Chapter 5

Design of the TMSFIT

5.1 Introduction

Previous chapter included the hypothesis, inputs, outputs, users, features and the technologies of the proposed timetabling management system. This chapter contains details of design (or analysis and design) of the TMSFIT. The top-level design of the proposed system also included. This chapter will describe the Analysis and designing part of the system such as research planning and system designing.

5.2 Research Planning

Following major tasks conducted through the analysis and design phases of the research.

- Project Planning was done using an online project-designing tool.
- System Requirement Specification (SRS) completed.
- System Design with necessary design diagrams could introduce.

5.2.1 Planning the Research Project

To planning and scheduling the research project, rough Gantt chart was drawn as the following table 5-1.

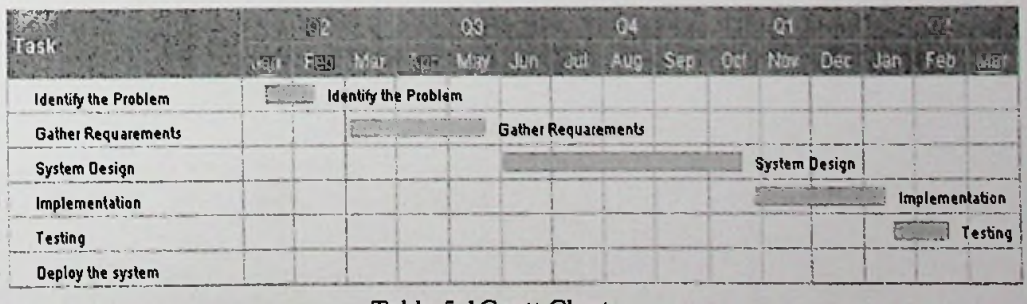

Table 5-1Gantt Chart

5.2.2 System Development Methodology for TMSFIT

Waterfall model was used as the system development methodology of this system. Because, it is having precise requirements and well understood milestones.

5.2.3 Selection of Software Process Mode for the Proposed TMSFIT

Detail requirement analysis was conducted at each different user category getting help of admin of the current timetable management system.

After the system study, the Software Requirement Specification (SRS) for the proposed system was prepared.

5.3 Analysis of the Existing Timetabling System

In order to identify the requirements of the existing system a detailed study at each user category was conducted. User interviews, observation and study of relevant documents were the techniques used to gather requirements of the current system.

5.4 Top level Design Diagram

The system's top level diagram presented as below Figure 5-1

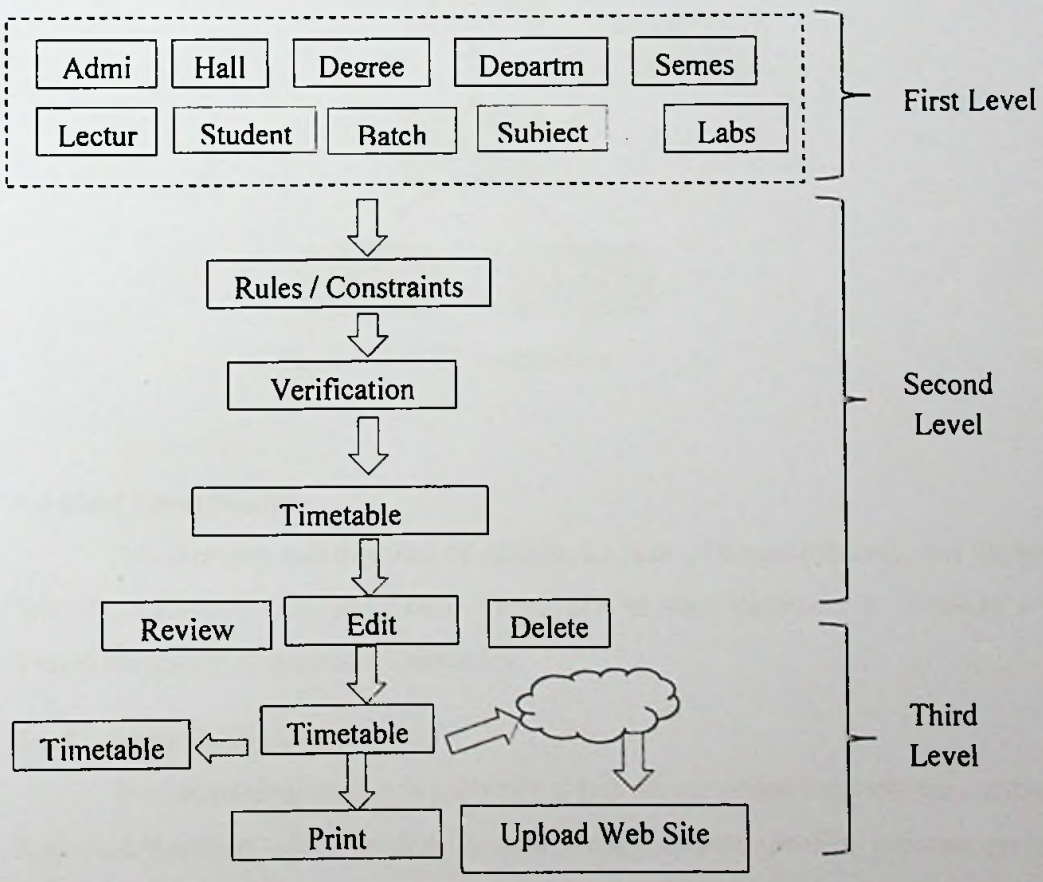

Figure 5-2 Top Level Design of the Proposed System

5.5 MVC Architecture

Yii framework employs the model-view-controller (MVC) design pattern and it is broadly accepted in Web programming [24]. Below Figure 5-2 will illustrate the MVC architecture and it shows how user request is handled by the Yii application as a standard workflow.

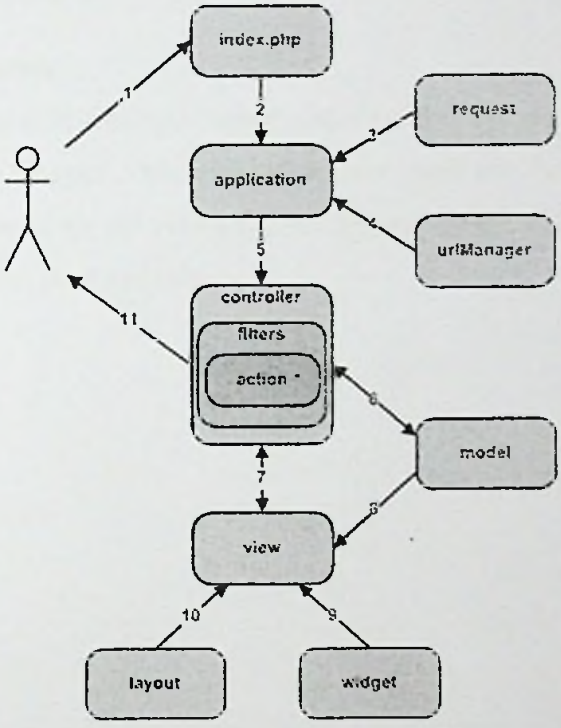

Figure 5- 2 MVC Architecture

5.6 First Level Module

This contains sub modules of Admin, Lecturer, Degree (course), and Student, Subject, Resources and the batch. Those are in ttms database. It interacts with Timetabling Engine generates timetables.

5.6.1 Timetabling Engine

The timetabling engine is primarily a web server which connects the database. It should maintain admin profile, student profile, lecturer profile, process queries; prepare outputs in various formats and so on. This engine also responsible for accuracy and up-to-date information in the database. It is basically designed for maintain the system integrity, security and the privacy.

5.6.2 Database Design of Timetable Management System

The Timetable Management System Database abbreviated as ttms. It stores data of students, lecturers, users, degree programs, subjects, timetables and some more. Student data, resources data, lecturer data, batch data, subject data and timetable data can retrieve from the database. Admin has the authority of modifying and deleting data. Details were taken from the faculty of Information Technology at University of Moratuwa.

5.6.2.1 ER Diagram

An Entity Relationship Diagram can show the database design of the TMSFIT. Each box of the diagram represents a table and ovals are the attributes of that table. Underlined attribute are the primary keys of those tables. Following Figure5-3 is the ER diagram of this database ttms.

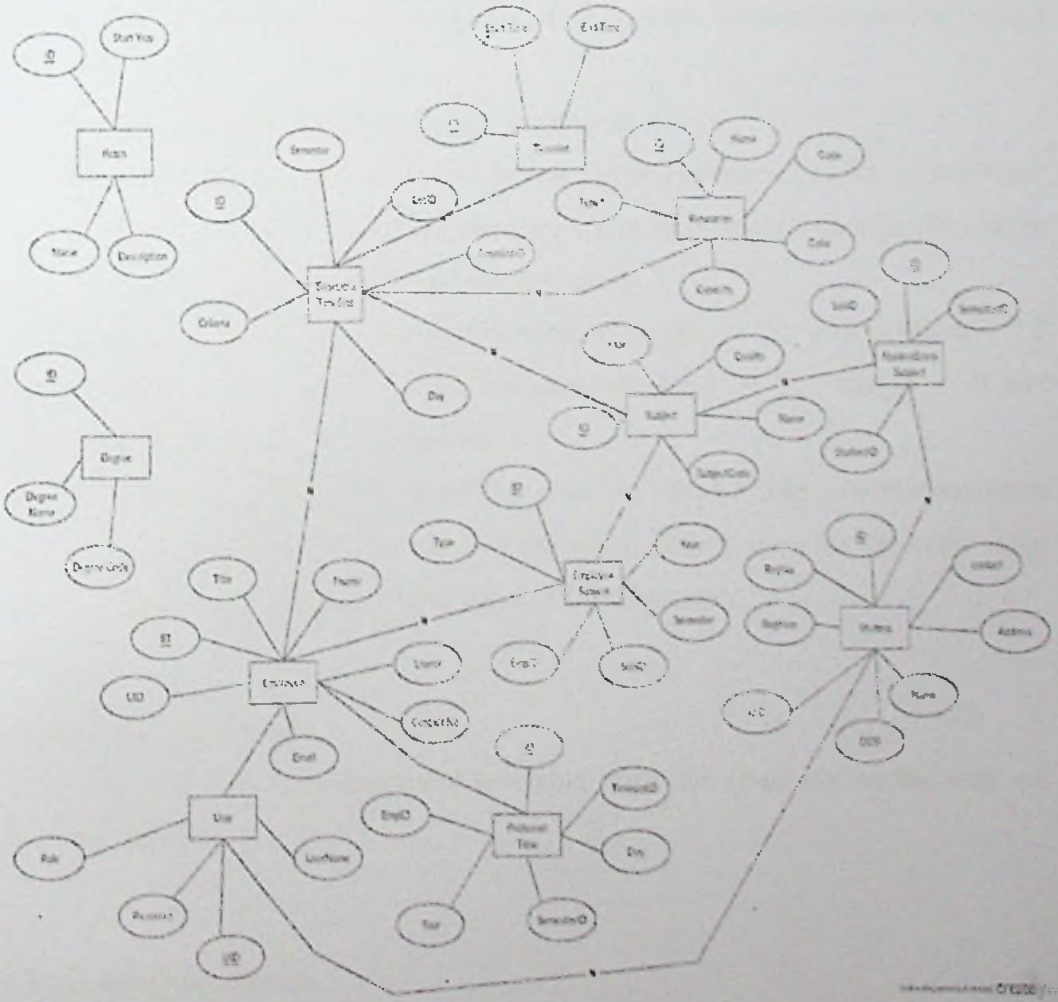

Figure 5-3 ER Diagram

5.7 Second Level Module

This includes the logic of the timetable, constraints or rules, verification, timetable generation, view, delete and edit. Algorithms usually kept in this level. These sub modules act as methods and product of this is the generated timetable.

5.7.1 How the Genetic Algorithm perform

Following is a brief description of algorithms performance.

- AlgorithmController is reside in the controllers module
- actionLoad() ---> basic action method calling from url
- SubjectClass reside in the model class
- loadAllClassByYearSem() --->Load all the subject registered by student for given year and semester
- Algorithm
- getInstance($subjectClass) --->Create Algorithm instance by passing loaded subjectclass
- We are using 100 chromosomes to generate this.
- Inside this method, creates prototype of chromosomes by calling initObject ($numberOfCrossoverPoints,$mutationSize,$crossoverProbability,$mutation Probability,$fitness,$subjectClass) method.
- We need to pass numberOfCrossoverPoints = 2, mutationSize = 2 ,crossoverProbability = 80, mutationProbability = 3 , fitness = 0 and subjectClasses for that method.
- After calling start () in Algorithm class. In here for each 100 chromosomes initialized new Chromosome by assigning subject class for time slot array. Then calculate fitness value for each Chromosome.

5.8 Third Level Module

This level includes generated timetable, view the timetable on the web and print the timetable.

5.9 Modeling the System

Modeling a system is the process of abstracting and organizing significant features of how the system would look like. Modeling is the designing of the software

applications before coding. Unified Modeling Language (UML) tools are used in modeling this system. UML offers different diagrams to model a system.

Diagrams of this research are listed below:

- Use case diagram
- Class diagram
- Object diagram
- Sequence diagram

In this project, the Use case diagram, Class diagram, Sequence diagram were used for system modeling.

5.9.1 Use Case Diagram

Use case diagrams describe what a system does from the standpoint of an external observer. They are used to show the interactions between users of the system and the system. A use case represents the several users called actors and the different ways in which they interact with the system. The main Use Case Diagram is given as below Figure 5- 4

Actors

- User (Admin/ Lecturer/ Student)

Use Cases

- Login
- Create/Modify and Delete Timetable
- Add Lecturer/Course or degree/batch/subject/resource and Student
- View Timetable
- Change Password
- Add Lecture Resources (Hall / Lab)
- Modify and Delete Lecturer/Course/Student/Resources (Hall/Lab)

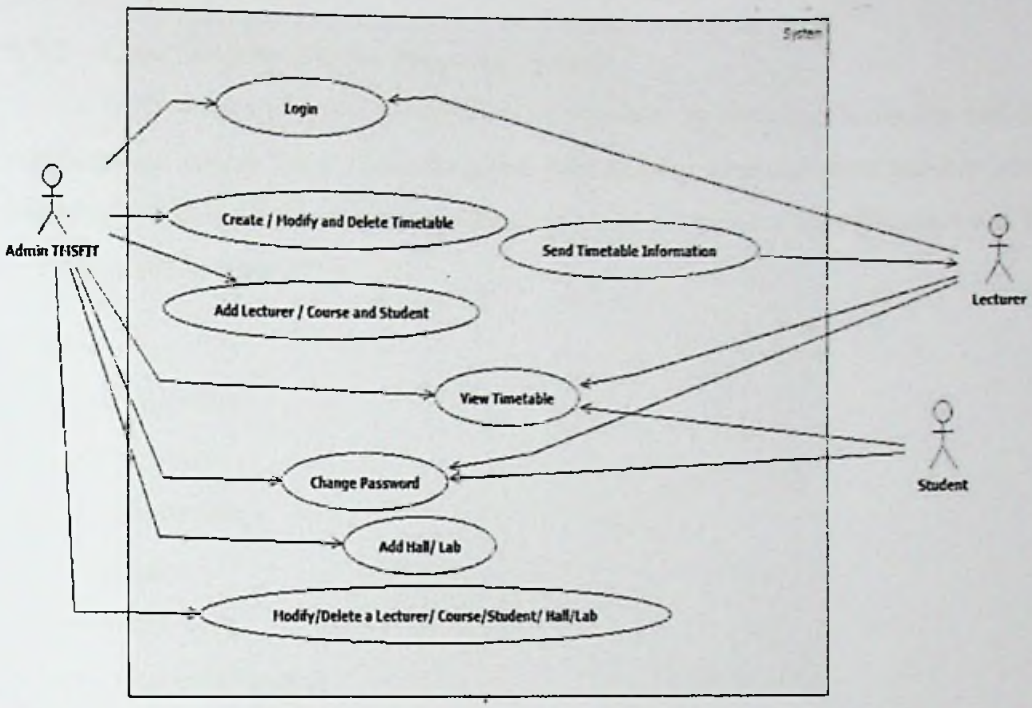

Figure 5-4 Main Use Case Diagram

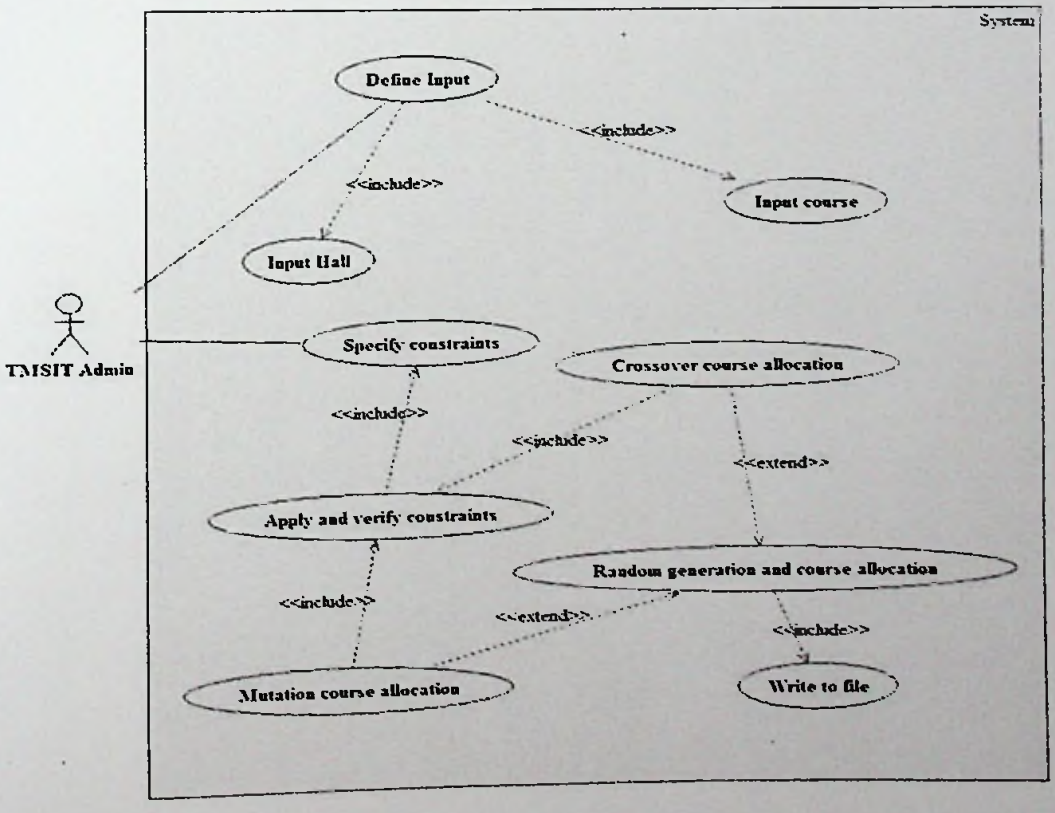

Figure 5-5 Use Case Diagram to Show the Interaction between The System and admin

5.9.2 Class Diagram for the Proposed System

A class diagram gives an overview of a system by showing its classes and the relationships among them. Class diagrams only display what interacts but not what happens during the interaction hence they are static diagrams. Class Diagram can be present as below Figure 5-6

Classes

- IT Faculty
- Lecturers
- Department
- Rooms
- Halls, Labs
- Degrees (Courses)
- Undergraduates
- Postgraduates

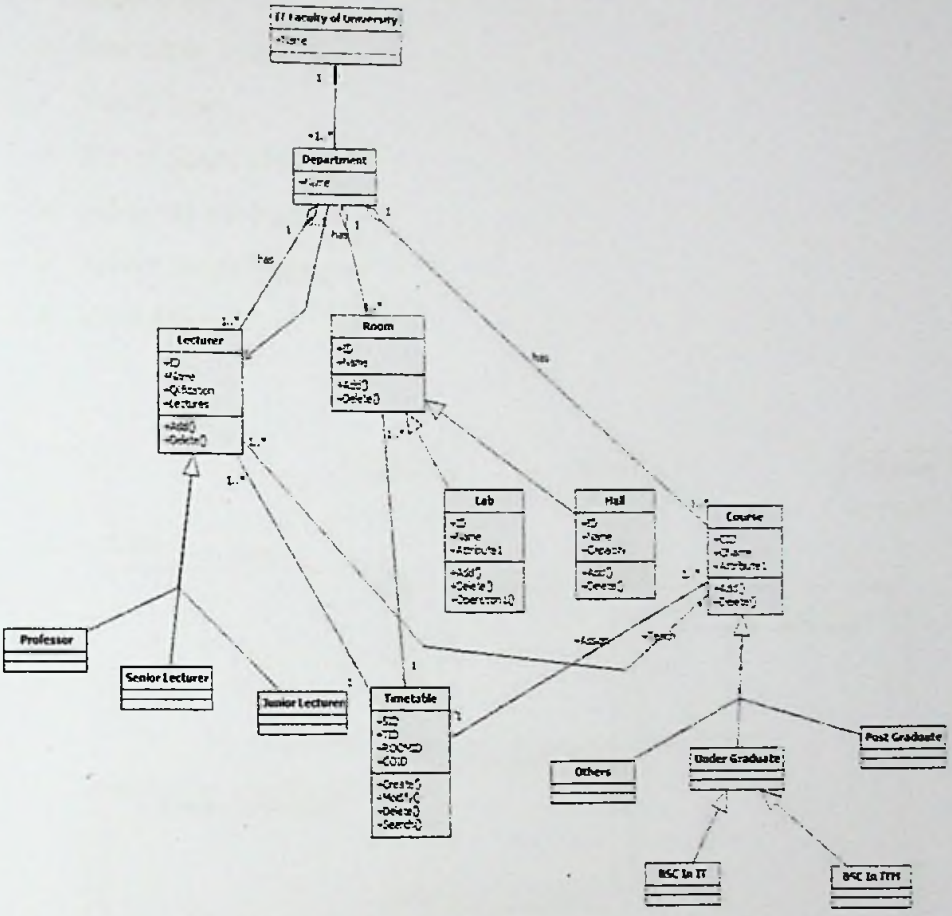

Figure 5- 6 Class Diagram for Overall View of the System

5.9.3 Sequence Diagram

A sequence graphically depicts how objects interact with each other via messages in the execution of a use case or operation. They illustrate how messages are sent and received between objects and the sequence of message transfer. It also details in Figure 5-7 how operations are carried out according to the time of operation.

Classes

- Admin
- Browser
- Web Server
- Database

Messages

- Login

- Valid User
- Successful Validation
- Verify User
- Server Sends Message
- Admin Home Page
- Server Sends Message
- Error Message

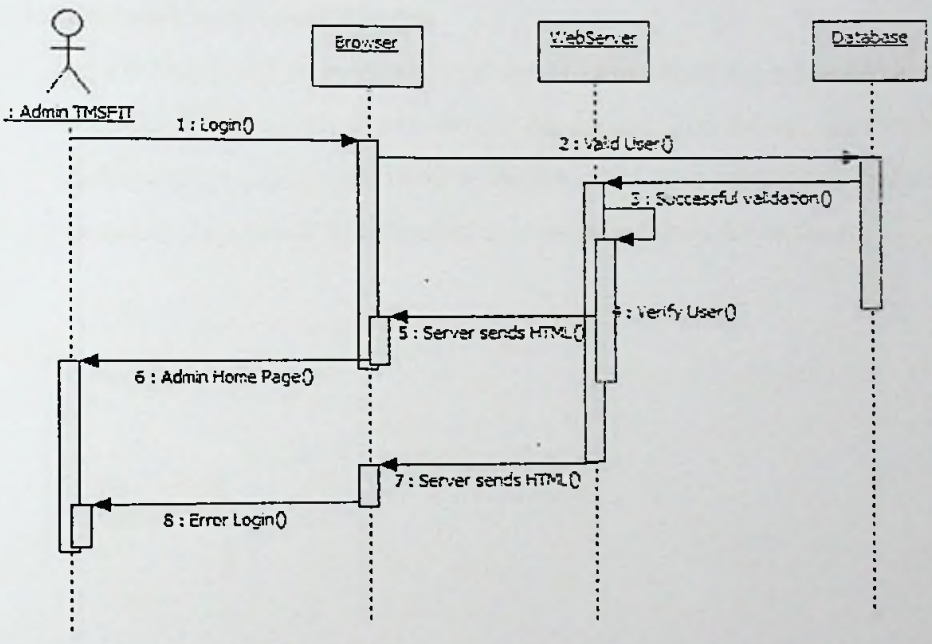

Figure 5-7 Sequence Diagram show how the different objects interact

5.10 User Interfaces Design

This is a description of logical characteristics of each interface between the software product and the users interact with the system. User interfaces are consist of any GUI standards, sample screen images, standard buttons (add, delete) and so on. See Appendix A

However, UI design must be considered whenever users interact with controls or displays. Throughout the project when designing the HCI, I have followed several UI designing techniques.

Basically, targeted users knowledge type is inexperience, so that I have to consider the following techniques when designing the UI.

- Increased Drop Down usage
- UI based field validation
- Highlighting the focused field
- User friendly Error Messages

5.10.1 Increased Drop Down Usage:-

By increasing the drop down field usage rather than typing a value to the system, user can simply select an item from a pre-loaded drop down. This technique is very useful when working with users to decrease the input errors and increase the user performance. Figure 5-8 will illustrate that increased drop down usage.

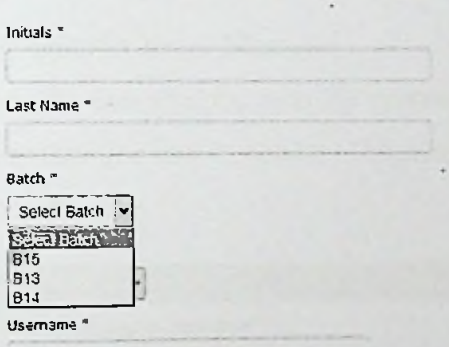

Figure 5- 8 Drop down Usage

5.10.2 UI based Field Validation:-

User is allowed only to enter character value to character type fields, which means rather than validating the user input by the system, numeric fields validations has been implemented at the user interface level.

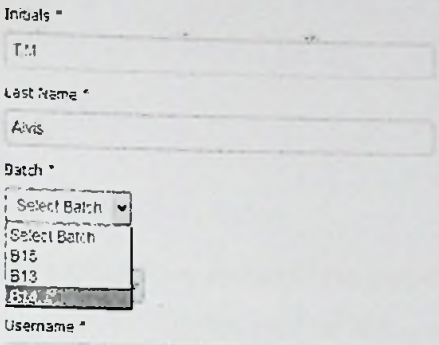

Figure 9 UI based field validation

5.10.3 Highlighting the Focused Field:-

When the user working with the system as an extra facility, currently prompted (focused) field would be colored by the system. Because of this technique, user can easily identify the current working field.

5.11 User-friendly Error Messages

When the user working with the system, some validation rules with the error messages navigate the user to correct operation. Below Figure 5-10 will show a user friendly error message.

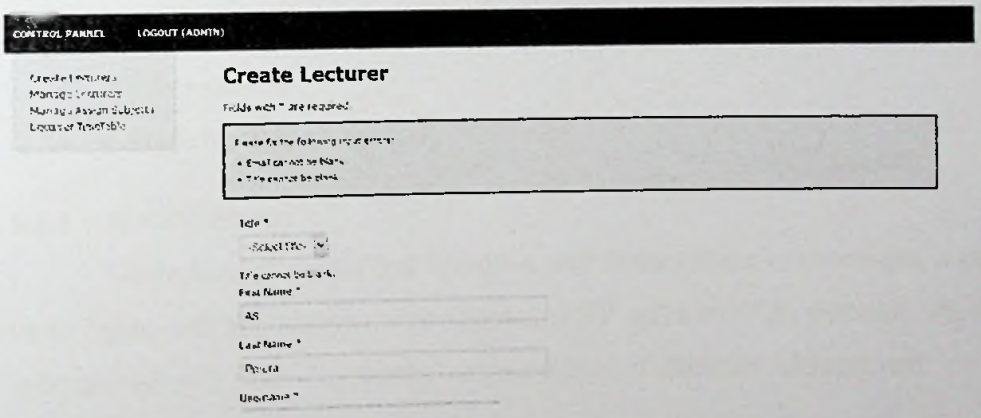

Figure 5- 10 User-friendly Error Messages

5.12 Summery

This chapter briefly described the analysis and design methodologies of the system. It described what each module does. Further, it has given design diagrams examples like Use case diagrams, Class diagrams, Sequence diagram. Next chapter will discuss the how solution is devised as implementation.