test form of Cardiothoracic Anaesthesia and Critical Care
Unit Management System**

To provide a better solution, please provide your correct feedback about the system.

Name of the User: Dr. M.T.D. Munasinghe

Role of the User: Medical officer

No	Evaluation Item	Rating				
		Very Good	Good	Average	Bad	Very Bad
01	User Friendliness of the system	✓				
02	Interface design of the system	✓				
03	Easy navigation through the functionalities of the system		✓			
04	Form validation	✓				
05	Proper, meaningful and understandable error messages generated by the system	✓				
06	Ease of generating reports		✓			
07	Response time of the system	✓				

Signature M.T.D. Munasinghe Date 20/10/2017

Thank you for your feedback.

Figure 5.2 User Acceptance Test Feedback form

User Acceptance Test Results

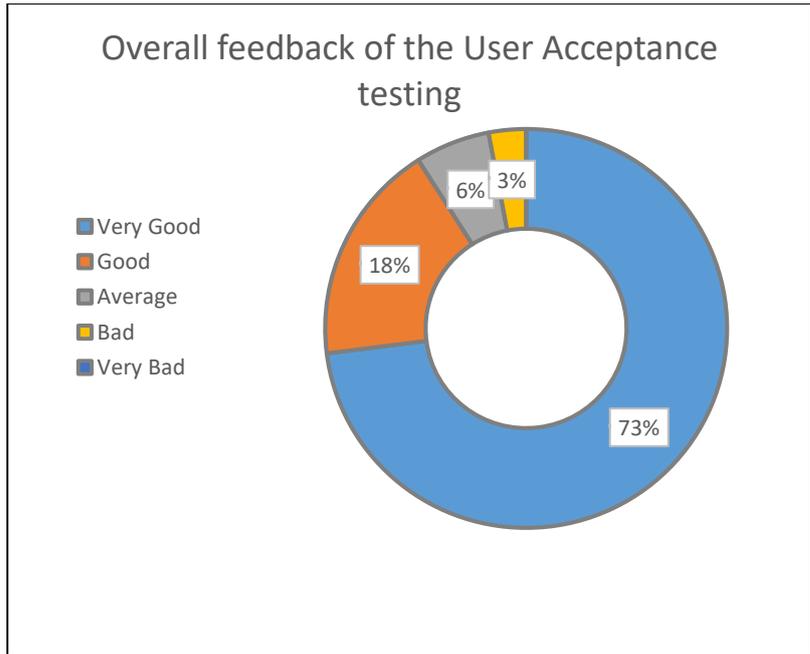


Figure 5.3 User Acceptance Test Results

Chapter 6 – Conclusion

6.1 Introduction

This Chapter will discuss the overview of the developed system, future enhancements which can be done to improve the functionality and usability of the system as well as the lessons learnt throughout the time period which the project work carried out.

6.2 Overview of developed system

Main objective of this project is to provide a more efficient and easily accessible system which ease the work-flow of the Cardiothoracic Anaesthesia and Critical Care Unit of Lady Ridgeway Hospital.

Earlier they have maintained the processes mainly by using a traditional manual system. They had faced lots of difficulties in the areas such as patient records management, History maintenance, Keeping track of the follow-up records, medication and prescription records and details management of the surgery plans of the unit.

Developed system has made a big impact on the work-flow of the unit. It was a great experience for both the working staff and management of the unit to work with a computer software which has undertaken many of the work including lot of paper-work. This provides facility to reduce time of processes and carry out their daily activities effectively and efficiently. Useful reports will be provided by the system that will be helpful to the administration and to the doctors of the unit. Management also found it easy as the system gives information that is needed to take correct and timely decisions.

Development of a system for a well-known government hospital in Sri Lanka was a great opportunity to use the knowledge gained throughout years, to be practically used and tested. This can be considered as a remarkable achievement during my preparation for Bachelor of Information Technology degree in the University of Colombo. It was a pleasure to provide a better and improved solution to the Cardiothoracic Anaesthesia and Critical Care unit of Lady Ridgeway hospital and they have provided a motivating support throughout the development of the project.

6.3 Lessons learnt

It was a great opportunity to interact with the government health field of Sri Lanka. It was able to get the overall view and ideas of Medical staff and what they think of having able to electronically manipulate medical records and get the help of information and technology to enhance the work process. The chance was provided to experience the real world challenges in development field and how can we practically use the knowledge gained to give a better solution to achieve the requirements.

Technical and analytical knowledge was improved and gained further experiences to work with programming languages such as PHP and Codeigniter framework.

It was able to learn how to use and apply the theoretical knowledge of HTML, CSS, JavaScript, AJAX and MVC Architecture to real world projects.

Dissertation writing was helpful to improve the professional writing skills.

Further, working with different kind of individuals was helpful to improve the intellectual skills and personality. It was able to learn how to gather relevant and useful information from people through conducting interviews and observations.

6.4 Future Enhancements

Issue a Digital card to every patient admitted to the Unit

Digital card with a barcode can be issued to the patients which all the patient details in the system can be retrieved through this card.

Network Implementation to share patient details with other hospitals.

A network can be introduced to share all the patient details with other hospitals and share information with doctors. Therefore the basic treatment to the patient can be done in any hospital which has the facilities to do so.

Alert System for Doctors about critical patients.

Alert system using the mobile network can be introduced to instantly be alerted about the patient and contact other doctors.

References

- [1] SearchSoftwareQuality. (2017), What is requirements analysis (requirements engineering)? - Definition from WhatIs.com, [Online] Available at: <http://searchsoftwarequality.techtarget.com/definition/requirements-analysis> [Accessed 2 Jun. 2017].
- [2] Digitalcairo.com (2017), Simple EMR, [Online] Available at: http://www.digitalcairo.com/simple_emr.aspx [Accessed 5 Jun. 2017]
- [3] Hhims.org. (2017), HHIMS Overview - hhimsv2, [Online] Available at: <http://www.hhims.org/hhims-overview> [Accessed 5 Jun. 2017]
- [4] En.wikipedia.org. (2017), Rational Unified Process, [Online] Available at: https://en.wikipedia.org/wiki/Rational_Unified_Process [Accessed 6 Jun. 2017]
- [5] <http://www.conceptdraw.com>. (2017), UML Diagram Types List, [Online] Available at: <http://www.conceptdraw.com/How-To-Guide/uml-diagram-types-list> [Accessed 10 Jun. 2017].
- [6] En.wikipedia.org. (2017), User interface, [Online] Available at: https://en.wikipedia.org/wiki/User_interface [Accessed 6 Jun. 2017].
- [7] Stf, "System Testing," Software Testing Fundamentals, 04-Feb-2015. [Online]. Available: <http://softwaretestingfundamentals.com/system-testing/>. [Accessed: 01-Oct-2017].
- [8] Stf, "System Testing," Software Testing Fundamentals, 04-Feb-2015. [Online]. Available: <http://softwaretestingfundamentals.com/system-testing/>. [Accessed: 14-Oct-2017].

Appendix A - System Documentation

Step by step guidance on how to install the system properly for the Cardiothoracic Anaesthesia and Critical Care Unit of Lady Ridgeway Hospital, Colombo is provided by the system documentation. The necessary software and hardware requirements are included in to this document which will be needed to run the system and the tools that will be helpful to do any modifications to the system in future. Hardware and Software requirements needed to run the system have been mentioned below.

Hardware Requirements

- Hard Disk Capacity – 40GB Hard Disk Capacity
- Processor – 3.0 GHz Intel Processor
- Memory – 512 MB Memory Capacity
- Display - 1024*768 Resolution Monitor
- Printer – Inkjet Printer
- Internet – ADSL Connection (Minimum Speed 512Kbps)

Software Requirements

- Operating System – Windows 7, Windows 8, Windows 10
- Server – WampServer Version 3.0.6 64bit
- Image Editor – Adobe Photoshop CC
- Web browser – Google Chrome Version 61.0.3163.100 (Official Build) (64-bit)/
Mozilla Firefox Version 56.0.2 (64-bit)
- IDE – NetBeans IDE 8.1

Installing the system

Step 1: Wamp server installation

- Download WAMP server from <http://www.wampserver.com/en> (Make sure to install correct version according to your operating system.)
- Refer to the Installation guide provided by the above web site and install the WAMP server on your computer.

Step 2: Re-storing the SQL file.

- Open your web browser and type <http://localhost/phpmyadmin> then navigate.
- Log in to the database and provide a username (“root”) and password if you want to set a password.
- Create an empty database providing a name “db_icu”.
- Click the “Import” button to choose a file.
- Browse the CD and select the “db_icu” file located inside the Database folder.
- Click “Go” button.

Step 3: Launching the system

- Open your web browser and type <http://localhost/cticu> and navigate.
- Provide valid username and password to log in to the system.
- Please refer Appendix C – User Documentation to get a quick guide on how to operate the system.

Appendix B – Design Documentation

Use Case diagrams with detailed information

Following is the use case diagrams in detail. To understand it easily and clearly, it was categorized into sub modules and each module explained with a small description.

User Management Module

User handling is a very important aspect of the system. This module allows administrator to easily add new user to the system, edit relevant user details and related activities. Also this assigning roles and make changes to them will be allowed. Figure B.1 has shown the use case diagram for this module and Table B.1 describes the description of the use case

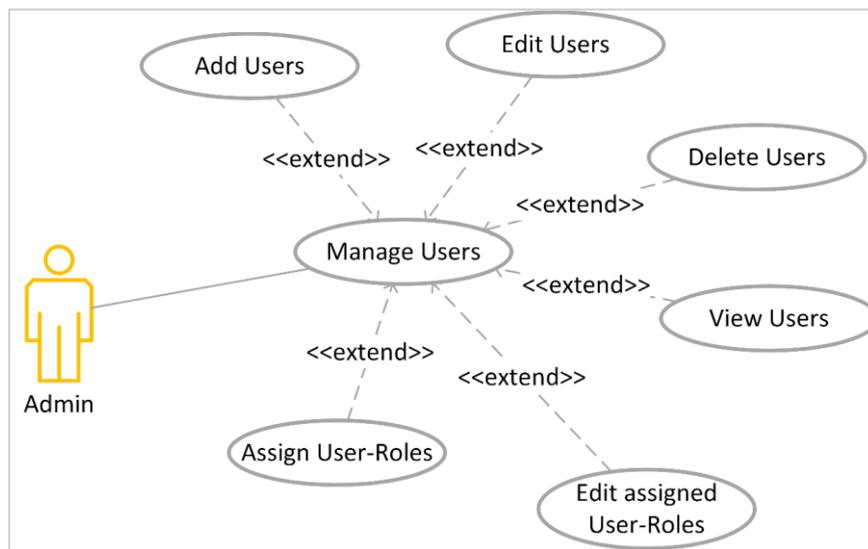


Figure B.1 Use case Diagram – User Management

Use case	User Management Module
Actors	Admin
Description	Perform user related activities in the system
Pre-conditions	
	<ol style="list-style-type: none"> 1. User should be logged in to the system 2. Specific roles should be assigned to the user to perform the specific task.
Basic Course of events	
	<ol style="list-style-type: none"> 1. Check the user availability. 2. Register new users to the system 3. Edit and delete user details 4. Assign roles to the user
Post-conditions	
	<ol style="list-style-type: none"> 1. User can perform specific tasks according to the role 2. User details can be used to report generation and system details.

Table B.1 Use case description – User Management

Patient Management Module

This is one of the main modules of the system. Admissions to the Unit from the ward is managed here.

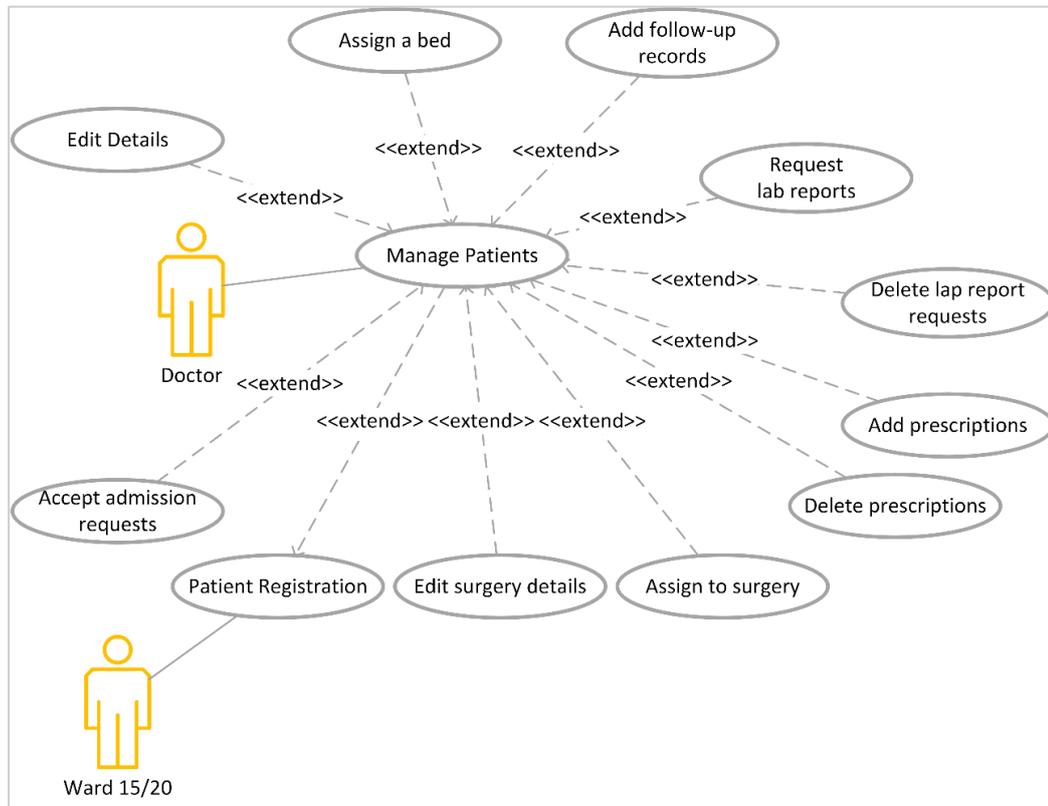


Figure B.2 Use case Diagram – Patient Management

Use case	Manage patients
Actors	Ward 15, Ward 20, Doctors
Description	Perform patient related activities in the system
Pre-conditions	
<ol style="list-style-type: none"> 1. User should be registered in the system. 2. Specific role should be assigned to the user. 3. User should be logged in to the system. 	
Basic Course of events	
<ol style="list-style-type: none"> 1. Patient registration. 2. Accept admission requests. 3. Assign beds. 4. Create follow-up records. 5. Enter prescription details 6. Create a lab report request 	
Post conditions	
<ol style="list-style-type: none"> 1. Discharge summary can be issued when discharging a patient. 2. Details can be used for the report generation. 	

Table B.2 Use case description – Patient Management

Medication Management Module

Medication Management module is usually controlled by the nurses in the Unit. This module allows the user to add, update and delete medication details of the Unit and update inventory.

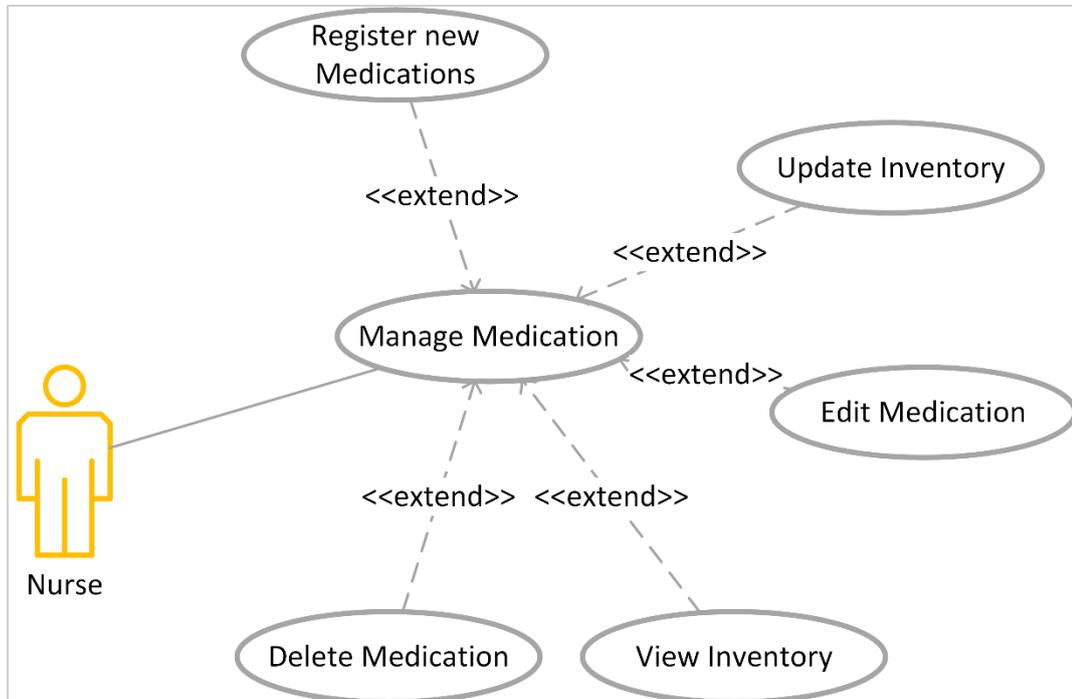


Figure B.3 Use case Diagram – Medication Management

Use case	Manage Medication
Actors	Nurse
Description	Perform medication related activities in the system
Pre-conditions	<ol style="list-style-type: none"> 1. System users should be logged in to the system 2. Medication should not be already exist in the system.
Basic Course of events	<ol style="list-style-type: none"> 1. Check for the availability of the Medication. 2. Register a medication. 3. Specify a re-order level to a specific medication format
Post conditions	<ol style="list-style-type: none"> 1. Medication details can be used for the prescriptions 2. Medication details can be used for report generation. . 3. Drug usage can be calculated.

Table B.3 Use case description – Medication management

Report Module

Daily and monthly reports will be generated using this module. These reports will be helpful to the administration.

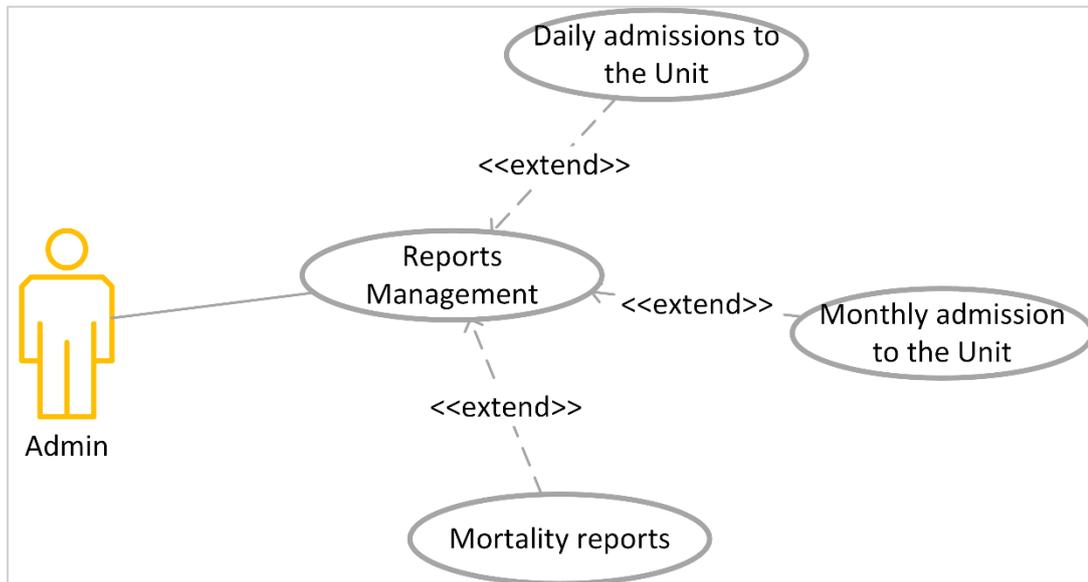


Figure B.4 Use case Diagram – Report generation module

Use case	Reports Managements
Actors	Admin
Description	Report generation
Pre-conditions	
1. System should contain the details which will be needed to generate a report	
Basic Course of events	
1. Specify report type	
2. Specify the time period	
3. Specify the section	
Post conditions	
1. Graphs can be generated according to the reports.	
2. Reports can be printed and converted to PDF or Excel files.	

Table B.4 Use case description – Report Generation

Activity Diagram

An activity diagram is a differences of a state machine in which the states represent the performance of activities and the transitions are triggered by the completion of the activities or sub activities. It represents a flow of activates of a Use case. Activity diagrams are usually drawn for a single Use case.

Login

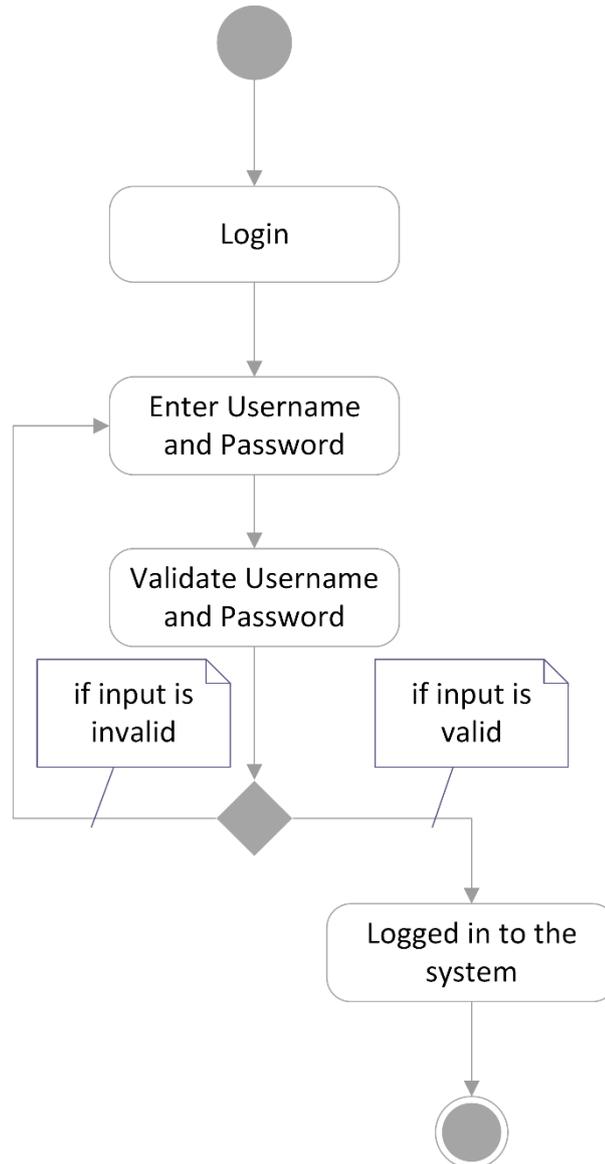


Figure B.5 Activity Diagram – Login

Creating Schedule Plan

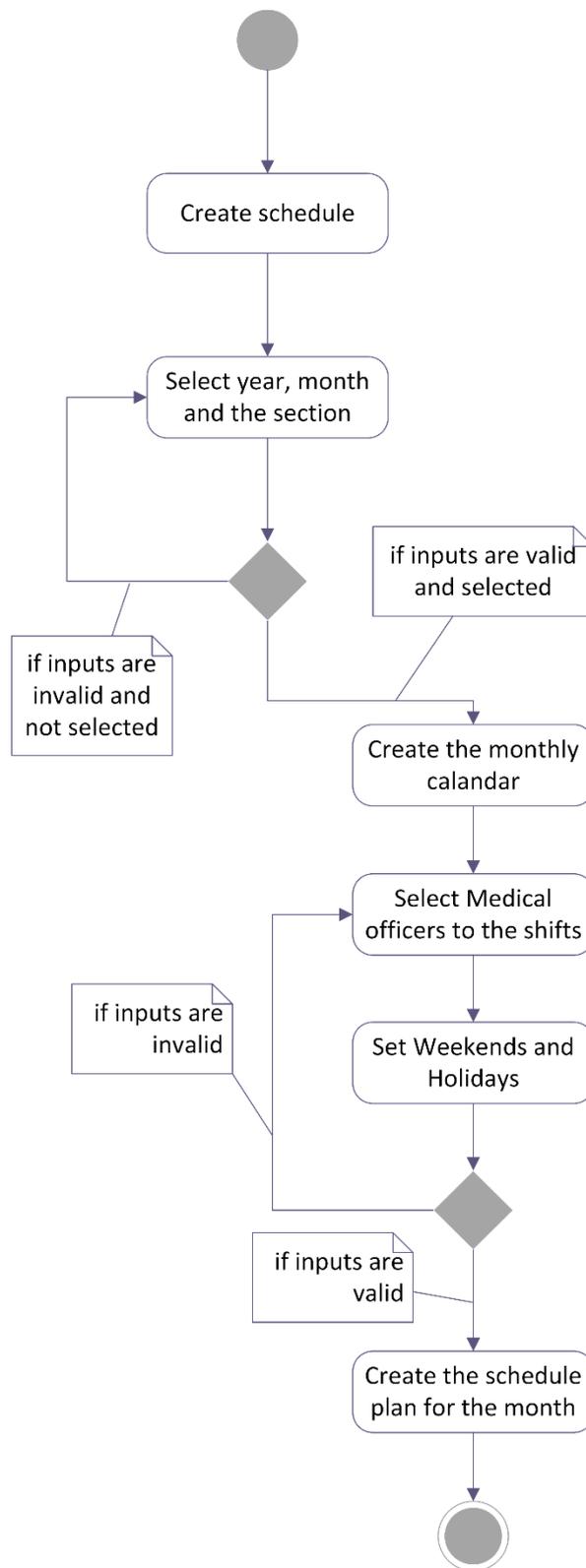


Figure B.6 Activity Diagram – Create schedule plan

Sequence Diagram

Sequence diagrams show how system and user interaction arranged in time sequence. In particular, it shows the instances involved in an interaction by their “lifelines” and the prompt which they arranged in a time sequence. It does not show the associations or connections among the objects.

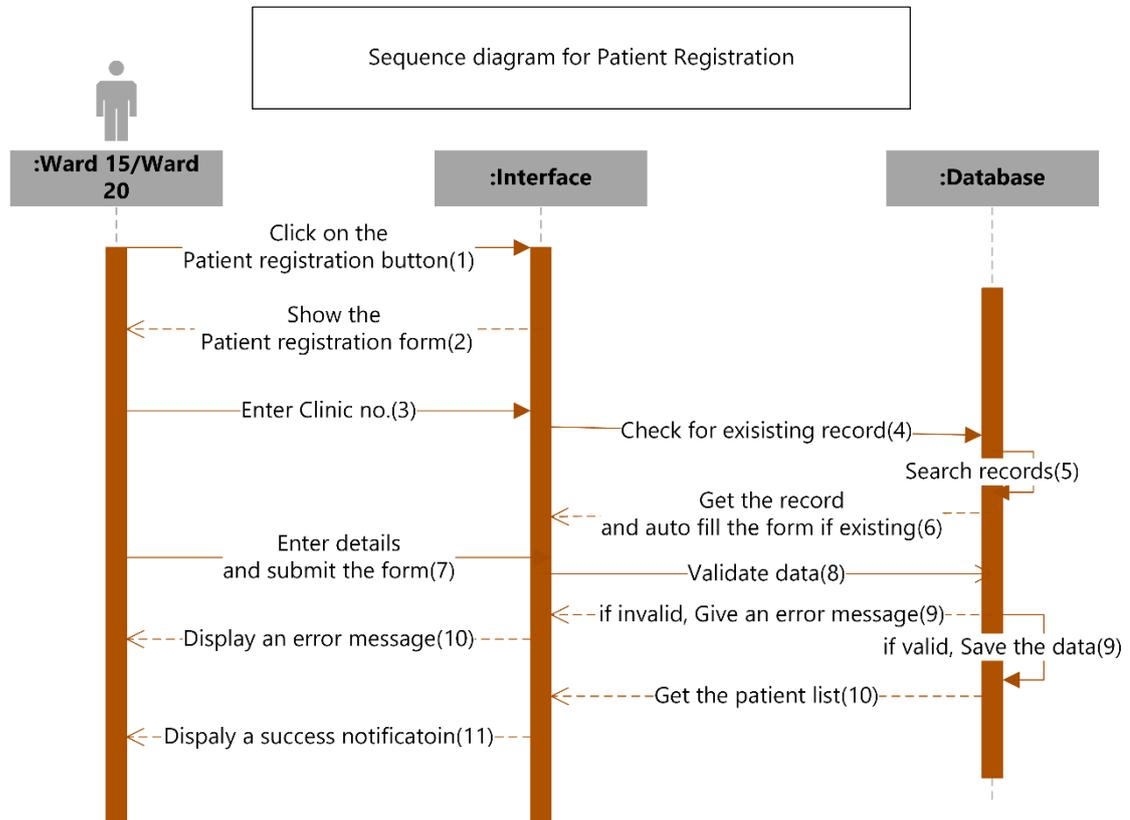


Figure B.7 Sequence Diagram – Patient Registration

Appendix C – User Documentation

Technical documentation are intend for those who are with a technical knowledge to understand the system whereas user documentation is intended to make the end user knowledgeable of how to perform the functions of the system and to get an understanding about the system to work with it. Identifying all the features of the system and how to use these functions and features efficiently is essential need of the user. Users who have privileges to access the system are guided and helped about how to navigate within the system and how to use the functions of the system by this document using screenshots and steps to perform tasks.

System login

Open the web browser that configured in the installation. Type the URL <http://localhost/cticu/> and navigate. After navigation user can visit the login page which is shown in figure C.1. User can only log in to the system by providing the correct username and password. Otherwise user will not be allowed to enter the system.

Cardiothoracic Anaesthesia and Critical Care Unit Lady Ridgeway Hospital



System Login

Username

Password

[Forgot password?](#)

Sign in

Figure C.1 Login Page

With a successful login to the system user will be directed to the dashboard which give the general information about the current states of the unit showed in Figure C.2

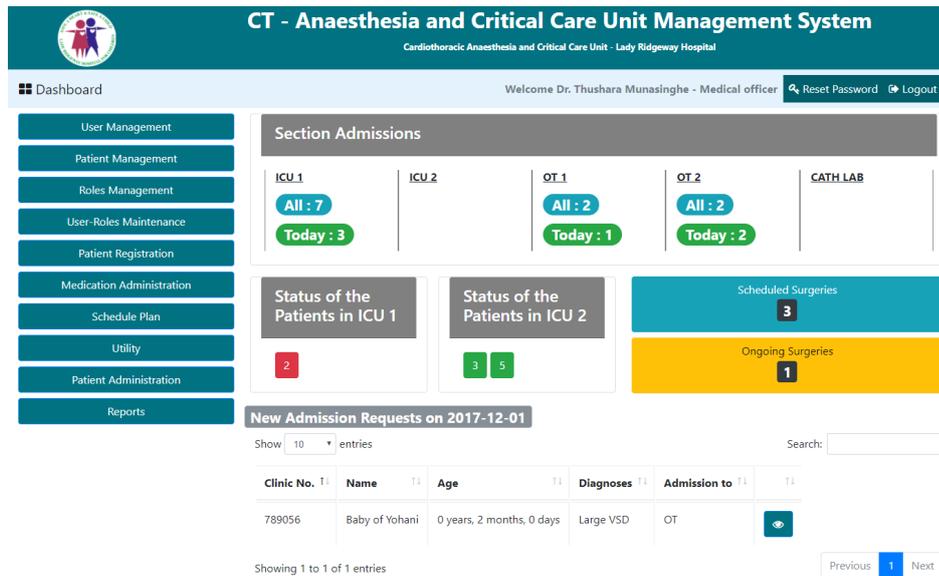


Figure C.2 Dashboard

Navigation bar

Link to the dashboard, Login information of the current user and log out button will be provided by the navigation bar. Shown in Figure C.3.



Figure C.3 Navigation bar

Side Menu

Side menu of the system for a specific user will be dynamically created according to the assigned roles of that user. Side menu of the Admin is showed in Figure C.4

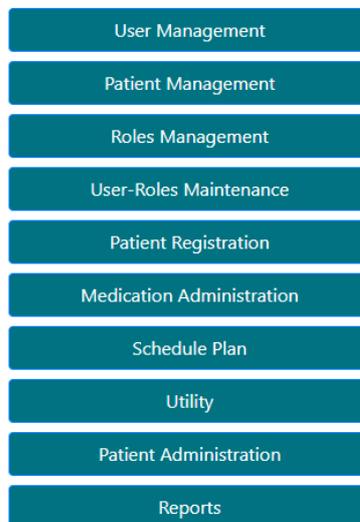
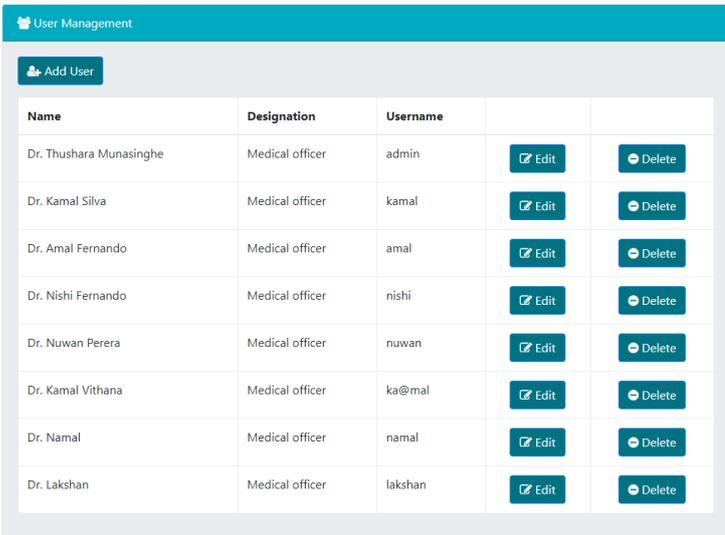


Figure C.4 Side Menu

User Management

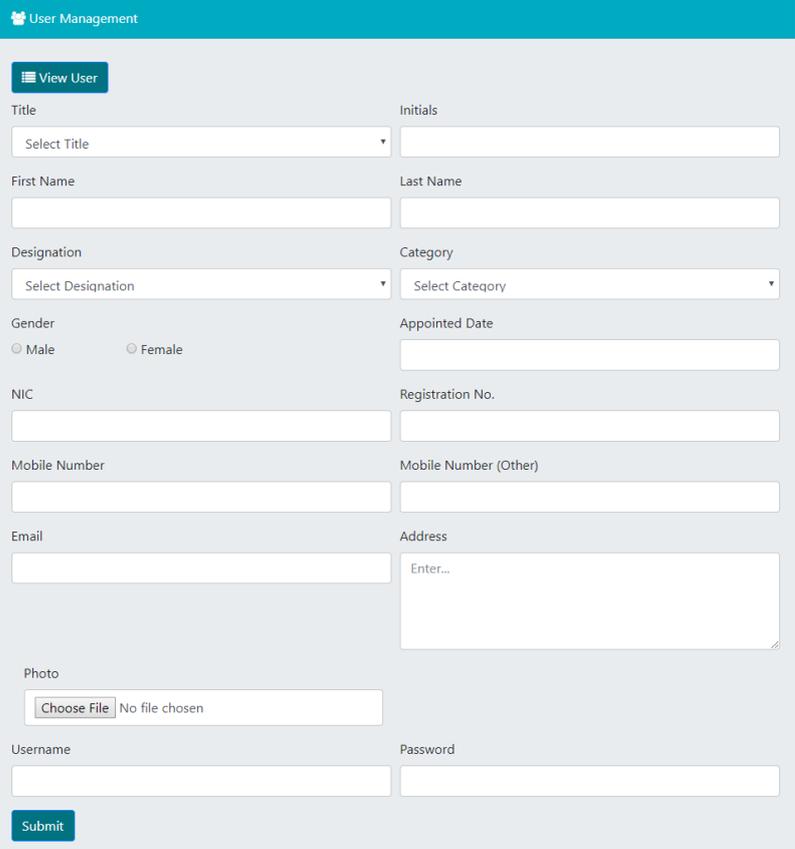
Data manipulating table can be seen in this page and Admin can register a new user to the system, edit information of a user and delete user can be done here by clicking “Add User”, “Edit”, “Delete” buttons respectively. (Figure C.5). A user can be added to the system after inserting relevant information of a user through the form showed in Figure C.6.



The screenshot shows a 'User Management' interface with a teal header and a grey sidebar containing an 'Add User' button. The main content area features a table with the following data:

Name	Designation	Username	Edit	Delete
Dr. Thushara Munasinghe	Medical officer	admin	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete
Dr. Kamal Silva	Medical officer	kamal	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete
Dr. Amal Fernando	Medical officer	amal	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete
Dr. Nishi Fernando	Medical officer	nishi	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete
Dr. Nuwan Perera	Medical officer	nuwan	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete
Dr. Kamal Vithana	Medical officer	ka@mal	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete
Dr. Namal	Medical officer	namal	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete
Dr. Lakshan	Medical officer	lakshan	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete

Figure C.5 User management screen



The screenshot shows a 'User Management' interface with a teal header and a grey sidebar containing a 'View User' button. The main content area features a registration form with the following fields:

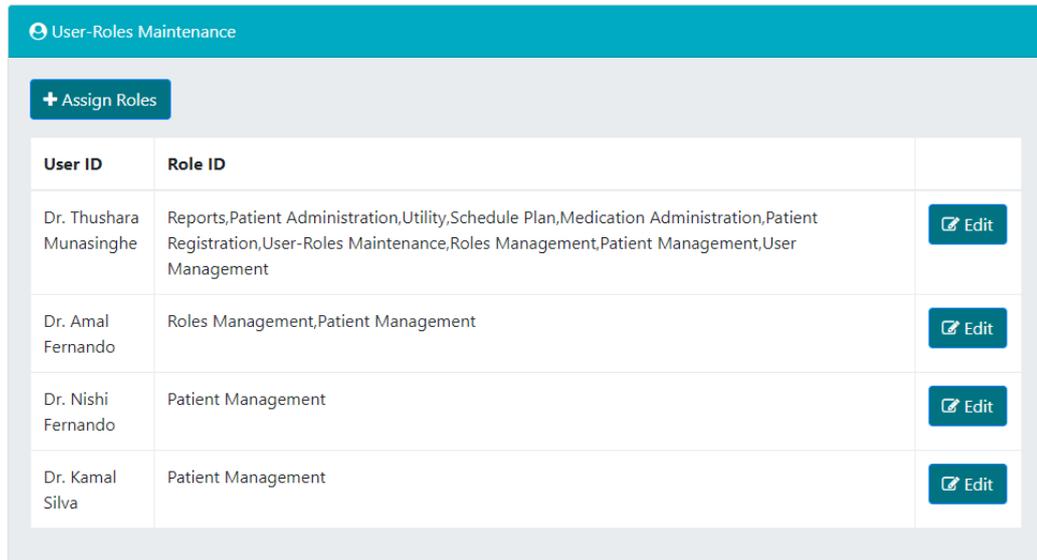
- Title: Select Title (dropdown)
- Initials: Text input
- First Name: Text input
- Last Name: Text input
- Designation: Select Designation (dropdown)
- Category: Select Category (dropdown)
- Gender: Radio buttons for Male and Female
- Appointed Date: Text input
- NIC: Text input
- Registration No.: Text input
- Mobile Number: Text input
- Mobile Number (Other): Text input
- Email: Text input
- Address: Text area (placeholder: Enter...)
- Photo: Choose File button (No file chosen)
- Username: Text input
- Password: Text input

A Submit button is located at the bottom left of the form.

Figure C.6 User Registration Form

User-Role Maintenance

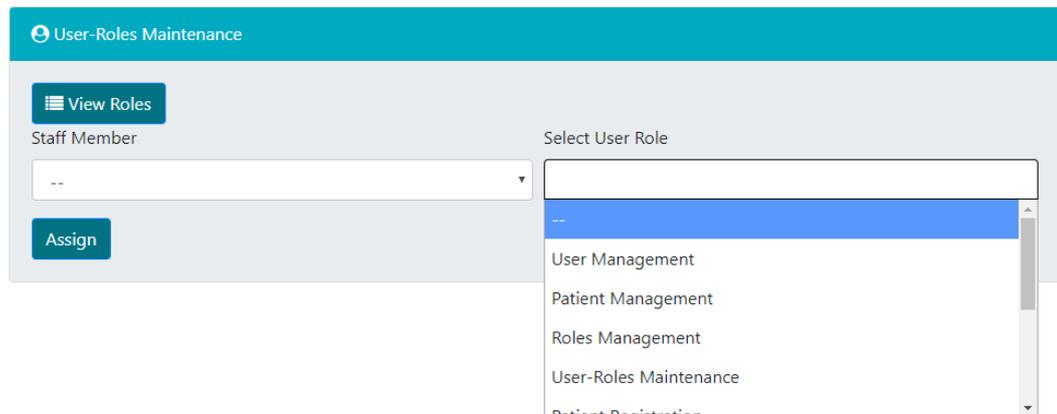
One of the most important topic which comes with the system is user-role maintenance. (Figure C.7) Assigning roles to a specific user is done here by clicking the “Assign Roles” button and it will be directed to the form which is used to assign roles. (Figure C.8)



The screenshot shows the 'User-Roles Maintenance' interface. At the top, there is a teal header with the title 'User-Roles Maintenance' and a back arrow icon. Below the header is a light gray panel containing a '+ Assign Roles' button. The main area is a table with three columns: 'User ID', 'Role ID', and an 'Edit' button. The table lists four users and their assigned roles.

User ID	Role ID	
Dr. Thushara Munasinghe	Reports, Patient Administration, Utility, Schedule Plan, Medication Administration, Patient Registration, User-Roles Maintenance, Roles Management, Patient Management, User Management	
Dr. Amal Fernando	Roles Management, Patient Management	
Dr. Nishi Fernando	Patient Management	
Dr. Kamal Silva	Patient Management	

Figure C.7 User-role Maintenance screen



The screenshot shows the 'Assign Roles' form. It has a teal header with the title 'User-Roles Maintenance' and a back arrow icon. Below the header is a light gray panel containing a 'View Roles' button. The form has a 'Staff Member' dropdown menu with a value of '--' and an 'Assign' button. To the right, there is a 'Select User Role' dropdown menu with a list of roles: '--', 'User Management', 'Patient Management', 'Roles Management', 'User-Roles Maintenance', and 'Patient Registration'.

Figure C.8 Assign Roles Form

Patient Registration

By providing the compulsory information of the patient he or she can be registered to the system.

Patient Registration

Clinic No. BHT No. Admission to Select Ward

Diagnoses Previous Medical and Surgical Problems

Personal Information

First Name Last Name Date of Birth

Age(Years): Age(Months): Age(Days):

Gender Male Female Blood Group Weight Kg Height cm

Contact Details

Guardian Name

Address District

Contact Number Land line Contact Number Mobile Email

Information for Research Purpose

Religion Family Income(Monthly)

Rs.

Figure C.9 Patient Registration form

Patient Management Module

Main topic of this system is Patient Management Module. User can navigate through several sections using the section navigation panel. Figure C.10.

Patient Management

- Admission Requests
- ICU 1
- ICU 2
- OT 1
- OT 2
- CATH LAB

Figure C.10 Navigation panel through sections

“Admission Requests” button is used to direct to the screen where it shows all the admission requests to the Unit. (Figure C.11).

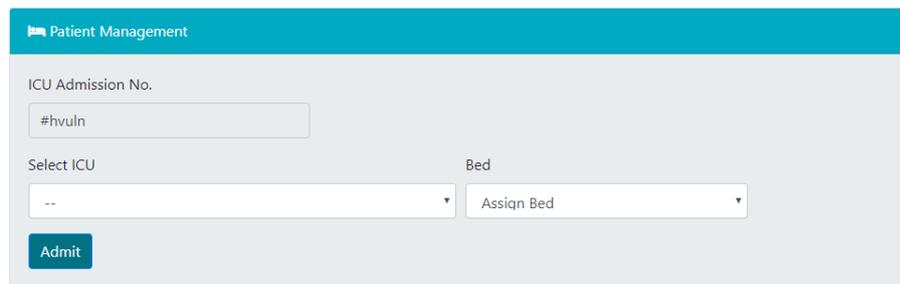


The screenshot shows a 'Patient Management' interface with a teal header. Below the header is a 'Back' button. A table displays admission requests with columns for Clinic No., Name, Age, Gender, Diagnoses, and Admission to. Two rows are visible: one for 'Baby of Nimal' (24556) admitted to ICU, and another for 'Baby of Manel' (12344) admitted to OT.

Clinic No.	Name	Age	Gender	Diagnoses	Admission to
24556	Baby of Nimal	0 years, 3 months, 0 days	male	TGA	ICU
12344	Baby of Manel	0 years, 1 months, 5 days	male	TGA	OT

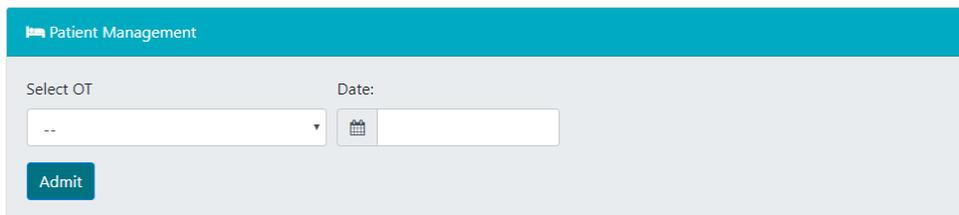
Figure C.11 Admission Requests

By clicking on a request itself it can be allocated to a specific section and assign a bed to the patient. (Figure C.12 and Figure C.13)



The screenshot shows the 'Admission to ICU' form in the 'Patient Management' system. It includes a text input for 'ICU Admission No.' with the value '#hvuln'. Below this are two dropdown menus: 'Select ICU' (currently showing '--') and 'Bed' (currently showing 'Assign Bed'). An 'Admit' button is located at the bottom left.

Figure C.12 Admission to ICU



The screenshot shows the 'Admission to OT' form in the 'Patient Management' system. It features a 'Select OT' dropdown menu (currently showing '--') and a 'Date:' field with a calendar icon. An 'Admit' button is positioned at the bottom left.

Figure C.13 Admission to OT

By clicking on a specific section name user will be directed to that specific section and it will give the full record of all the patient inside a specific ICU. (Figure C.14).

By clicking “Full Profile” (Figure C.15) and “Surgery Details” full record of the patient and their surgery details can be viewed respectively. Patient can be assigned to a surgery by clicking on “Assign to surgery”. Patient details can be edited using the “Edit full profile”

Doctors will be able to create a follow-up record of a patient and all the previously updated records will be visible next to it with the updated user and the time and they will be able to create prescriptions and request Lab reports.

Discharge summary of the patient will be generated by clicking the “Discharge” button.

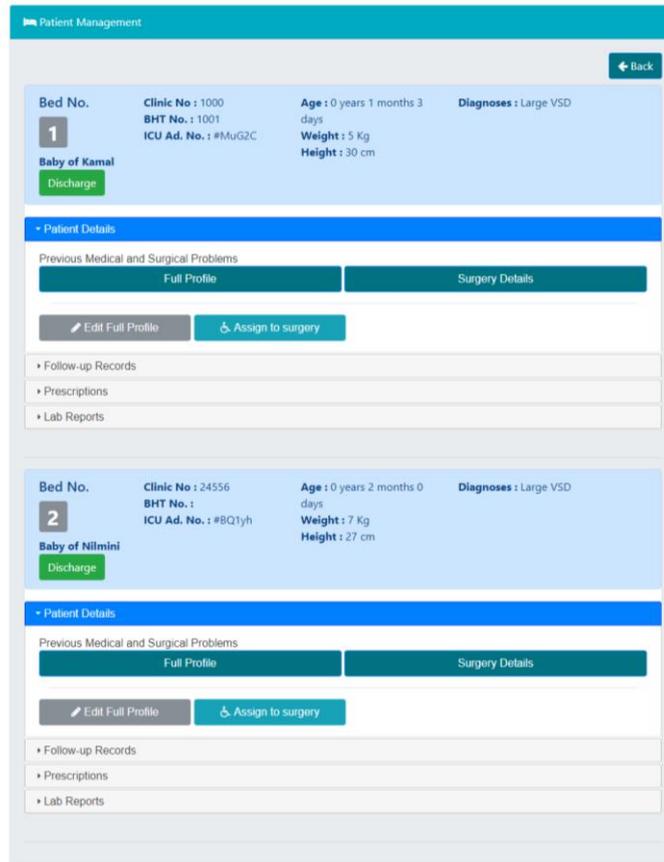


Figure C.14 Patient details inside an ICU

Clinic No.: 24556	BHT No.:
ICU Admission No.: #BQ1yh	
Diagnoses: Large VSD	
Previous Medical and Surgical Problems: Previous Medical and Surgical Problems	
Full Name: Baby of Nilmini	Gender: male
Date of Birth: 2017-10-01	Age (At the time of Admission): 0 years 2 months 0 days
Weight: 7Kg	Height: 27cm
Blood Group: O+	
Guardian Name: Nilmini	Email: nilmin@gmail.com
Address: Address	District: CMD
Contact Number: 1234567	Contact Number(Other):
Religion: Buddhist	Family Income: 1
Close	

Figure C.15 Full Profile of the patient

This shows the surgery plans of a specific unit. Inserting a starting time will change the status of a surgery from ‘Scheduled’ to ‘On-going’ and by inserting an End time to the surgery will change its status from ‘On-going’ to ‘Finished’. (Figure C.16)

The screenshot shows a 'Patient Management' interface with a teal header and a 'Back' button. It displays two surgery entries:

- On-going (Yellow background):**
 - Surgery: R/Upper and Middle Lelectomy
 - Type: Primary
 - Operator: Dr. Amal Fernando
 - Aneesthesiologist: Dr. Amal Fernando
 - Scheduled Date: 2017-10-30
 - Operation Theater: OT 1
 - Start Time: 07:17:36
 - End Time: [Input field]
 - Comments: [Input field]
- Scheduled (Light Blue background):**
 - Surgery: LMBT shunt
 - Type: Secondary
 - Operator: Dr. Amal Fernando
 - Aneesthesiologist: Dr. Amal Fernando
 - Scheduled Date: 2017-11-08
 - Operation Theater: OT 1
 - Start Time: [Input field]

Figure C.16 Details inside a specific OT

Patient Administration

Nurses will be able to create a CBS (Comfort Behavior Scale), Administer Medication by giving a date and time and upload lab reports to inward patients through here. (Figure C.17).

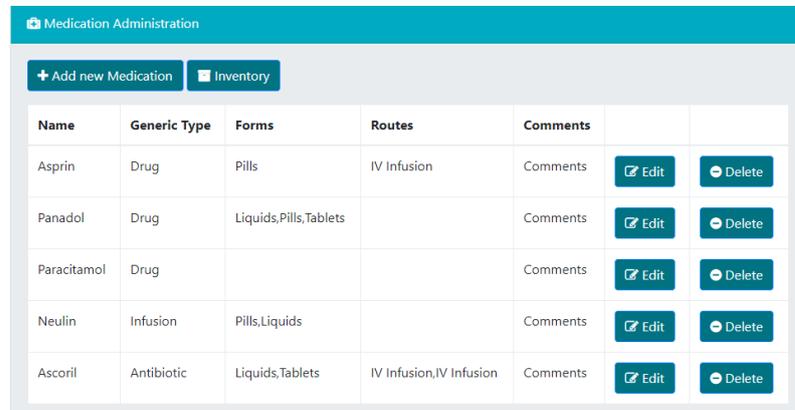
The screenshot shows a 'Patient Administration' interface with a teal header and a 'Back' button. It displays patient information and navigation options:

- Bed No.:** 1
- Clinic No.:** 1000
- BHT No.:** 1001
- ICU Ad. No.:** #MuG2C
- Age:** 0 years 1 months 3 days
- Weight:** 5
- Height:** 30
- Diagnoses:** Large VSD
- Patient Name:** Baby of Kamal
- Navigation:**
 - Previous Medical and Surgical Problems: Full Profile, Surgery Details
 - Comfort Behavior Scale
 - Drug Administration
 - Lab Reports

Figure C.17 Patient Administration screen

Medication Administration

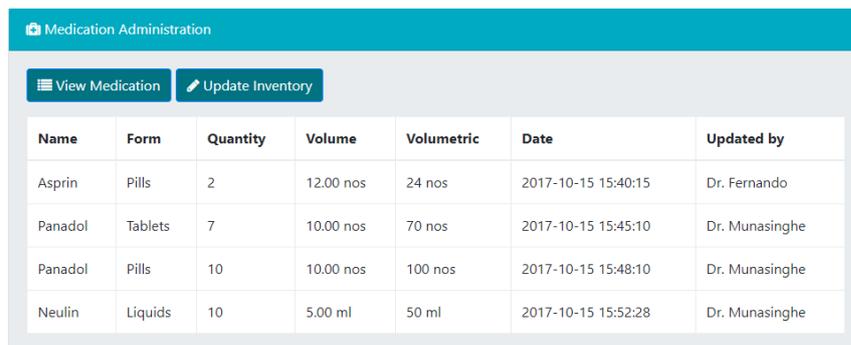
Registration of a new medication, edit details of a medication, delete a medication, viewing Inventory and updating Inventory can be done using “Add new Medication”, “Edit”, “Delete”, “Inventory” and “Update Inventory” buttons. (Figure C.18 and Figure C.19).



The screenshot shows the 'Medication Administration' interface. At the top, there are two buttons: '+ Add new Medication' and 'Inventory'. Below these is a table with the following columns: Name, Generic Type, Forms, Routes, and Comments. Each row represents a medication and includes 'Edit' and 'Delete' buttons.

Name	Generic Type	Forms	Routes	Comments		
Asprin	Drug	Pills	IV Infusion	Comments	Edit	Delete
Panadol	Drug	Liquids,Pills,Tablets		Comments	Edit	Delete
Paracetamol	Drug			Comments	Edit	Delete
Neulin	Infusion	Pills,Liquids		Comments	Edit	Delete
Ascoril	Antibiotic	Liquids,Tablets	IV Infusion,IV Infusion	Comments	Edit	Delete

Figure C.18 Medication Management screen



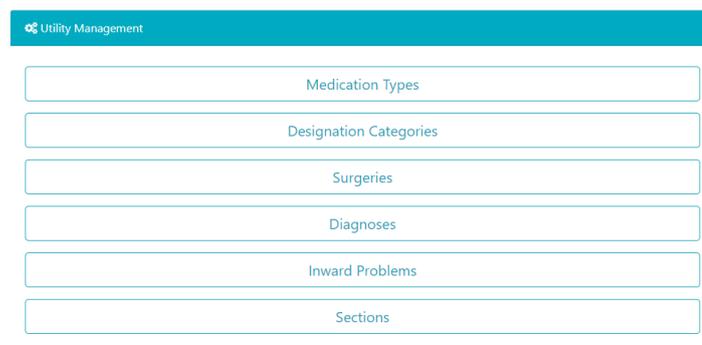
The screenshot shows the 'Medication Administration' interface with the 'Inventory' button selected. Below the navigation bar are two buttons: 'View Medication' and 'Update Inventory'. Below these is a table with the following columns: Name, Form, Quantity, Volume, Volumetric, Date, and Updated by.

Name	Form	Quantity	Volume	Volumetric	Date	Updated by
Asprin	Pills	2	12.00 nos	24 nos	2017-10-15 15:40:15	Dr. Fernando
Panadol	Tablets	7	10.00 nos	70 nos	2017-10-15 15:45:10	Dr. Munasinghe
Panadol	Pills	10	10.00 nos	100 nos	2017-10-15 15:48:10	Dr. Munasinghe
Neulin	Liquids	10	5.00 ml	50 ml	2017-10-15 15:52:28	Dr. Munasinghe

Figure C.19 Inventory of Medication

Utility

Entering system information to the system will be done through here. Useful forms will be provided in each section to add, edit details. (Figure C.20)



The screenshot shows the 'Utility Management' interface. It features a list of utility items, each in a separate box:

- Medication Types
- Designation Categories
- Surgeries
- Diagnoses
- Inward Problems
- Sections

Figure C.20 Utility Management item navigation

Schedule Plan

Monthly schedule plan (Roster) can be created and assign Medical officers to shifts can be done using the Figure showed below. (Figure C.21)

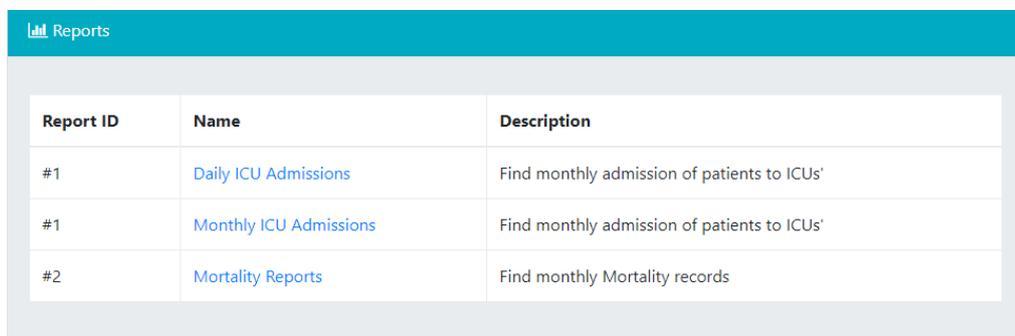


Screenshot of the Schedule Plan form. The form has a teal header with a calendar icon and the text "Schedule Plan". Below the header, there are three dropdown menus: "2017", "April", and "ICU 1". To the right of these dropdowns is a blue button labeled "Process".

Figure C.21 Schedule generation form

Report Generation

This screen can be used to generate useful reports for the administration. (Figure C.22)



Screenshot of the Reports screen. The screen has a teal header with a bar chart icon and the text "Reports". Below the header is a table with three columns: "Report ID", "Name", and "Description". The table contains three rows of data.

Report ID	Name	Description
#1	Daily ICU Admissions	Find monthly admission of patients to ICUs'
#1	Monthly ICU Admissions	Find monthly admission of patients to ICUs'
#2	Mortality Reports	Find monthly Mortality records

Figure C.22 Inventory of Medication

Appendix D - Management Reports

Mortality Reports

Mortality reports can be generated using the discharge status of a discharged patients and Management can track the percentages of Mortality within a given time period through these reports. Mortality records can be compared using the monthly percentages or yearly percentages.

Unit Admissions

Admissions to the Unit can be filter out using the District of the admission was made, to which section the admission was made.

Count of the patients admitted to a specific section within a given time period can be tracked using this report. (Figure D.2)

#	Admission Date	Admission To	Number of Admissions
1	2017-11-03	ICU	2
2	2017-11-03	OT	1
3	2017-11-04	ICU	1
4	2017-11-05	OT	2

Figure D.1 All admissions to the different sections

#	Admission Date	Admission To	Number of Admissions
1	2017-11-04	ICU	1
2	2017-11-05	OT	2

Figure D.2 All admissions to the different sections within a given time period

Reports

From To ICU

#	Admission Date	Admission To	Number of Admissions
1	2017-11-03	ICU	2
2	2017-11-04	ICU	1

Figure D.3 All admissions to a specific section within a given time period

Drug Usage

Monthly or yearly drug usage can be identified by these reports. Records can be filtered using the date and the sections. Graphs can be generated with the records which will be really helpful to compare records with each other.

Appendix E – Test Results

User Authentication Test results

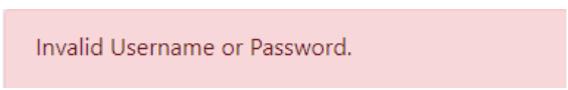
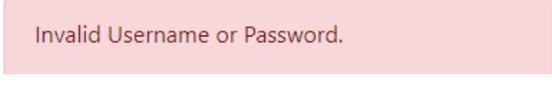
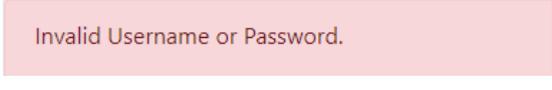
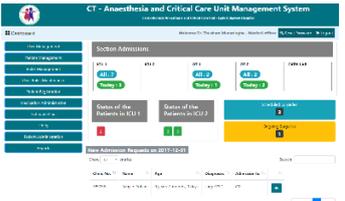
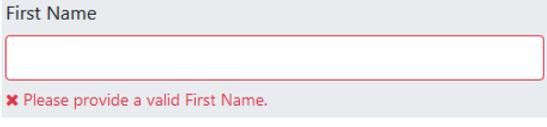
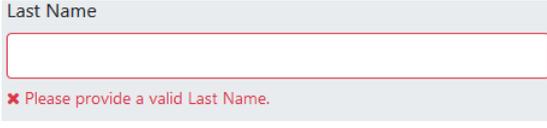
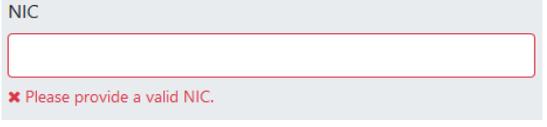
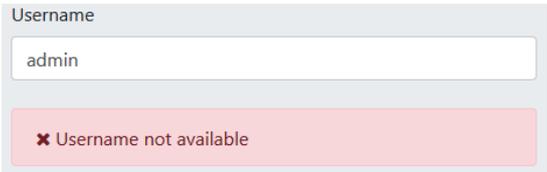
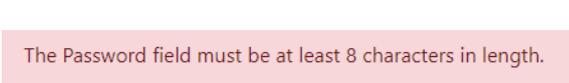
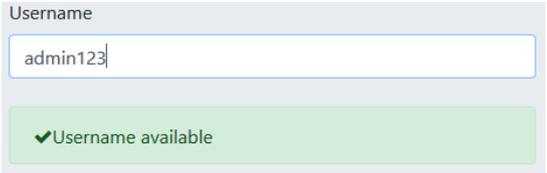
Test No	Steps of the Test	Expected Results	Actual Results	Status
01	Try to log in to the system by keeping all the fields empty	Display an error message		Pass
02	Enter correct username Enter incorrect password	Display an error message		Pass
03	Enter incorrect username Enter correct password	Display an error message		Pass
04	Enter Incorrect username Enter incorrect password	Display an error message		Pass
05	Enter correct username Enter correct password	Successful login to the system. Direct to the dashboard		Pass

Table E.1 User Authentication test results

User Management Test Results

Test No	Steps of the Test	Expected Results	Actual Results	Status
01	Try to submit a form by keeping all the fields empty	Display error messages		Pass

02	Try to submit a form by keeping Title field empty	Display an error message		Pass
03	Try to submit a form by keeping First name field empty	Display an error message		Pass
04	Try to submit a form by keeping Last Name field empty	Display an error message		Pass
05	Try to submit a form by keeping Designation field empty.	Display an error message		Pass
06	Try to submit a form by keeping NIC field empty.	Display an error message		Pass
07	Try to enter an existing Username	Display an error message		Pass
08	Try to enter a password with invalid format.	Display an error message		Pass
09	Enter a non-existing Username	Display a successful message		Pass
10	Submit the form by filling all the required fields	Display a successful notification and re-direct to the User list.		Pass

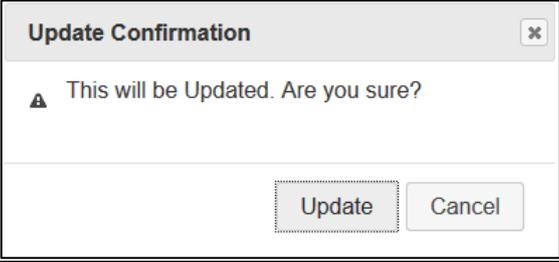
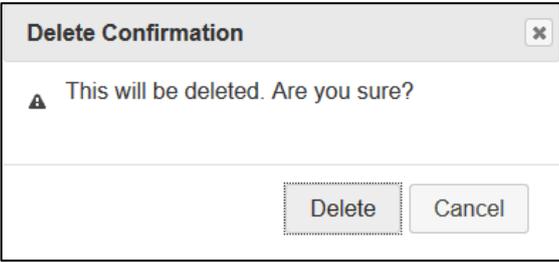
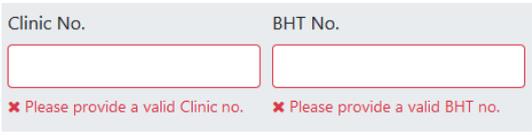
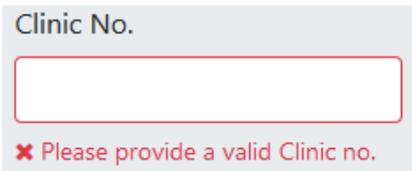
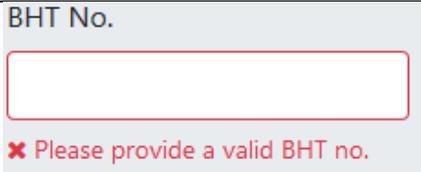
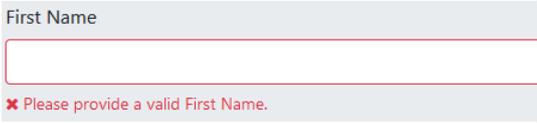
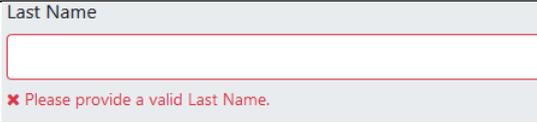
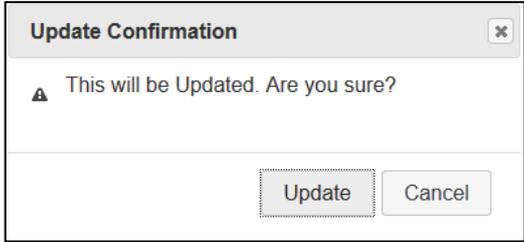
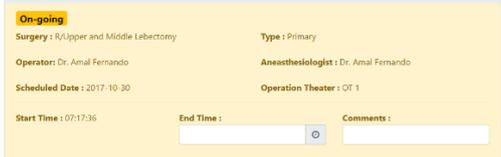
11	Try to update user details	Display a confirmation box		Pass
12	Update user details successfully	Display a successful notification and re-direct to the User list		Pass
13	Try to delete a user	Display a confirmation box		Pass
14	Successfully delete a user	Display a successful notification and re-direct to the User list		Pass

Table E.2 User Management Test results

Patient Management Test Results

Test No	Steps of the Test	Expected Results	Actual Results	Status
01	Try to submit a form by keeping all the fields empty	Display error messages		Pass
02	Try to submit a form by keeping Clinic No. field empty	Display an error message		Pass
03	Try to submit a form by keeping	Display an error message		Pass

	BHT No. field empty			
04	Try to submit a form by keeping First name field empty	Display an error message		Pass
05	Try to submit a form by keeping Last name field empty	Display an error message		Pass
06	Submit the form by filling all the required fields	Display a successful notification and re-direct to the Patient list.		Pass
07	Try to update patient details	Display a confirmation box		Pass
08	Update patient details successfully	Display a successful notification and re-direct to the Patient list.		Pass
09	Update the time of the scheduled surgery	Change the status of the surgery to "On-Going"		Pass
10	Try to create a prescription keeping required fields empty	Display error messages		Pass

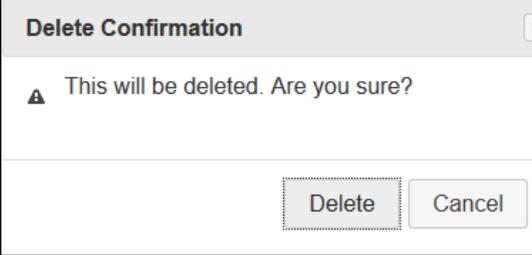
			Unit(Measure): <input type="text" value="Select"/> Duration (Days): <input type="text"/>																																					
			✖ Please provide a valid Measuring unit ✖ Please provide a valid Duration																																					
11	Submit the prescription note by filling all the required fields	Display a successful notification and show the updated prescription list.	 <table border="1" data-bbox="735 504 1267 683"> <thead> <tr> <th>Medication</th> <th>Dose</th> <th>Duration</th> <th>Started Date</th> <th>Last Administration</th> <th></th> </tr> </thead> <tbody> <tr> <td>Asprin</td> <td>3g</td> <td>1 days</td> <td></td> <td></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Asprin</td> <td>3ml</td> <td>3 days</td> <td></td> <td></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Asprin</td> <td>5g</td> <td>4 days</td> <td></td> <td></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Asprin</td> <td>4ml</td> <td>3 days</td> <td></td> <td></td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td>Panadol</td> <td>5g</td> <td>3 days</td> <td></td> <td></td> <td><input type="button" value="Delete"/></td> </tr> </tbody> </table>	Medication	Dose	Duration	Started Date	Last Administration		Asprin	3g	1 days			<input type="button" value="Delete"/>	Asprin	3ml	3 days			<input type="button" value="Delete"/>	Asprin	5g	4 days			<input type="button" value="Delete"/>	Asprin	4ml	3 days			<input type="button" value="Delete"/>	Panadol	5g	3 days			<input type="button" value="Delete"/>	Pass
Medication	Dose	Duration	Started Date	Last Administration																																				
Asprin	3g	1 days			<input type="button" value="Delete"/>																																			
Asprin	3ml	3 days			<input type="button" value="Delete"/>																																			
Asprin	5g	4 days			<input type="button" value="Delete"/>																																			
Asprin	4ml	3 days			<input type="button" value="Delete"/>																																			
Panadol	5g	3 days			<input type="button" value="Delete"/>																																			
12	Try to delete a prescription	Display a confirmation box		Pass																																				
13	Successfully delete a prescription note.	Display a successful notification and display the updated prescription list.		Pass																																				
14	Successfully update a follow-up record.	Display the updated user and the time.		Pass																																				

Table E.3 Patient Management Test results

Patient Administration Test results

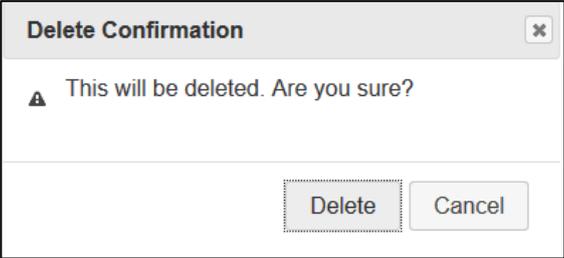
Test No	Steps of the Test	Expected results	Actual Results	Status
01	Try to submit a CBS record without filling all the fields	Display an error message		Pass
02	Try to administer a medication without filling the date and time field.	Display error messages		Pass
03	Successfully administer a medication	Display a successful alert.		Pass
04	Try to delete a uploaded lab report	Display a confirmation box		Pass
05	Successfully upload a Lab Report	Display a successful alert.		Pass
06	Successfully delete a Lab Report	Display a successful notification and display the updated lab reports list.		Pass

Table E.4 Patient Administration test results

Medication Management Test results.

Test No	Steps of the Test	Expected results	Actual Results	Status
01	Try to submit a form by keeping all	Display error messages		Pass

	the fields empty																																						
02	Try to submit a Medication registration form by keeping Name field empty	Display an error message		Pass																																			
03	Try to submit a form by keeping Generic type field empty	Display an error message		Pass																																			
04	Submit the form by filling all the required fields	Display a successful notification and redirect to the Medication list.		Pass																																			
05	Try to update the inventory by keeping all the fields empty	Display error messages		Pass																																			
06	Update the inventory by filling all the required fields	Display a successful notification and redirect to the Inventory	<table border="1"> <thead> <tr> <th>Name</th> <th>Form</th> <th>Quantity</th> <th>Volume</th> <th>Volumetric</th> <th>Date</th> <th>Updated by</th> </tr> </thead> <tbody> <tr> <td>Asprin</td> <td>Pills</td> <td>2</td> <td>12.00 nos</td> <td>24 nos</td> <td>2017-10-15 15:40:15</td> <td>Dr. Fernando</td> </tr> <tr> <td>Panadol</td> <td>Tablets</td> <td>7</td> <td>10.00 nos</td> <td>70 nos</td> <td>2017-10-15 15:45:10</td> <td>Dr. Munasinghe</td> </tr> <tr> <td>Panadol</td> <td>Pills</td> <td>10</td> <td>10.00 nos</td> <td>100 nos</td> <td>2017-10-15 15:48:10</td> <td>Dr. Munasinghe</td> </tr> <tr> <td>Neulin</td> <td>Liquids</td> <td>10</td> <td>5.00 ml</td> <td>50 ml</td> <td>2017-10-15 15:52:28</td> <td>Dr. Munasinghe</td> </tr> </tbody> </table>	Name	Form	Quantity	Volume	Volumetric	Date	Updated by	Asprin	Pills	2	12.00 nos	24 nos	2017-10-15 15:40:15	Dr. Fernando	Panadol	Tablets	7	10.00 nos	70 nos	2017-10-15 15:45:10	Dr. Munasinghe	Panadol	Pills	10	10.00 nos	100 nos	2017-10-15 15:48:10	Dr. Munasinghe	Neulin	Liquids	10	5.00 ml	50 ml	2017-10-15 15:52:28	Dr. Munasinghe	Pass
Name	Form	Quantity	Volume	Volumetric	Date	Updated by																																	
Asprin	Pills	2	12.00 nos	24 nos	2017-10-15 15:40:15	Dr. Fernando																																	
Panadol	Tablets	7	10.00 nos	70 nos	2017-10-15 15:45:10	Dr. Munasinghe																																	
Panadol	Pills	10	10.00 nos	100 nos	2017-10-15 15:48:10	Dr. Munasinghe																																	
Neulin	Liquids	10	5.00 ml	50 ml	2017-10-15 15:52:28	Dr. Munasinghe																																	

Table E.5 Medication Management test results

Schedule Plan Test Results

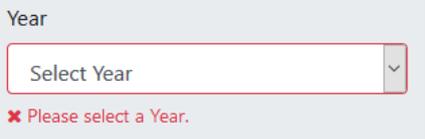
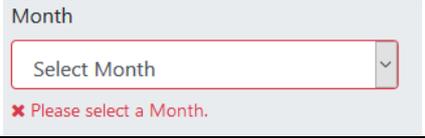
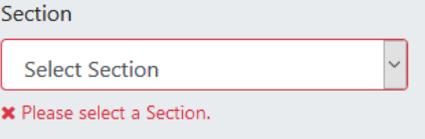
Test No	Steps of the Test	Expected results	Actual Results	Status
	Try to create a schedule without selecting a Year	Display an error message		Pass
	Try to create a schedule without selecting a Month	Display an error message		Pass
	Try to create a schedule without selecting Section	Display an error message		Pass
	Creating the schedule by filling all the required fields.	Display the colander for selected month with the shifts of the specific section		Pass
	Successfully assigning doctors to shifts.	Display a successful notification and show the created schedule plan		Pass

Table E.6 Schedule plan test results

Appendix F - Code Listing

Please refer the CD-ROM for the complete set of codes for the all code fragments.

Following code snippets are not showing full codes because of the space limit.

- User login.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class User extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->model('user_model');
        $this->load->model('category_model');
        $this->load->model('designation_model'); //Calling all the models which are used in this
        controller.
    }
    public function index() {
        $this->load->view('login'); //Default function (calls it self whenever the function is not
        specified.)
    }
    public function dashboard() {
        if ($this->session->userdata('logged_info') == FALSE) {
            redirect();
        }
        $this->load->view('dashboard');
    }
}
```

- Validating Username and Password.

```
//Validating the Username and Password in the login form.

public function validate() {

    $this->form_validation->set_rules('user_name', 'Username', 'trim|required');

    $this->form_validation->set_rules('password', 'Password',
'trim|required|callback_authenticate'); // "Callback" is used to call a function defined in the
controller to use as a rule.

    if ($this->form_validation->run() == FALSE) {

        $this->load->view('login');

    } else {

        $this->load->view('dashboard');

    }

}

//Get data from the database and create a session with logging info and user roles. If not, give
an error message.

public function authenticate($password) {

    $username = $this->input->post('user_name');

    $result = $this->user_model->validate($username, $password);

    if ($result) {

        $this->session->set_userdata('logged_info', $result); //Creating a session with
logging information

        $roles = $this->user_model->get_user_roles($result[0]->user_id);

        $this->session->set_userdata('roles', $roles);

        return TRUE;

    } else {

        $this->form_validation->set_message('authenticate', 'Invalid Username or Password.');
```

//Error message

```
        return false;

    }

}
```

- Code snippet for Check username existence

```
//Check for the username existence.

public function username($username) {
    $result = $this->user_model->username($username);
    if ($result) {
        $this->form_validation->set_message('username', 'Username already exists. ');
        return FALSE;
    } else {
        return TRUE;
    }
}

public function validate_username() {
    $check = $this->username($this->input->post('username'));
    if ($check) {
        echo 'Username available';
    } else {
        echo 'Username not available';
    }
}
```

- Code snippet for logout user

```
//Destroying the session created for the specific user and redirect to default page(login page)

public function logout_user() {
    if ($this->session->userdata('logged_info') == FALSE) {
        redirect();
    }

    $this->session->unset_userdata('logged_info'); //Specifying the user, by using session
data stored in session called "logged info".

    $this->session->sess_destroy();

    redirect("", 'refresh');
}
```

- Code snippet for generating code for ICU Admission No

```

<?php
class Patient_model extends CI_Model {
    public function __construct() {
        parent::__construct();
    }

    public function generate_icu_adm_no() {
        $seed = str_split('abcdefghijklmnopqrstuvwxyz'
            . 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
            . '0123456789');

        shuffle($seed);

        $rand = "";

        foreach (array_rand($seed, 5) as $k)
            $rand .= $seed[$k];

        if ($this->check_icu_adm_no('#' . $rand)) {
            return '#' . $rand;
        } else {
            $this->generate_icu_adm_no();
        }
    }

    public function check_icu_adm_no($icu_no = null) {
        $this->db->select('tbl_admission.*');
        $this->db->from('tbl_admission');
        if (!empty($icu_no)) {
            $this->db->where('icu_adm_no', $icu_no);
        }
        $query = $this->db->get();
        //echo $this->db->last_query();
        if ($query->num_rows() > 0) {
            return FALSE;
        } else {
            return TRUE;
        }
    }
}

```

- Code snippet for User-roles Maintenance

```
class User_roles extends CI_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->model('user_roles_model');
        $this->load->model('user_model');
        $this->load->model('roles_model');
    }
    public function index() {
        $this->load->view('login');
    }
    public function maintain() {
        $data['list'] = $this->user_roles_model->get_roles();
        $this->load->view('user_roles_maintenance/manage', $data);
    }
    public function list_user_roles() {
        $data['list'] = $this->user_roles_model->get_roles();
        $this->load->view('user_roles_maintenance/list', $data);
    }
    public function assign_roles() {
        $data['users']=$this->user_model->get_user();
        $data['roles']=$this->roles_model->get_roles();
        $this->load->view('user_roles_maintenance/assign', $data);
    }
}
```

- JavaScript code snippet for load content

```
function load_content(path, location) {  
    var dataString = "";  
    $.ajax({  
        type: 'POST',  
        data: dataString,  
        url: path,  
        success: function (data) {  
            $("#" + location).html(data);  
        },  
        error: function (request, status, error) {  
            alert(request.responseText);  
        }  
    });  
}
```

- JavaScript code snippet for save form inputs

```
function save(form_id, path, location) {  
    var form = document.getElementById(form_id);  
  
    if (form.checkValidity() === false) {  
        form.classList.add('was-validated');  
    } else {  
        var dataString = $("#" + form_id).serialize();  
        $.ajax({  
            type: 'POST',  
            data: dataString,  
            url: path,  
            success: function (data) {  
                $("#" + form_id)[0].reset();  
                $("#" + location).html(data);  
            },  
        });  
    }  
}
```

```

error: function (request, status, error) {
    alert(request.responseText);
}
});
}
}

function save_multipart(form_id, path, location) {
    var form = document.getElementById(form_id);
    if (form.checkValidity() === false) {
        form.classList.add('was-validated');
    } else {
        var dataString = new FormData(document.getElementById(form_id)); //FormData is a
in built class in the browser.

        var photo = document.getElementById('photo').files[0];

        dataString.append('photo', photo);

        $.ajax({
            type: 'POST',
            data: dataString,
            cache: false,
            contentType: false,
            processData: false,
            url: path,
            success: function (data) {
                $("##" + location).html(data);
            },
            error: function (request, status, error) {
                alert(request.responseText);
            }
        });
    }
}
}

```

- JavaScript code snippet for assign officers to the roster.

```
function assign_officer(id, day, session, path) {
    var dataString = "s_year=" + $("#year").val() + "&";
    dataString += "s_month=" + $("#month").val() + "&";
    dataString += "s_day=" + day + "&";
    dataString += "s_session=" + session + "&";
    dataString += "s_section=" + $("#section").val() + "&";
    $.ajax({
        type: 'POST',
        data: dataString,
        url: path,
        cache: false,
        dataType: 'json',
        success: function (data) {
            $('#officer').empty();
            $.each(data, function (key, val) {
                $('#officer').append('<option value = "' + val.user_id + "' ' + val.sel + '>' +
val.fname + ' ' + val.lname + '</option>');
            });
            $('#officers').modal('show');
            $('#save_button').unbind().click(function () { //unbind - removing the previous data
binded with the model dialog box.
            var dataString = "; dataString += "s_year=" + $("#year").val() + "&";dataString +=
"s_month=" + $("#month").val() + "&";
            dataString += "s_day=" + day + "&";dataString += "s_session=" + session +
"&";dataString += "s_section=" + $("#section").val() + "&";dataString += "s_officer=" +
$("#officer").val() + "&";
            $.ajax({
                type: 'POST',
                data: dataString,
                url: '<?php echo site_url('schedule/save_assigned_officers/'); ?>',
                cache: false,
                dataType: 'json',
                success: function (data) {
```

```

        var r = data;
        if (id != "") {
            $('# + id).html(r);
            $('#officers').modal('hide');
            $('#officers').data('modal', null) //Making the modal empty, removing the data.
            id = "";
        },
        error: function (request, status, error) {
            alert(error);
        });});},
    error: function (request, status, error) {
        alert(error);
    }
});}

```

- Code snipped for creating a new user.

```

//Loading the form to create a new user.
public function add_user() {
    if ($this->session->userdata('logged_info') == FALSE) {
        redirect();
    }
    $data['designation'] = $this->designation_model->get_designation();
    $data['category'] = $this->category_model->get_catgry();
    $this->load->view('user_management/add', $data);
}
//Saving user information.
public function save_user() {
    if ($this->session->userdata('logged_info') == FALSE) {
        redirect();
    }
}

```

```

$this->form_validation->set_rules('user_name','Username',
'trim|required|min_length[8]|callback_username');//Validating information.

$this->form_validation->set_rules('password', 'Password', 'trim|required');

if ($this->form_validation->run() == FALSE) {
    $data['designation'] = $this->designation_model->get_designation();
    $data['category'] = $this->category_model->get_catgry();
    $this->load->view('user_management/add', $data);
} else {
    $data = array('title' => $this->input->post('title'),
        'initials' => $this->input->post('initials'),
        'fname' => $this->input->post('fname'),
        'lname' => $this->input->post('lname'),
        'designation_id' => $this->input->post('designation_id'),
        'catgry_id' => $this->input->post('ctegry_id'),
        'gender' => $this->input->post('gender'),
        'appoint_date' => $this->input->post('appoint_date'),
        'nic' => $this->input->post('nic'),
        'reg_no' => $this->input->post('reg_no'),
        'contact1' => $this->input->post('contact1'),
        'contact2' => $this->input->post('contact2'),
        'email' => $this->input->post('email'),
        'address' => $this->input->post('address'),
        'user_name' => $this->input->post('user_name'),
        'password' => $this->input->post('password'),
        'photo' => $this->uploadPhoto());

$this->user_model->save($data);

    $data['list'] = $this->user_model->get_user();

    $this->load->view('user_management/list', $data); //Redirecting to the user list after
saving a new user.

    }
}

```

- Uploading a photo

```
function uploadPhoto() {
    if ($this->session->userdata('logged_info') == FALSE) {
        redirect(); }
    $photo = "photo";
    $config['upload_path'] = "./assets/photo";
    $config['allowed_types'] = "gif|jpg|png";
    $config['max_size'] = 1024 * 16;
    $config['encrypt_name'] = false;
    $this->load->library('upload', $config);
    if (!$this->upload->do_upload($photo)) {
        echo $this->upload->display_errors(", ");
    } else {
        $data = $this->upload->data();
        return $data['file_name'];
    }
}
```

- Code snippet for Assign roles.

```
public function assign_roles() {
    $data['users']=$this->user_model->get_user();
    $data['roles']=$this->roles_model->get_roles();
    $this->load->view('user_roles_maintenance/assign', $data);
}
```

```
public function save_roles() {
    $this->form_validation->set_rules('user_id', 'Staff Member', 'trim|required');
    if ($this->form_validation->run() == FALSE) {
        $this->load->view('user_roles_maintenance/assign');
    } else {
        foreach ($this->input->post('role_id') as $key => $value) {
            $data = array('user_id' => $this->input->post('user_id'),
                'role_id' => $value);
            $this->user_roles_model->save($data);
        }
        $data['list'] = $this->user_roles_model->get_roles();
        $this->load->view('user_roles_maintenance/list', $data);
    }
}
```

Appendix G - Client Certificate



Cardiothoracic Anaesthesia and Critical Care Unit
Lady Ridgeway Hospital
Dr. Danister de Silva Mawatha
Colombo 08
Sri Lanka
Tel: +94 11 2 693711

07/11/2017

Project Examination Board,
University of Colombo School of Computing,
No 221/2 A,
Dharmapala Mawatha,
Colombo 07.

Dear Sir/ Madam,

Letter of certification

This is to inform you that Ms. D. M. U. I. Dissanayake (Index no. 1411934) has successfully developed a Web based solution (Web based Cardiothoracic Anaesthesia and Critical Care Unit Management System) for Cardiothoracic Anaesthesia and Critical Care Unit of Lady Ridgeway Hospital.

She had analyzed the situation and the work flow of this Unit well and developed this system which I was pleased to implement. She had observed every aspect of the processes carefully making sure there were no complications.

Performance and reliability of the system proved to be successful and it can be accepted as a standard solution for the Unit.

Sincerely,

Dr. L.A.P. Perera
Senior Consultant Anaesthesiologist (In-charge)
Cardiothoracic Anaesthesia & Critical Care
Lady Ridgeway Hospital



Glossary

- Ajax - Stands for Asynchronous JavaScript and XML.
A collection of web technologies running on client side to develop interactive web applications.
- Apache - Free, open source web server software.
- CSS - Stands for Cascading Style sheets
A styling language that describes the styles on HTML document.
- Database - A collection of related data that is centralized, which can be easily accessed, controlled, updated and manipulated.
- HTML - Stands for Hypertext Markup Language.
Basic markup language for creating web pages.
- PHP - Stands for Hypertext Preprocessor
Server side scripting language for web based development
- SHA1 - Stands for Secure Hash Algorithm.
It is a Cryptographic hash function.
- Use case diagram - This diagram show how the user interact with the system.
- WAMP - Open source Windows web development environment. It includes Apache, My SQL and PHP.

Index

A

Admission. ix, x, 2, 6, 14, 15, 31, 51,
58, 59, 81

Anaesthesia 1, iii, v, 1, 63, 67

C

Cardiothoracic iii, v, 1, 7, 63, 67

CATH Lab xii, 6, 7

Client..... viii, 1, 4, 57, 103

Critical..... 1, iii, v, 1, 15, 63, 67

D

diagnoses..... 6

Dischargeiii, ix, 2, 15, 16, 22, 70, 82

doctor 7, 63, 64, 96

drug 3, 8, 10, 16, 87

F

follow-up..... 2, 7, 15, 59, 63, 70, 81, 92

H

history iii, 2, 6, 7, 14, 15

Hospital.....1, iii, v, vi, xii, 1, 8, 10, 63,
67

I

ICU....ix, x, xii, 6, 7, 14, 15, 32, 51, 81,
82

M

Medical... .. iii, vii, xii, 1, 2, 7, 8, 9, 10,
11, 12, 14, 16, 17, 26, 36, 60, 64, 85

Mortality.....iii, 2, 11, 15, 16, 59, 86

N

Nurse 71

O

Operation.....xii, 6, 7, 32

OTix, x, xii, 6, 7, 33, 81, 83

P

patients .iii, ix, 1, 2, 6, 9, 14, 15, 17, 32,
64, 70, 83, 86

PHPiii, 38, 39, 56, 64, 104

R

reports.iii, vi, 2, 3, 4, 11, 12, 17, 37, 59,
63, 72, 81, 83, 85, 86, 87, 93

roster..... 1, 2, 8, 11

S

surgery.....iii, 1, 2, 6, 7, 11, 14, 15, 31,
59, 63, 81, 83, 91

U

unit.iii, ix, 1, 2, 3, 6, 7, 8, 11, 13, 16,
18, 19, 27, 58, 63, 77

W

ward..... 6, 7, 70