LECTURE HALL SCHEDULING AND TIMETABLE MANAGEMENT SYSTEM

KMPBN PERERA
2019

LECTURE HALL SCHEDULING AND TIMETABLE MANAGEMENT SYSTEM

A dissertation submitted for the Degree of Master of Information Technology

K.M.P.B.N. Perera
University of Colombo School of Computing
2019

ABSTRACT

University of Vocational Technology is an Institute of Higher Education & Research that awards academic degrees in various academic disciplines. So, the internal structure of the university should well establish to produce effective & efficient persons to society.

The university is consisting of full time and part-time batches and there are five semesters, thirteen-degree courses and having more than thousands of students. That count is smallest when compared with the other state universities. But that rate is a considerable amount to the emerging university.

Presently, when allocating a lecture hall, lecturers cannot see the current allocations and lecture schedules may crash with other lectures. Because of the above problems and increasing the student and staff population of the university, the lecture room scheduling, timetabling and distributing lecture notes are problematic things to the university.

Therefore, to solve this complicated situation, I propose to develop an automated system. It is consisting of student enrolment, lecture hall scheduling, and virtual learning environment. By using this proposed system lecturers can reserve the lecture hall by their preference and students may get an email notification with the venue and the time of the lecture each day.

Lecturers can log in to the system and view the dashboard calendar and check whether the existing bookings. Then they can reserve the lecture hall or laboratory according to their preference. Also, lecturers can change the schedules and generate email notification for students and other stakeholders.

This system was used MVC architecture for developments and it makes the development context simpler and strong, the system has been built on top of the Model View Controller framework. Using MVC architecture, the system can be structured to support the reuse of program code, which is an important feature of Object orient development. Furthermore, I used PHP, HTML5, CSS and JavaScript and JQuery as development technologies.

Finally, the proposed system will be combined Lecture hall scheduling, Timetabling, Student Feedback, Virtual Learning Environment and provide a productive system for the university.

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: B. N. Perera

Registration Number: 2016/MIT/056

Index Number:16550566

_____

Signature: Date:

This is to certify that this thesis is based on the work of Mrs. B. N. Perera under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr M. G. N. A. S. Fernando

_____

Signature: Date:

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regard to my guide Dr M. G. N. A. S. Fernando for his invaluable guidance, monitoring and constant encouragement throughout this project. His wisdom, knowledge, and commitment to the highest standards inspired and motivated me.

Secondly, I would like to express my gratitude to all the lecturers who directed me to develop the system and gave the theoretical knowledge. I also take this opportunity to express a deep sense of gratitude to my client the University of Vocational Technology for the guidance, assistance given during the all-time period starting from the requirement gathering process. Also, another word of thanks goes to the Head of the Department in the Department of Information Technology at the University.

Furthermore, I wish to mention my husband Rajitha Wedage and without whom, this effort would have been worth nothing. You and my babies' support and constant patience have taught me so much about sacrifice, discipline, and compromise. They give me consistent love during a difficult time.

Last, but not least, I express my heartfelt gratitude to my parents, sisters and my office colleagues and friend who were with me during the project period by providing advice, guidance and valuable support to make this effort successful.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

UoVT – University of Vocational Technology

CSS - Cascading Style Sheet

DBMS - Database Management Systems

ER - Entity Relationship

GUI - Graphical User Interface

HTML - Hypertext Mark-up Language

MVC - Model-View-Controller

OO - Object Orient

OOA - Object Oriented Analysis

OOD - Object Oriented Design

PDF - Portable Document Format

RAM - Random Access Memory

SQL - Structured Query Language

UML - Unified Modeling Language

SMS – Short Message Service

TV – Television

HOD – Head of the Department

XAMPP - Cross Platform, Apache, MariaDB, PHP, and Perl

CHAPTER 01

1 Introduction

1.1 Problem Domain

University of Vocational Technology's (UoVT) lecture hall scheduling & timetabling were done manually by an associate officer. This manual process is time consuming and tiresome as well as arising many kinds of human errors. In the current manual approach, each lecturer should provide their timetable to the associate officer and that officer should check the availability and allocate the halls and display them on notice board weekly or sometimes two or three times per week. When allocating halls, they were not considered about the facilities and the number of students in the class. Because of that many problems were arising.

Key issues that identified:

1. Lecture hall is not adequate for all students in the class
2. Required teaching facilities were not in lecture hall (Multimedia projector, speakers, TV)
3. In the same time slot, two or more lecturers were waiting for the same lecture hall.
4. Until the last moment, the students were not knowing about the venue of the lecture.
5. Difficulties about distributing lecture notes among students and collecting assignments

Because of the above reasons, the requirement of the system is highly needed.

1.2 Motivation for the project

As a continues learner I have been interested to develop systems and make day to day activities easy. Because of that to overcome the above problems supposed to develop a student registration system to start the project.

After the student registration, students should know their lecturers and the timetables and according to the timetable, they should know the venues of the lectures. All the time students were not knowing the venues of the lectures until the last moment because of the changes. Always crashing the lecture schedules and students are coming to the lecture halls as latecomers. Sometimes lectures also seeking the lecture hall by going each doorstep and spending their time in vain. Therefore, to solve this problem I have been encouraged to develop a lecture hall scheduling system.

Another huge issue is there was no virtual environment handling by the university to distributing the lecture notes. Sometimes lectures forwarding the lecture notes by email and sometimes distributing by hand. Because of that, some students are getting the lecture notes, and some are not getting the lecture notes. So, to solve this problem introduce to develop a virtual learning environment to the university.

Further, more I selected the iterative enhancement model to develop the system gradually.

1.3 Aims and Objectives of the proposed system

The aim of the proposed system is a lecture hall scheduling, timetabling and virtual learning environment to the University. And, I should identify the number of students who wish to participate in the lecture. So, I am attaching a student registration system to this project. Also, to make the advancement of this project I propose to add a content uploading and downloading module.

The proposed system is fully automated with the features of text notifications and email notifications. This will help lecturers', office staff and students to overcome the manual lecture halls allocation difficulties.

Further, the proposed system is web-based and will provide a better user experience by reducing time-consuming and the human errors that happened. Lecturers can reserve lecture halls with required facilities. Finally, the lecture hall scheduling details were sent to the relevant lecture, student representative and office assistant who were on duty on that day via email and text. If there any remarks lecture can add comments by accessing the system.

The main objectives of this system are listed below:

1. Integrate with a student registration module to identify the number of registered students in each degree course.
   The importance of identifying the number of students is, the lectures can get the maximum number of students that wish to participate in the lectures

2. Lecturers should have permission to reserve lecture halls and change allocations with their preference.
   If the lecturer changes the existing allocation, the student representative will receive an e-mail notification immediately.

3. Student representatives of each degree course and relevant lecture should have the facility to get email notifications.

4. Registered students can upload their assignments and download lecture notes via the content management module.

1.4 Scope of the project

This system is developed for the academic staff of the university to reduce the time consuming regarding the lecture hall scheduling, timetabling and distributing lecture notes and collecting assignments.

Modules that consisting with the system:

1.4.1 User login:

Initially, only the administrator can access the system and admin should grant privileges for other users, (Head of the departments (HOD), lectures). There is a user registration form for new users.

1.4.2 Dashboard:

- Student Registration
- Timetable Allocation
- Lecture hall scheduling
- Download the timetable
- Download the lecture hall scheduling table
- View the facilities of lecture halls
- Edit lecture halls & timetable
- Add courses
- Add lecture halls
- Content management
- Change reservation

1.4.3 Functional Requirements:

The users can view the dashboard and the following available features.

- Lecturer should be able to see personalized timetable
- The lecturer should be able to request a timetable modification for specific period.
- Download the timetable –the relevant lecturer can download the timetable.
- Download the latest schedules of the lectures
- Ability to change the lecture schedules
- Ability to search location, with different criteria. (regarding to the building, degree, semester)
- Ability to add courses.
- Enable and disable personalized notifications.
- Upload lecture notes and assignments.
- Download lecture notes and assignments.

1.4.4 Non-Functional Requirements:

- Responsiveness and user friendliness
- Speediness of the system
- Less weight
- Reliability and correctness

1.4.5 Technologies to be used:

- HTML5/CSS
- PHP
- MySql
- Wordpress
- Javascript
- Jquery

1.4.6 End users:

- Head of the Department
- Lecturers
- The students
- Associate Officers

CHAPTER 02

2 Background

2.1 Introduction

This chapter is mainly considering the similar systems which are currently available on the internet. There are some online systems which used to handle the lecture hall scheduling and timetabling in universities. This manual time-tabling system used in the institution is time-consuming and energy-sapping which leads to loss of information, general apathy, clashes and wastage of resources. The solution to these problems is a lecture hall scheduling with a web-based timetabling system. Computerization makes it easy for a university to do its lecture hall scheduling and timetabling process effectively. Currently available system is difficult to customize according to the fluctuations of the allocation process. That having an implementation system is user-friendly and less technical skills need to operate the system. Ability to search location, staff based on different criteria.

2.2 Analysis

Basic requirements of the proposed system are listed below:

1. Integrate with a student registration module to identify the number of registered students in each degree course.
2. Lecturers should have permission to reserve lecture halls and change schedules with their preference.
3. Student representatives of each degree course and relevant lecture should have the facility to get email notifications.
4. Registered students can upload their assignments and download lecture notes via content management module.

After identifying the above matters the below-detailed description was mentioned.

1. There were Lecturers, students and administrative staff members interacting with the system.
2. Student can register for only one course and each course is having a maximum of 50 students
3. 13-degree courses offering from the university [1]
4. There are two faculties in the university and having professors, doctoral, senior lecturers, lecturers and demonstrators.
5. Lecturers are doing lectures for one or more-degree courses
6. There are three building and consisting of classrooms and laboratories
7. Laboratories were divided for five types called ICT labs, Mechanical labs, Construction Technology and Resource Management labs, Language labs food labs
8. Lecturer can reserve a lecture hall by using the timetable and regarding with the facilities they wanted to do the lecture
9. After reserving the lecture hall student representative will get an email notification.
10. If there any changes happen before the reserved date again student representative will get an e-mail notification.
11. Each lecturer will be uploading the assignment and lecture notes to the virtual environment
12. Students will be using the virtual environment to upload their assignment answers and download the lecture notes

2.3 Review of present well-known online booking systems

There are many lecture hall scheduling and timetabling systems available in all over the world. These systems are used to improve the efficiency of the scheduling process. following are the similar systems are:

- **Online booking system in university of Peradeniya [2]**

The Engineering Faculty of the University of Peradeniya is having an online lecture hall scheduling system. It is very useful for lecturers and students to find the lecture rooms quickly and plan the upcoming lecture schedules in the right manner. Using this system lecturers can choose the suitable lecture hall by referring to the lecture hall facilities shown in the system.

Some features of the system:

1. Reserve the lecture rooms by checking its facilities like

Eg:

The seminar rooms are allocated according to the following priority order

a. International meetings, conferences organized by the Faculty
b. Domestic meetings, conference organized by the Faculty
c. Internal meetings, seminars organized by the Faculty
d. Regular lectures of the Faculty
e. Meetings, lectures organized by the outsiders of the Faculty

2. Anyone can register and view the reservations and external users can make bookings with charges.

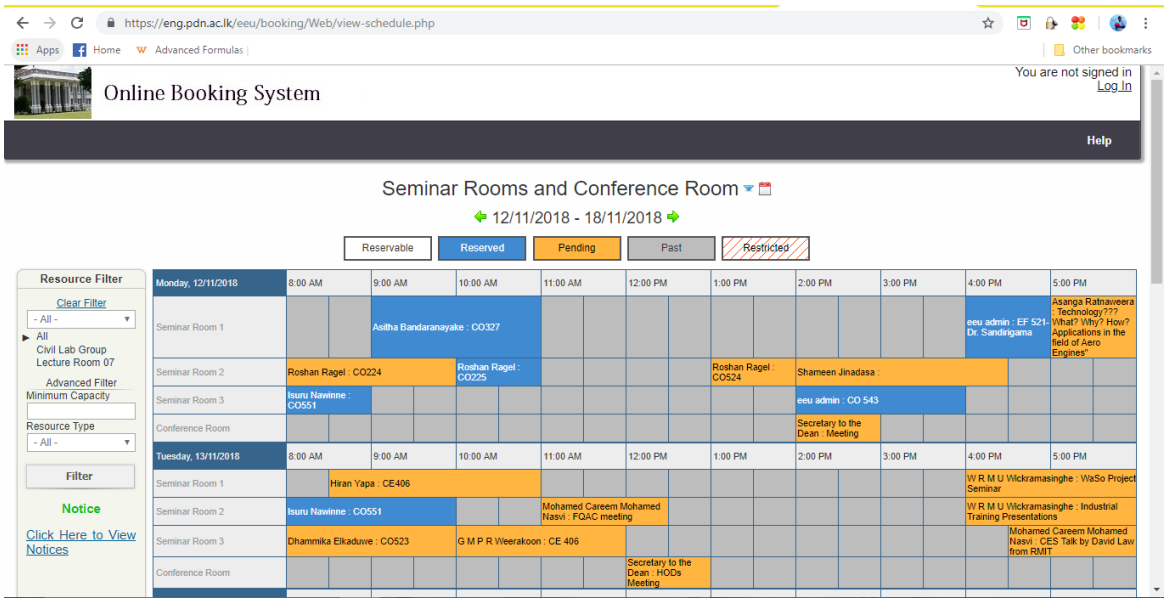


Figure 2.1 Room allocation in university of Peradeniya

- **University of Otago, New Zealand Lecture room booking and timetabling [3]**

In University of Otago is having a web-based lecture room booking and timetable planning system. Timetables of lectures, tutorials, laboratories and other classes for most papers are arranged centrally to make the most effective use of the University's buildings and facilities.

Some of the features of the University of Otago boking system is indicated below:

1. **View the university timetable**
   The University Timetable may be viewed online through the Web Timetable Viewer. Users can choose to view the timetable by paper(s) or room(s) within a selected time frame.

2. **Display the timetable planning chart**
   It is easier for students and the lecturers to plan their own work.

3. **Booking rooms**
   Students can make a request to book a room for appropriate group activities where spaces are available.

4. **Timetable information for staff**
   This will provide information about late timetable change request process.

- **<u>Room and lecture hall scheduling in University of Hamburg Germany [4]</u>**

In the university of Hamburg Germany is having a room and lecture hall scheduling process and lectures can check the availability from the individual week tables and make reservations. The process of booking is mentioned below:

1. Check the availability
2. Carry out preliminary planning and make arrangement with lecturer
3. Make booking request
4. Wait for confirmation
5. Check booking entry

The University of Hamburg is a well-known university in the world, and it is spreading its departments in a large geographical area. So, to manage all the working schedules is a must thing to the university. To manage the students and lecturers are more difficult by using a manual timetabling and lecture room booking system. Because of that, the online system is most precious to the university administration.

- **<u>Lecture room booking system in Ruhr-University Bochum Germany [5]</u>**

The lecture hall assignments are recorded via a database program. All faculties and institutes have decentralized access to this database and can reserve their lecture hall reservations, for which they have pre-emption rights, directly via the database. In this system reservation request is send to the specified email address and then only it starts their process.

2.4 Critical Analysis of Similar Systems

In the above Section, 2.2 described the similar systems and some of their features. In this section compare these features with respect to the proposed Web based lecture room allocation and timetabling System. Table 2.2 shows a summary of the critical analysis of similar systems.

Feature	Peradeniya	Otago	Hamburg	Ruhr
Reservation request via e-mail	No	No	No	Yes
Check the availability before booking	Yes	Yes	Yes	Yes
Check the facilities	Yes	Yes	No	No
Change the booking	Yes	Yes	Yes	Yes
SMS & e-mail alert	No	No	No	No
Make request from external users	Yes	No	No	No

Table 2.1 Critical analysis of similar systems

2.5 Summary

Web based Lecture hall scheduling and timetabling system is very much effective for the universities because they are handling large number of students and lecturers regarding with several manners. It automates the whole allocation process. Above section 2.2 mentioned the several systems which have different features. Also, table 2.1 has been shown the critical analysis of the feature in those systems. The proposed system combined all these features together to satisfy user requirements.

CHAPTER 03

3 Methodology

3.1 Introduction

This chapter gives an overview of system analysis, design, and implementation. Furthermore, according to the requirement gathering and analyzing can draw the entity-relationship diagram, unique use case diagram and sequence diagram. Requirements gathering and analysis is a weighty practice for a successful project. The methodology is describing Domain understanding, requirements gathering, organizing the requirements, construct them in a relevant manner, prioritization and justification. Suitable procedures were involved to make this phase effectively Moreover, it describes functional requirements, non-functional requirements and hardware requirements of the system.

3.2 Analysis the current system

To analyze and recognize the main system functionalities carried out the domain analysis. Current lecture hall scheduling and the timetabling system are fully manual and having a lot of troubles. They have not a proper method to allocate the lecture halls and inform about the allocations for lectures and the students.

According to the analysis following system functionalities were identified:

- Create the timetable by the head of the departments.
- Allocate the lecture halls/labs according to the timetable by officer-in-charge.
- Display the timetable with allocated lecture hall on notice board.

Figure 3.1 represent the high -level use case diagram for existing manual system

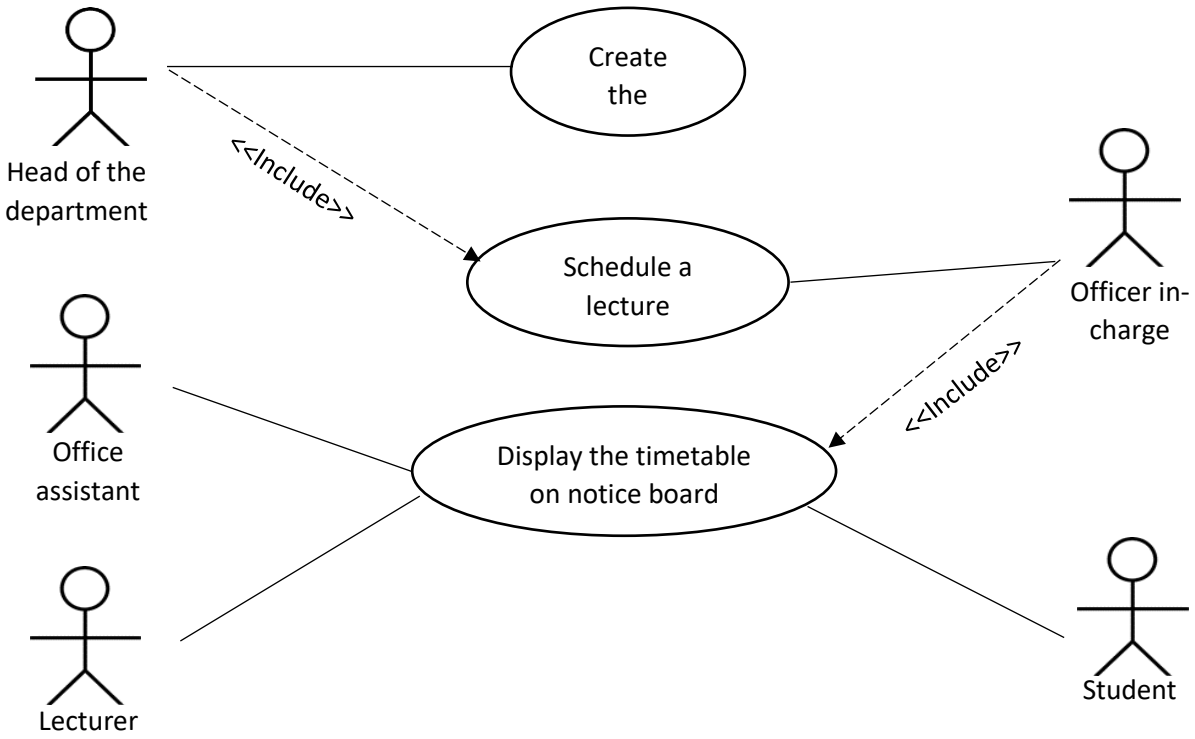


Figure 3.1 high -The high-level use case diagram for existing manual system

3.3 Requirements gathering

Several kinds of procedures were engaged here to collect requirements from different stakeholder viewpoints. The main approaches used for the requirement gathering process were; interview, observation, document review, and scenarios.

Both open-ended and close-ended interviews were conducted for prompt existing also to identify new problems. Relevant documents and current systems were analyzed and studied through a sampling observation methodology. Collected information was represented using scenarios. The use case diagram of each module was designed to get a clear picture of the gathered requirements.

3.4 Problems of existing process and proposed solutions

The requirements gathering phase identify the problems in the current process and following discuss the propose solutions for relevant problems. Table 3.1 displays Problems encountered and solutions.

<table>
<tr><th></th><th>Problem</th><th>Propose Solution</th></tr>
<tr><td>1.</td><td>When lecture hall scheduling, scheduled lecture hall is not enough for all students.</td><td rowspan="2">Before allocating the lecture hall or lab, person can check the facilities of the lecture hall.</td></tr>
<tr><td>2.</td><td>Required facilities were not in lecture hall (Multimedia projector is not in the lecture hall, speakers, TV)</td></tr>
<tr><td>3.</td><td>Another lecturer also waiting to that lecture hall in the same time slot</td><td rowspan="2">In the calendar view displaying the reserved dates for lectures.</td></tr>
<tr><td>4.</td><td>Time clashes between lectures.</td></tr>
<tr><td>5.</td><td>Students were not knowing the lecture hall that they were needing to go until the last moment.</td><td>Generating e-mail notification for students and the lectures</td></tr>
<tr><td>6.</td><td>Difficulties about distributing lecture notes among students and collecting assignment</td><td>- Lecturer can upload lecture notes and assignments
- Students can upload the assignment answers and download the lecture notes</td></tr>
</table>

Table 3.1: Problem encountered and solutions

3.5 Requirements of the proposed system

3.5.1 Functional requirements

A functional requirement specifies “What should the software system do?
This was to express the required behavior of the system to be built or what the system supposed to do. Below listed are the main functional requirements of the proposed system.
Successfully login users can display the dashboard and there are several features available for users.

- Lecturer should be able to see personalized timetable.
- The lecturer should be able to request a timetable modification for a specific period. (one week)
- Download the timetable – relevant lecturer can download the timetable.
- Download the latest allocation of the lecture hall.
- Ability to change an allocation.
- Ability to add courses, faculties, departments, buildings, facilities and students’ feedback
- Enable and disable personalized notifications.

3.5.2 Non-functional requirements

Non-functional requirements are the requirements which are not straightly affected with the exact basic functions provided by the system. But, the weakness of meeting non- functional system requirements may make the whole system impracticable. Non-functional requirements are relevant with quality attributes, quality of service requirements and non-behavioral requirements.

The non-functional requirements of the system are listed below.

- The system would provide a responsive and user-friendly environment including flexible interfaces.
- A person who has average computer skills can work with the system with a short period of training.
- Should keep up security and reliability of the system.
- The system enhancements should be able to be done.
- Speediness of the system.
- Less weight
- Reliability and correctness.

3.6 Proposed architecture for the system

System was used MVC [10] architecture for developments. MVC which represents **M**odel **V**iew **C**ontroller is a software design pattern for developing web applications. It divides given software application into three interconnected portions, which has a full skill to support speedy web application development and dynamic interaction with the database. Figure 3.2 shows the collaboration tracked in the MVC framework.

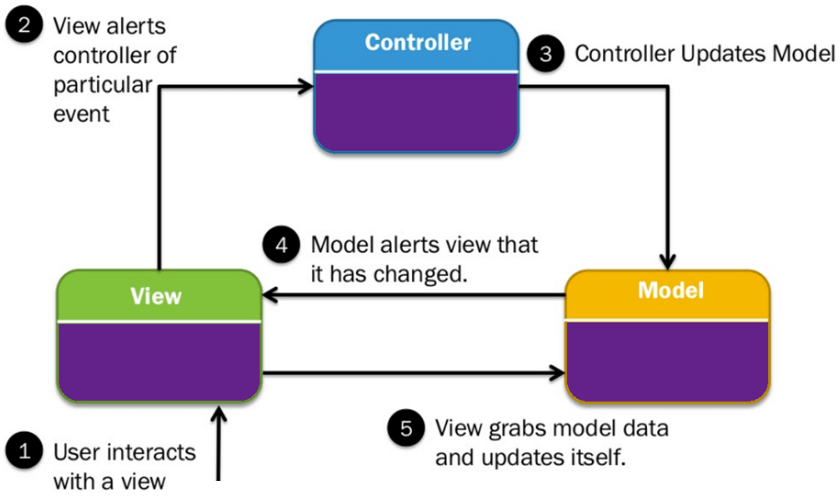


Figure 3.2 Interaction followed in the MVC framework

Controller place at the heart of the architecture and it interconnected with the handler over the web browser. Behind the scene, it associates with the Model and the View mechanisms of the architecture.

The Model encloses the database. Therefore, the Controller can prompt the database according to the handler activities on the browser. It meant that the database is improved by the requirements given by the user. This develops user interactivity and therefore the complete experience.

The Controller shifts the Model according to the query request given by the user, and in return, Model notifies View to update the user interface (UI) on the screen relevant to the database modification. The cycle restart when the user enters another request as query

3.7 Design techniques

Design techniques are the methods used to model the system. There are different design techniques such as Modern Structured design, Rapid application development, and prototyping. Object-oriented design technique was chosen out of them because of code reusable and recycling facility, design benefits and maintainable facilities with Objects and classes.

- Use-Case diagrams – demonstrates what the system wants to do.
- Sequence diagrams – shows how the objects relate actively.
- Entity relationship diagrams

To make the expansion background simpler and robust, the system has been built on top of the Model View Controller framework. Using MVC architecture, the system can be organized to support reuse of program code, which is a significant feature of Object-oriented development

3.8 Proposed System Design

The following diagram in Figure 3.3 represents the high-level Use-Case of the student category of the projected system.

Actor: student

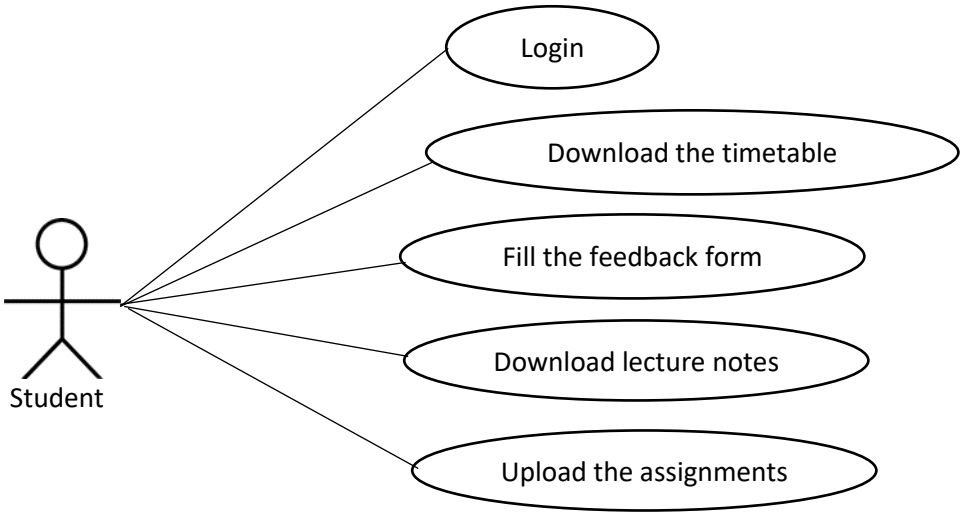


Figure 3.3 Student use case for proposed system

Actor: Lecturer

The following diagram in Figure 3.4 represents the high-level Use-Case of the lecturer category of the projected system.

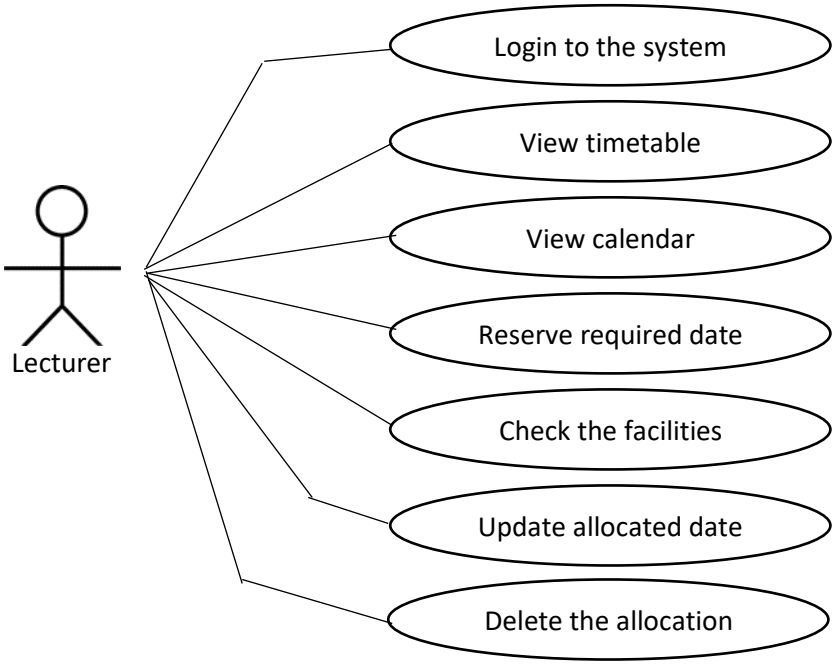


Figure 3.4 Lecturer use case for proposed system

Actor: Administrator

The following diagram in Figure 3.5 represents the high-level Use-Case of the administrator category of the projected system.

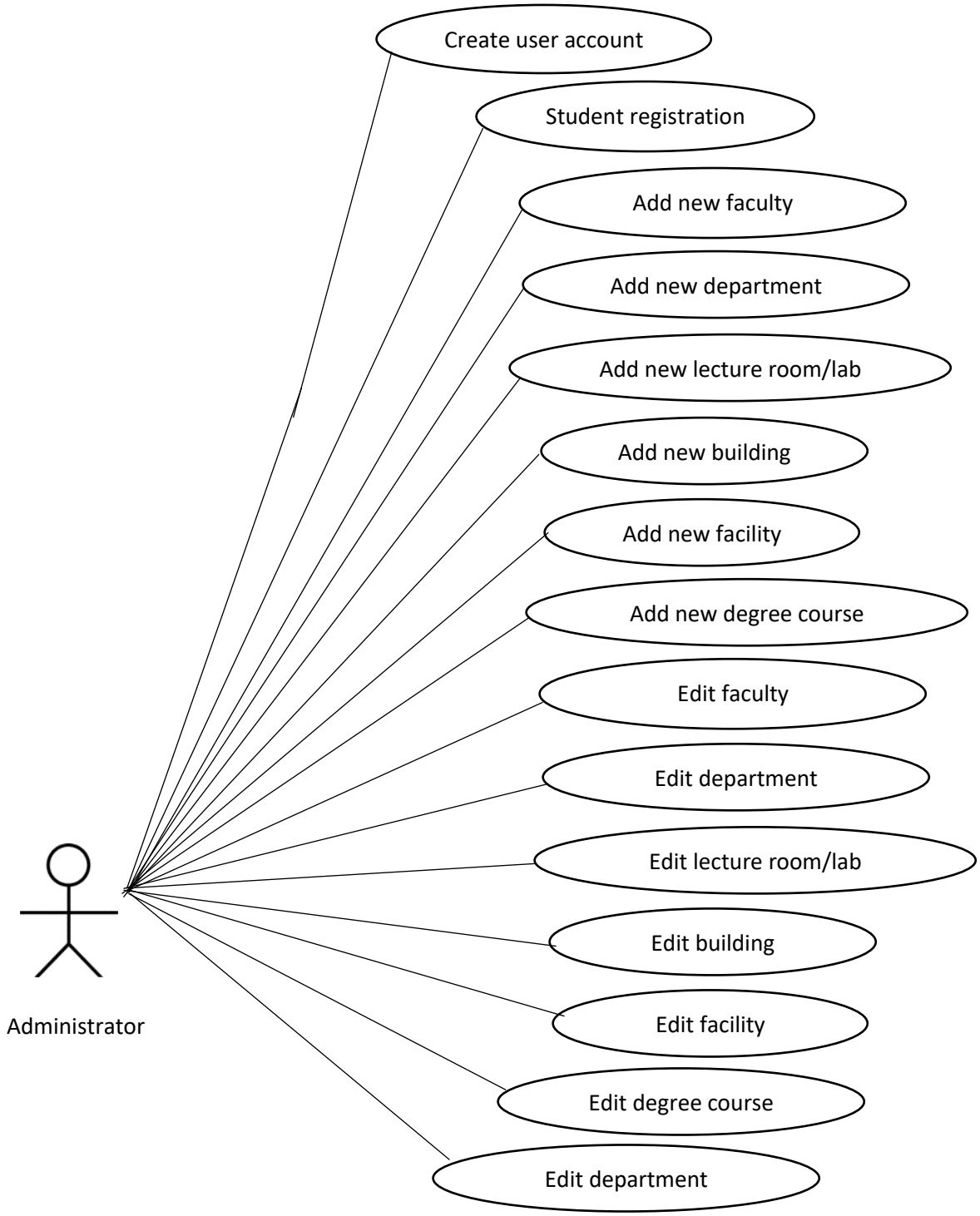


Figure 3.5 Administrator use case for proposed system

The system will promote the users to bring out their responsibilities effectively and efficiently by reducing the manual human effort and by generating reports to take decisions on time. The recommended result has been separated into a few units to make the development and understanding easier.

The modules are as follows:

1. Lecture hall scheduling
2. Timetable generation
3. Student registration
4. Upload lecture notes
5. Download lecture notes
6. Upload assignment answers

- Lecture hall scheduling

User should be able to allocate the lecture rooms/labs by using the dashboard calendar. Also, user can view the whole allocations via the calendar.

- Timetable generation

After allocating the lecture hall/lab automatically generate the timetable view

- Student registration

Student should be able to register to the system

- Upload the lecture notes

Lecturers can login to the system and upload the lecture notes and assignments.

- Download the lecture notes

Students can login to the system and download the lecture notes

- Upload assignment answers

Students can login to the system and upload the assignment answers.

3.9 Database design

Database design is completed over data modeling. Database design is used to stimulate the construction of business items used in the client/server system. The database tables were normalized to 3rd normal form to avoid redundancies. The following Entity Relationship diagram in Figure 3.6 represents the structure of the relational database formation of the system.

User login

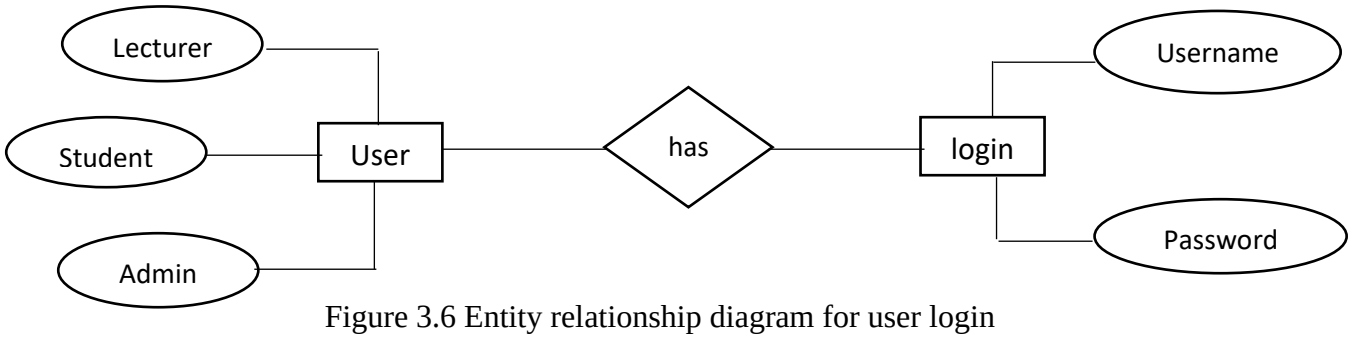


Figure 3.6 Entity relationship diagram for user login

3.9.1 Lecturer

The following Entity Relationship diagram in Figure 3.7 represents the structure of the relational database formation of the lecturer module of the system.

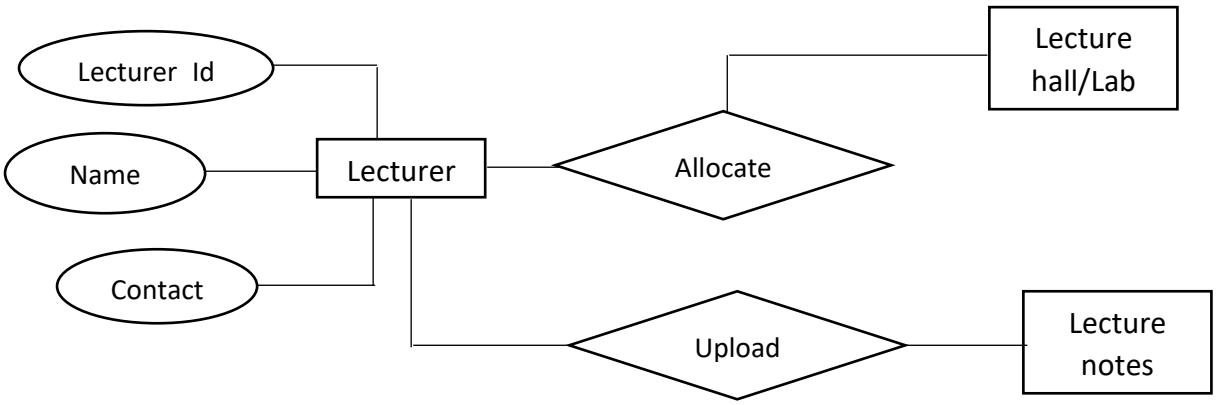


Figure 3.7 Relationships for entity lecturer

3.9.2 Student

The following Entity Relationship diagram in Figure 3.8 represents the structure of the relational database formation of the student module of the system.

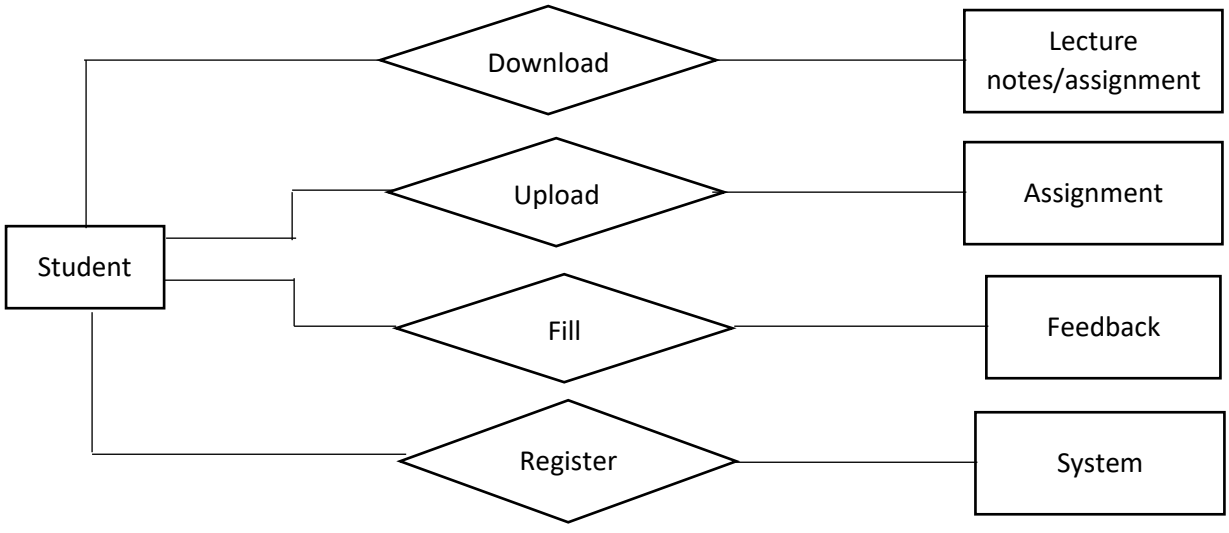


Figure 3.8 Relationships for entity Student

3.9.3 Administrator

The following Entity Relationship diagram in Figure 3.9 represents the structure of the relational database formation of the add module of the administrator user level.

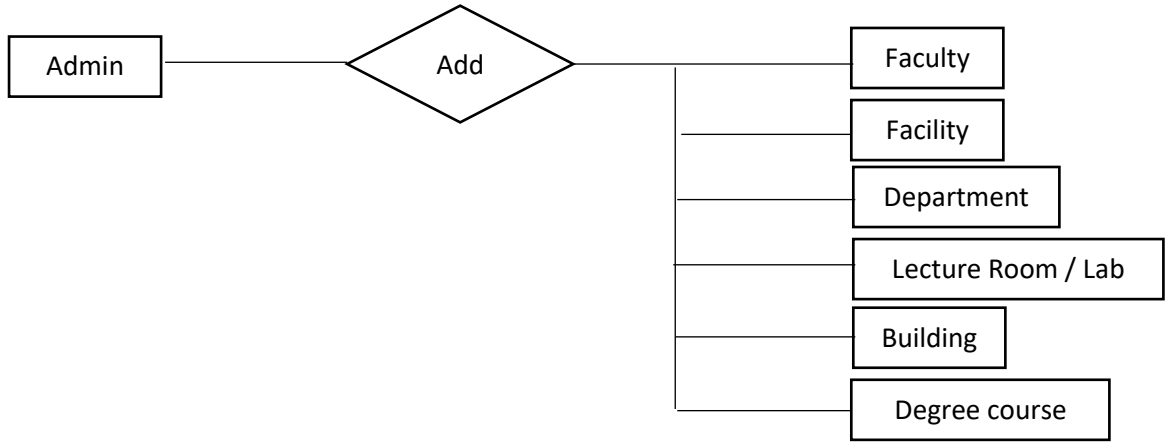


Figure 3.9 Relationships for entity Administrator

The following Entity Relationship diagram in Figure 3.10 represents the structure of the relational database formation of the delete module of the administrator user level.

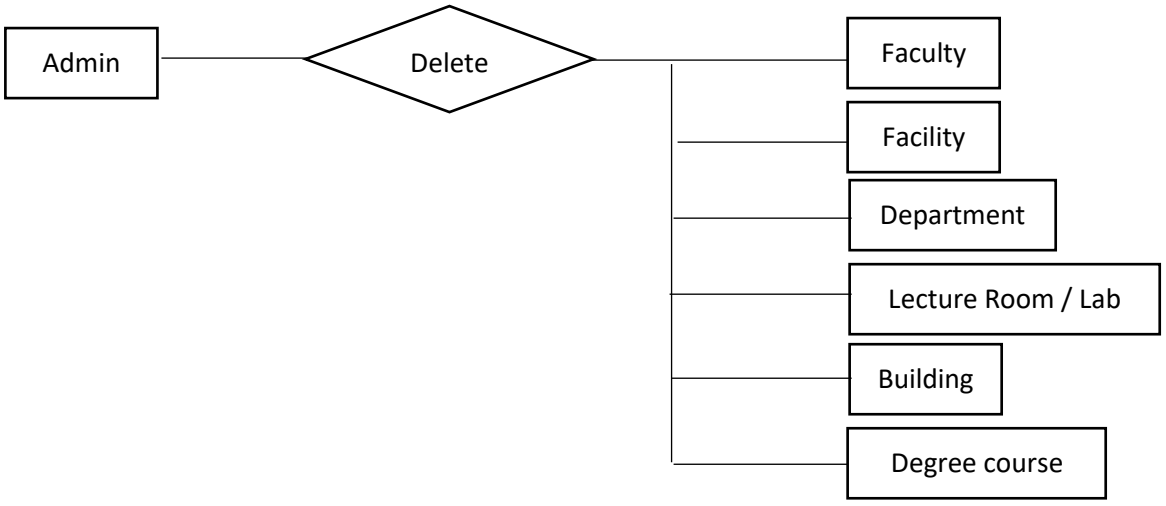


Figure 3.10 Relationship for entity Administrator

The following Entity Relationship diagram in Figure 3.11 represents the structure of the relational database formation of the update module of the administrator user level.

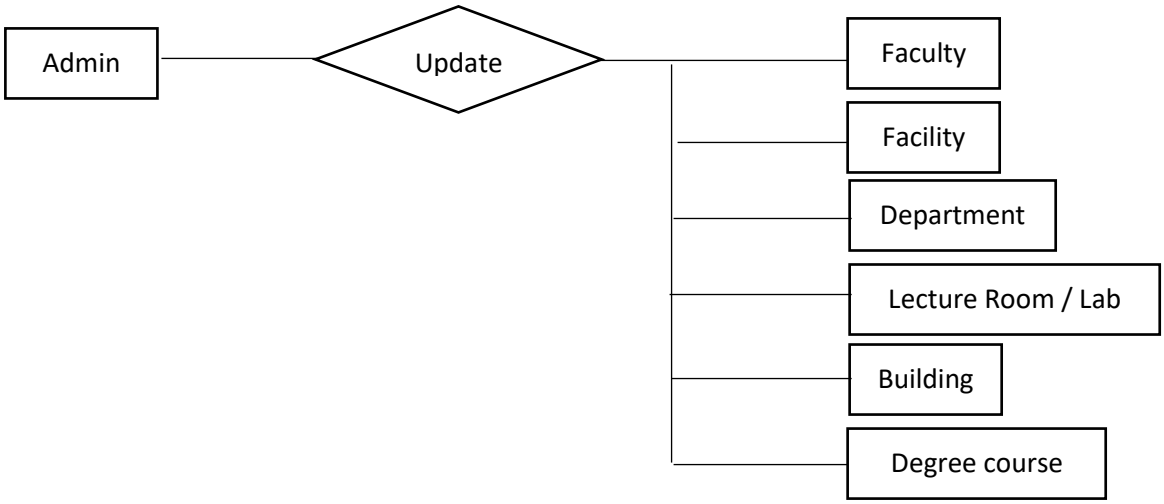


Figure 3.11 Relationship for entity Administrator

The following Entity Relationship diagram in Figure 3.12 represents the lecture hall scheduling and timetabling system

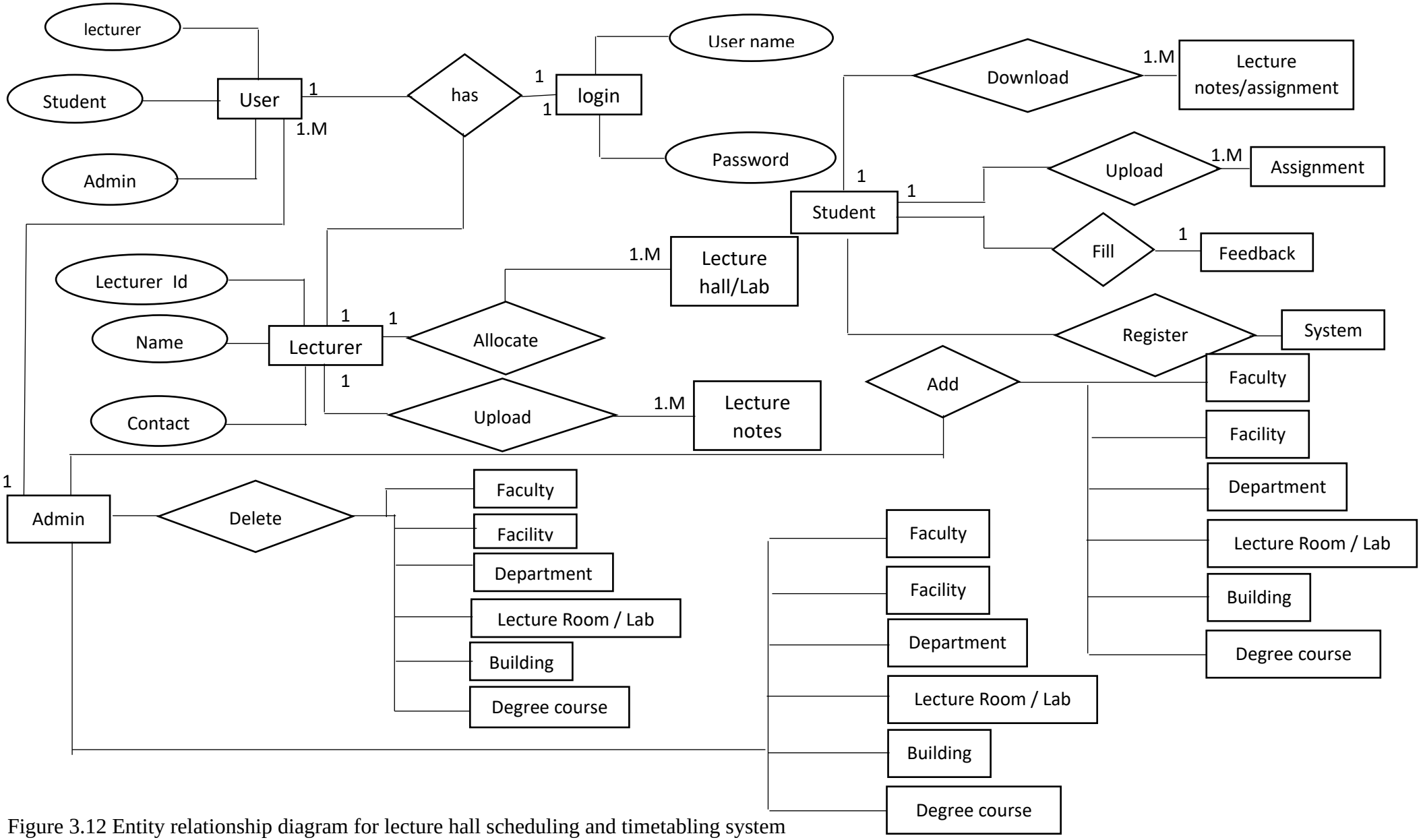


Figure 3.12 Entity relationship diagram for lecture hall scheduling and timetabling system

3.10 Sequence diagram of the system

3.10.1 Sequence diagram for Administrator functionalities

Following figure 3.13 is shows the sequence diagram of the system

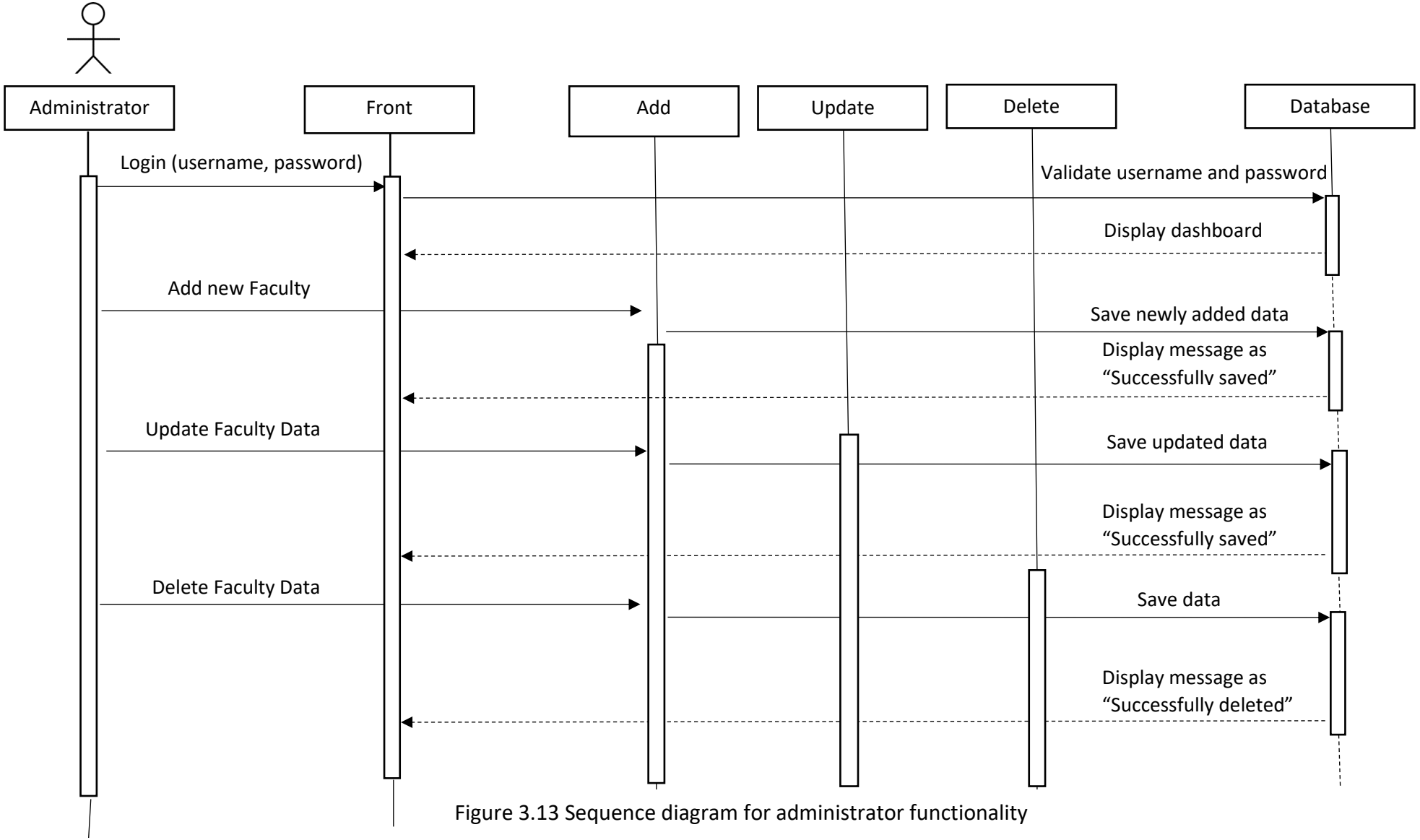


Figure 3.13 Sequence diagram for administrator functionality

3.10.2 Sequence diagram for Lecturer functionalities

Following figure 3.14 is shows the sequence diagram of the lecturers' functionalities for the system.

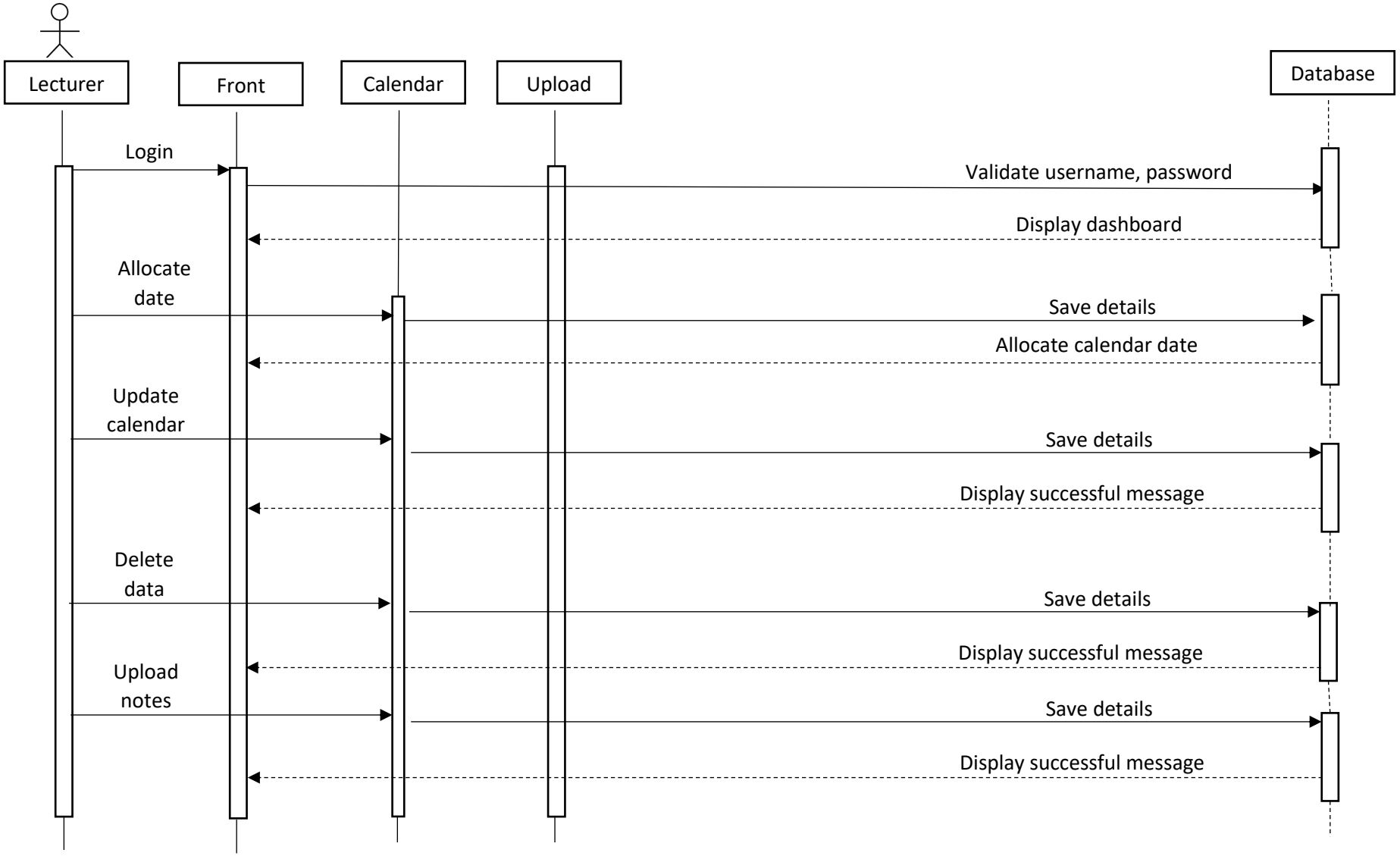


Figure 3.14 Sequence diagram for lecturer functionality

3.11 User interface design

The user interface is the primary and first interaction with the user and system. So, it should interact with the user. User Interface (UI) Design attentions on what users imagining through system and confirming that the interface has essential facilities that are relaxed to understand, entrée and use. There are some properties that the user interface must-have.

- Avoid unnecessary elements and keep the interfaces simple.
- Purposefully use color and texture**:** make straight consideration to or redirect consideration away from items using color, texture, contrast.
- Use auto select choices as well as let the users select selections for well flexibility.
- Use common UI elements and build consistency: By using more comfortable common elements, for users and able to get things finished fast.
- Use layout to create order and clearness: Different sizes, fonts, and preparation of the text to help growth scalability, legibility and, readability.

3.12 Implementation

The implementation process uses to convert system specifications to the executable system. In this phase convert design to user-friendly views with front-end interfaces and back end systematic logics for performing functionalities. A comfortable language and appropriate tools were selected in the process of coding and development. The codes were written and settled in an understandable format. Remarks added for special logic to help for future implementations.

3.13 Implementation environment

The server-side environment and client-side environment were considered in the implementation process. These environment requirements divide into software requirements and hardware requirements. These requirements are described below.

3.13.1 Server environment

System hosted location and client-server architecture operators are considering in this environment. The server environment is used to host the developed system, which can handle all the data processing activities. The server should be a Mid-upper range server with 12 GB RAM and 300 GB Storage.

3.13.2 Client environment

Client software run in between the user and server be responsible for a user interface for user to cooperate with the server, for example, internet explorer like web browser act as client software. System is rich with the latest technologies and tools, so updated web browsers should use to avoid browser compatibility issues. The system can work with different hardware and software requirements, minimum system requirements are listed below. Table 3.2 identify hardware and software requirements of the client environment.

	Hardware	Software
1.	Core i7 upwards	Windows 2010 (64 bit) or higher Operating System
2.	8GB above RAM	Xampp 7.0
3.	At least 1Tb hard disk	Crystal Reports XIII
4.	Printer for report printing	

Table 3.2 Hardware and software requirements of client environment

3.14 Development tools and techniques

3.14.1 Tools

- PhpMyAdmin

Used to develop the database to the system. This was facilitated to implement the relevant tables, schemas, and relationships of the system

- Adobe photoshop

Used for photo editing purposes

- PHP

Used to pass the data to the database in a relevant manner. It was making an advance environment of the system.

- MVC architecture

Model, view and controller architecture is used to organize the system in a web-based environment.

- HTML

HTML was used to figure the base graphical user Interface of the system.

- CSS

CSS Styles was used to style interfaces more user approachable and attractive. Styles decided the appearance of the system.

- Javascript

Used JavaScript to code all the client-side validation.

- JQuery

This is also based on JavaScript was used to implement some features such as the pre-coded time picker module.

3.14.2 Network Architecture

The system contains central web server users connects to the server via the internal network. Users interact with the system via a web browser, and internal network protected by firewall.

Following figure 3.15 is shows the network architecture of the system

Figure 3. 15: Network architecture

3.15 Application Development Architecture

3.15.1 Main user interfaces

The below section of the document provides a limited graphical user interface of the system.

Login Page:

Following figure 3.16 is show the login page of the system

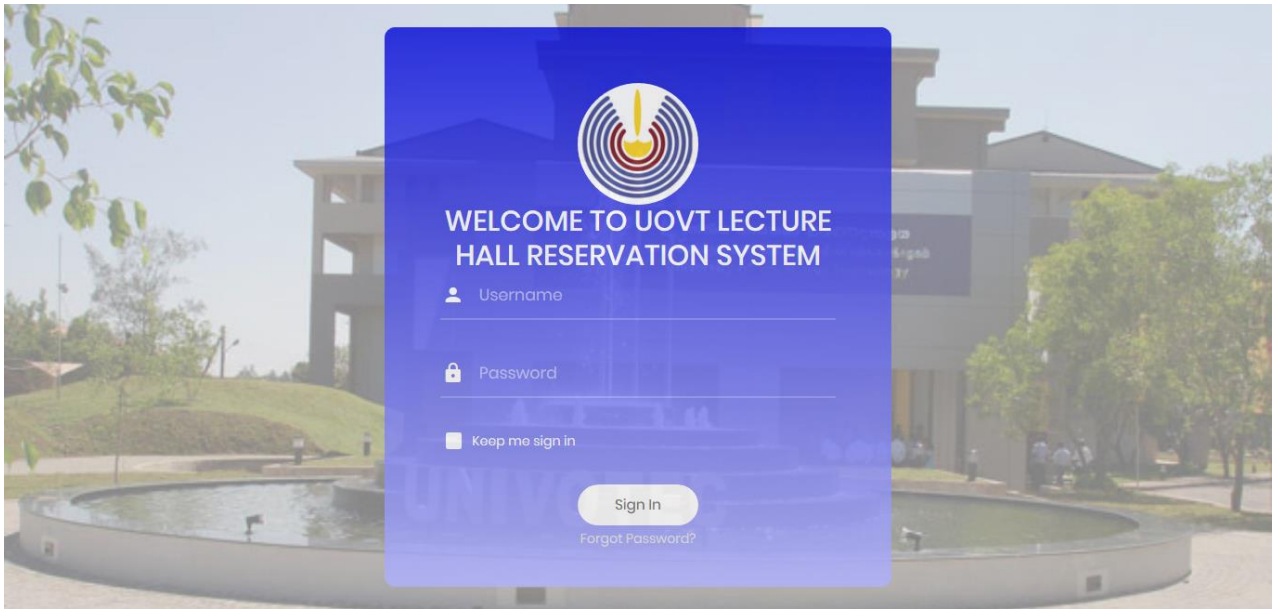


Figure 3.16 Login Page interface

Dashboard calendar view:

Following figure 3.17 is show the dashboard calendar of the system

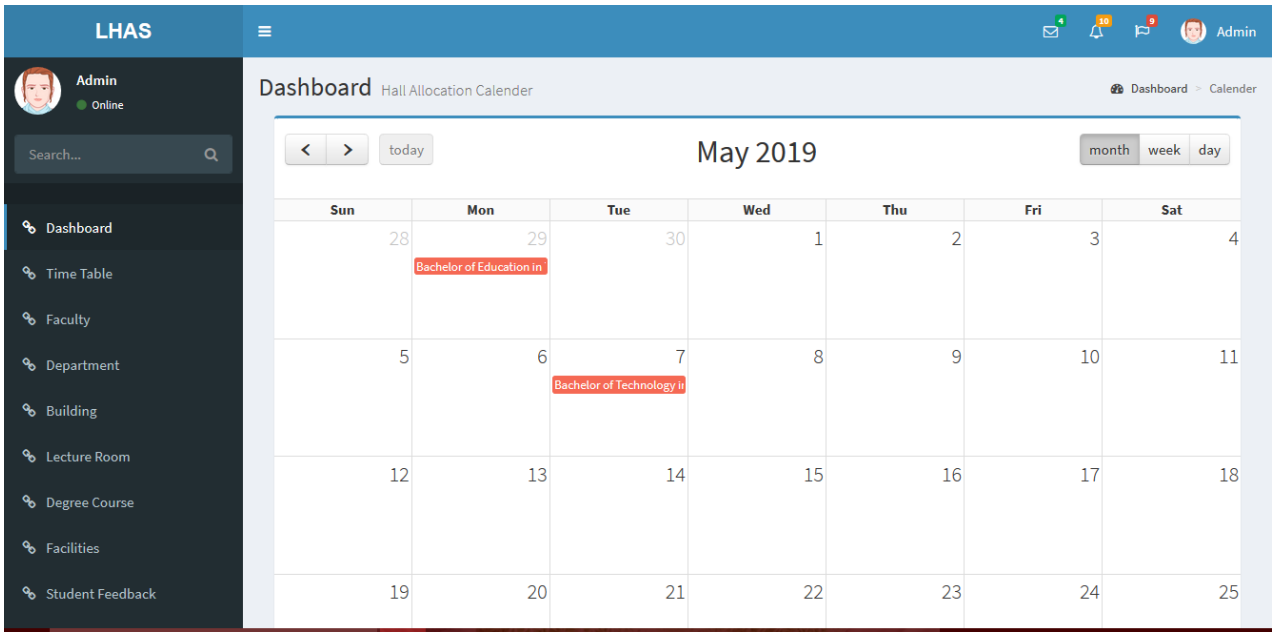


Figure 3.17 Dashboard calendar interface

Add new "Faculty" to the system:

Following figure 3.18 is show the addition of the new faculty to the system

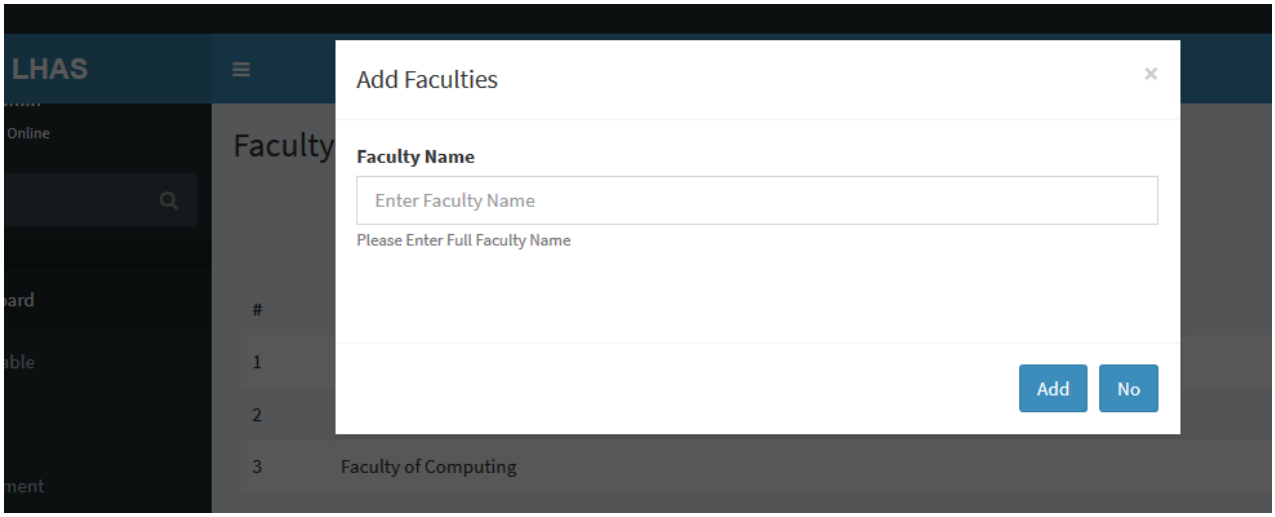


Figure 3.18 Add new faculty

Details of the Faculty:

Following figure 3.19 is show the faculty details of the system

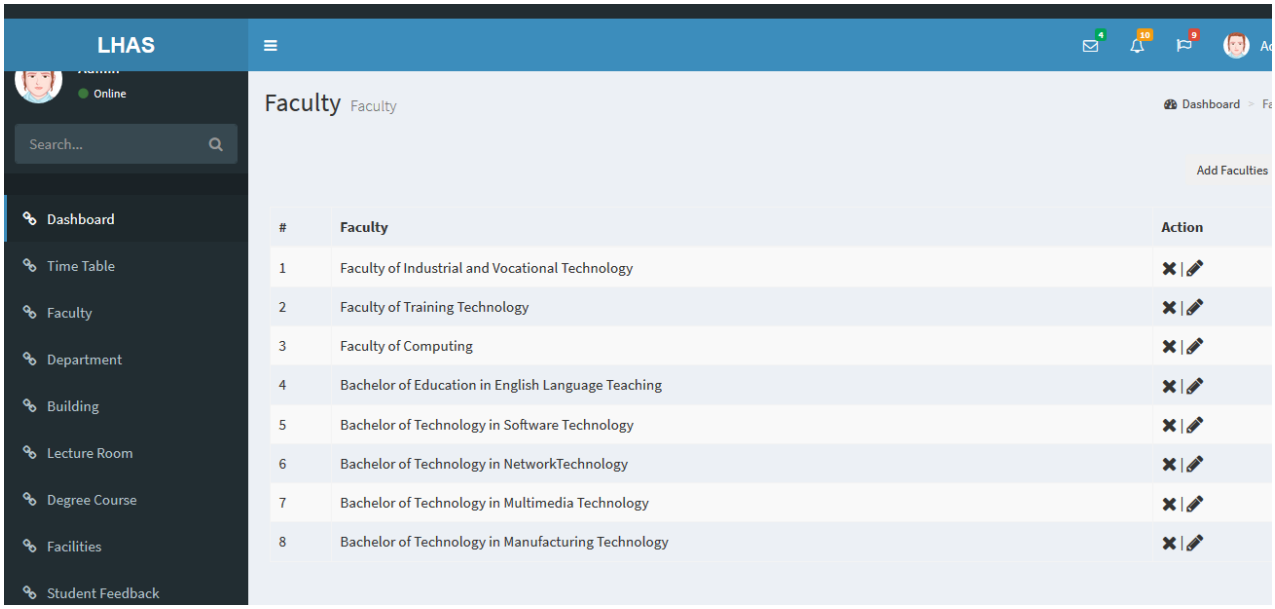


Figure 3.19 Faculty details

Student feedback form:

Following figure 3.20 is show the student feedback form of the system

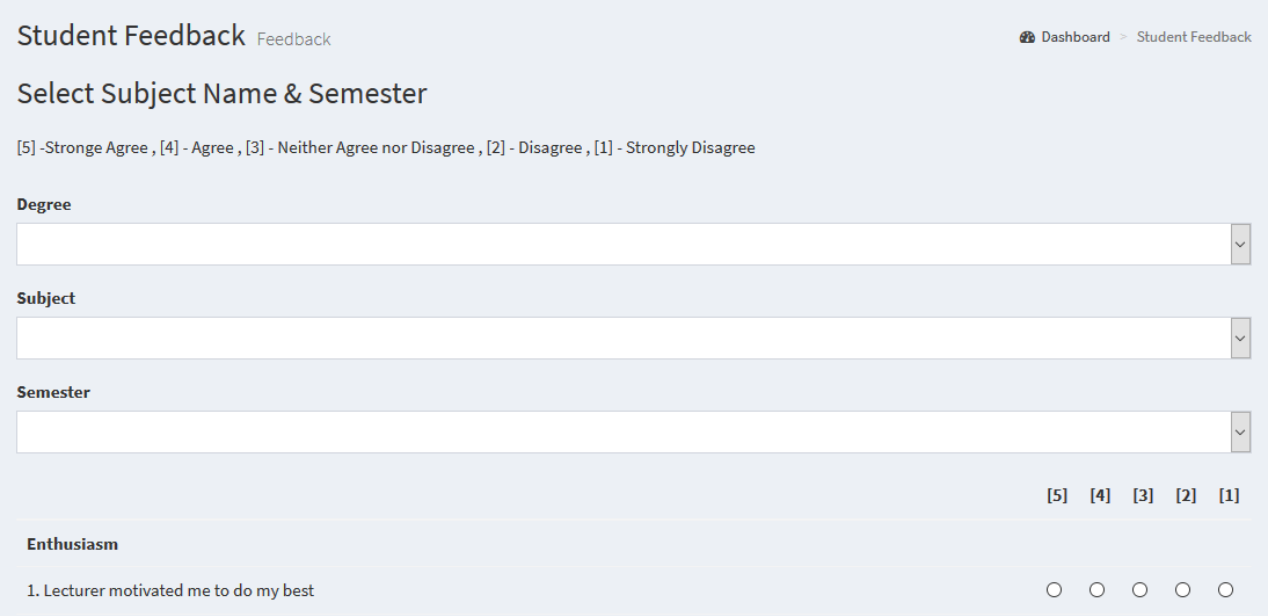


Figure 3.20 Student feedback

2.4 Summery

Web-based Lecture hall scheduling and timetabling system is very much effective for the universities because they are handling many students and lecturers regarding several manners. This chapter was described as the automation process of the system and the internal architecture of the system. Each figure is shown the specific functionalities and it helps to understand the system for everybody.

Finally, the proposed system combined all these features to satisfy user requirements.

CHAPTER 04

4 Evaluation

4.1 Introduction

Software testing [8] is well-defined as an action to check whether the real outcomes match the projected results and to verify that the software structure is defect free. It includes the implementation of a software module or system module to estimate one or more belongings of attention. Software testing also helps to recognize errors, breaches or missing necessities in dissimilar to the actual requirements. Furthermore, testing is required for system validation and verification. It directly describes are we built the right product for the right person and check whether the system meets its requirements.

4.2 Test procedures

Testing Procedures [9] are the testing practices, processes, and techniques used to ensure that the software application is tested and validating before release or deployment. In this system, structural testing techniques were tracked in this phase with "white box" testing. Functional testing techniques were used, which contains "black box" testing which tests the performance of a system or program. These techniques were exploited in different testing levels like unit testing.

The system followed the object-oriented methodology, so object-oriented testing was also carried out in this phase. Individual operations associated with objects were tested initially, followed by testing individual classes and clusters of objects, and finally test the system as a whole. User acceptance testing was completed in the client site, participation of a few staff members.

4.3 Test Plan and Test Cases

Testing begins with the implementation; code is reviewed while developing stage for testing. The test plan included all stages of testing and used as a guide for the complete testing process. Before the system implementation, the test plan was designed.

A test plan includes:

- Test objectives
- Schedule and logistics
- Test strategies and especially test cases.

Below tables tabulates some test cases for test basic functions:

Following table 4.1 shows the test cases for add new faculty to the system

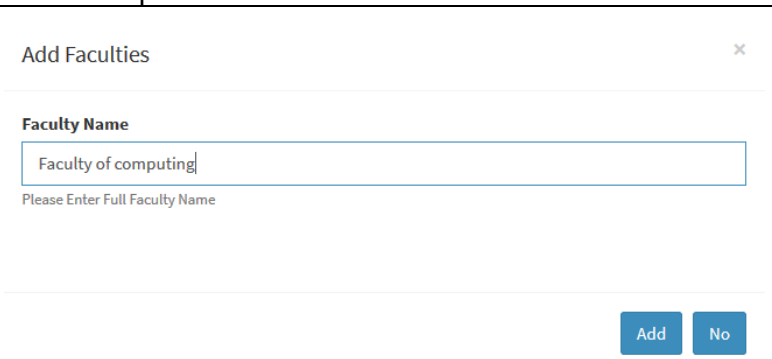
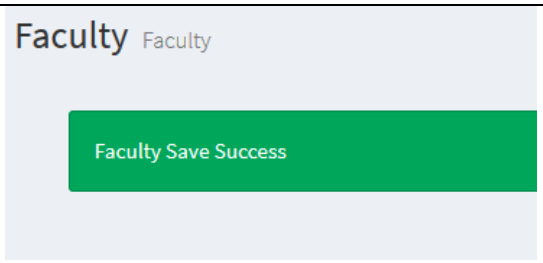
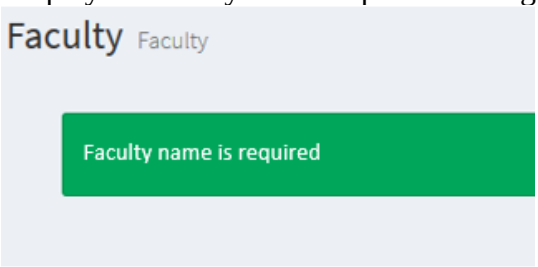
Test case Id	01		
Tested component	New Data Entry		
Module name	Add new faculty details		
Test case	Add a new faculty		
Expected output	Display message as successfully entered the Faculty data		
Test case description			
No	Test case	Actual output	Status
1	Enter new faculty name		Pass
2	Press save button		Pass
3	Press save button without data	Display manditory fields required message 	Pass

Table 4.1 Test case for add new faculty

Following table 4.2 shows the test cases for add new schedule to the system.

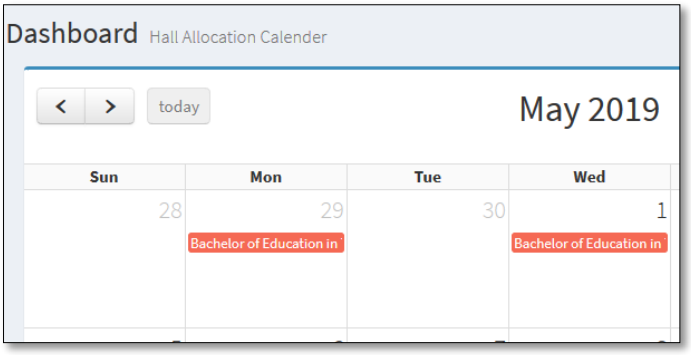
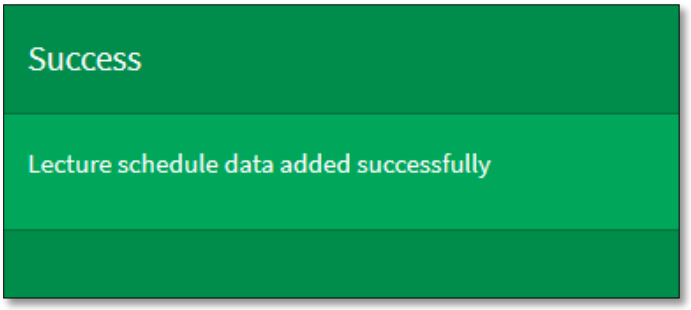
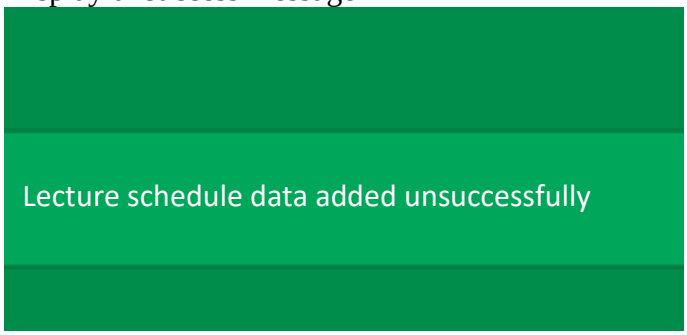
Test case Id	02	
Tested component	Calendar view	
Module name	Timetable scheduling	
Test case	Add a new allocation	
Expected output	Display details on the required date in the calendar	
Test case description		
No	Test case	Actual output	Status
1	Enter new allocation	Click on the calendar 	Pass
2	Press save button	Display success message 	Pass
3	Press save button without data	Display unsuccess message 	Pass

Table 4.2 Add new schedule

4.4 Acceptance Testing

Acceptance Testing is a close of software testing where a system is verified for satisfactoriness. This test determines to evaluate the system's agreement with the business necessities and evaluate whether it is suitable for delivery.

The user acceptance testing was conducted in the actual working environment of the University of Vocational Technology with actual data sets. Handlers were nominated from the academic staff and they were asked to work with related modules in the system according to their privileges. Activities, email generation were monitored while users working with the system. To consider as a successful project the outcome needs to be accepted by the end-users. Therefore, the system was handover to users for a test run with a questionnaire to capture the customer feedbacks. Therefore, a sample questionnaire was designed to give users.

4.5 Summary

After doing all the testing activities end-user submitted a statement of comments that indicating the system success or the system issues that have. When considering the user acceptance test results all the functional requirements given by the client were fulfilled and the user can carry out their tasks effectively and efficiently with the new system. Also, a few minor level modifications were made by the request of the client.

CHAPTER 05

5 Conclusion

5.1 Introduction.

This last chapter discusses the objectives archived and the future enhancements of the developed system. The University of Vocational Technology is planning to develop a module to record student attendance and it will be used for checking the 80% of attendance and issue the admission card for the semester examinations.

Their target is to link all the modules into the same web-based environment and ease the access for students and other users. The system was built using these following technologies MVC, object-oriented, HTML, CSS and JavaScript. This system introduces to maintain student's attendance, lecture hall scheduling, timetabling and exam preparation in the university.

5.2 Problems Encountered

From the beginning capturing the correct requirements was difficult because of the narrow IT knowledge of the client. Also, the manual process of the lecture hall scheduling was done by the involvement of several persons. One person was creating the timetable and forwarding it to the staff assistant and he or she manually checking the documents and allocate the lecture rooms, labs and create the daily timetable with allocation of the lecture hall to put the notice board. Always the users were trying to convert the manual process directly to the automated process. So, it was difficult to explain the reality to the users and break up their requested requirements and the traditional attitudes regarding the system functionalities.

Another major problem encountered during the development process is the lack of knowledge regarding the development tools, and languages to fulfill the project supervisor's suggestions. So, Online tutorials and books were used to capture the essential level of knowledge and took professional person guidance to develop the system more advance and reach to industrial level.

5.3 Lessons Learnt

The knowledge added all over the project was treasured. Because this system was related to the real environment and it was not an imaginational system. So, all the time I coordinated with the persons who are manually doing this process and took the real situation to develop my system. Opening from the feasibility studies, to the end of development this process gave incomparable experience in many ways. This project gave me a chance to test and develop the most important theories and pieces of knowledge learned through the MIT degree program. It also facilitated to learn very interesting new and updated technologies to improve system performance. Moreover, superior efforts were taken to escape the traditional implementation processes and learn the MVC architectural pattern to apply the system.

Additionally, working on the project encouraged me to improve technical skills as well as academic skills by cooperating with many persons from supportive fields. Also, this MIT program is to improve the ability to work under pressure and majorly manage other duties with higher studies.

5.4 Future enhancements

According to the client acceptance test approved in the evaluation phase, the client response was positive and considerable. Some future enhancements were requested to implement the system in a more advanced way. Following are the

- Mark the student's attendance by using new module
- Develop an admission issuing module
- Link the student's attendance to the admission issuing module.
- Add feature to upload the student's medical reports for absence.

6 REFERENCES

[1] uovt 2018, programs, uovt,viewed 9 July 2018 <http://univotec.ac.lk/index.php/programmes/>

[2] Dorien F.T.F.Meijer CluwenBSc 2015, Dynamic Room Allocation, utwente, viewed 5 July 2018 < http://essay.utwente.nl/67019/1/MeijerCluwen_AM_EEMCSv2.pdf>

[3] edustoreNg 2016, research, edustoreng, viewed 9 July 2018<https://edustore.ng/product/design-implementation-web-based-hall/>

[4]pdn 2018, eeu, eng.pdn, viewed 8 July 2018 < https://eng.pdn.ac.lk/eeu/booking/Web/?>

[5] otago 2018, study, viewed 6 July 2018 <https://www.otago.ac.nz/study/otago001326>

[6] hamburg 2018, raum-und-hoersaalvergabe, viewed 8 July 2018 < https://www.uni-hamburg.de/raum-und-hoersaalvergabe.html>

[7] ruhrbochum 2018, ruhr, viewed 5 July 2018 <https://www.ruhr-uni bochum.de/campusservice/hoersaalvergabe.html>

[8] G. J. Myers, C. Sandler, T. Badgett, The Art of Software Testing, John Wiley & Sons, Sep 23, 2011 - Business & Economics

[9] I.Sommerville, Software Engineering, 8th edition, Addison-Wesley, 2007.

[10] H. Mcheick and Y. Qi, “Dependency of components in MVC distributed architecture,” in Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference, 8-11 May 2011, pp. 6-10.

[11] Aurora Computer Studies, tutorial, viewed 4 April 2019 <https://www.auroracs.lk/>

7 APPENDIX A - Code Listing

Views – User – Login.php

The user interacts with the system through this view. View also displays error messages passed from the model=>controller.

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <title>Login</title>

  <!-- Tell the browser to be responsive to screen width -->

  <meta content="width=device-width, initial-scale=1, maximum-
scale=1, user-scalable=no" name="viewport">

  <!-- Bootstrap 3.3.7 -->

  <link rel="stylesheet" href="<?php echo base_url();
?>public/assets/bower_components/bootstrap/dist/css/bootstrap.min
.css">

  <!-- Font Awesome -->

  <link rel="stylesheet" href="<?php echo base_url();
?>public/assets/bower_components/font-awesome/css/font-
awesome.min.css">

  <!-- Ionicons -->

  <link rel="stylesheet" href="<?php echo base_url();
?>public/assets/bower_components/Ionicons/css/ionicons.min.css">

  <!-- Theme style -->

  <link rel="stylesheet" href="<?php echo base_url();
?>public/assets/dist/css/AdminLTE.min.css">

  <!-- iCheck -->

  <link rel="stylesheet" href="<?php echo base_url();
?>public/assets/plugins/iCheck/square/blue.css">



  <!-- Google Font -->

  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300
,400,600,700,300italic,400italic,600italic">
```

```
</head>
<body class="hold-transition login-page">
<div class="login-box">
  <div class="login-logo">
    <a href="../../index2.html"><b>Welcome to UoVT Lecture Hall Reservation</b></a>
  </div>
  <!-- /.login-logo -->
  <div class="login-box-body">
    <p class="login-box-msg">Sign in to start your session</p>

     <form class="form-horizontal" action="user_sign_in" method="post">
      <div class="form-group has-feedback">
        <input type="email" id="userName" name="tbx_username" class="form-control" placeholder="Email">
        <span class="glyphicon glyphicon-envelope form-control-feedback"></span>
      </div>
      <div class="form-group has-feedback">
        <input type="password" id="userPassword" name="tbx_password" class="form-control" placeholder="Password">
        <span class="glyphicon glyphicon-lock form-control-feedback"></span>
      </div>
      <div class="row">
        <div class="col-xs-8">
          <div class="checkbox icheck">
            <label>
              <input type="checkbox"> Remember Me
            </label>
          </div>
        </div>
        <!-- /.col -->
```

```
        <div class="col-xs-4">
          <button class="btn btn-primary btn-block btn-flat"
id="btn_login" name="btn_login" >Sign In</button>
        </div>
        <!-- /.col -->
      </div>
    </form>


     <a href="#">I forgot my password</a><br>
    <a href="register.html" class="text-center">Register a new
membership</a>


  </div>
  <!-- /.login-box-body -->
</div>
<!-- /.login-box -->


<!-- jQuery 3 -->
<script src="<?php echo base_url();
?>public/assets/bower_components/jquery/dist/jquery.min.js"></scr
ipt>
<!-- Bootstrap 3.3.7 -->
<script src="<?php echo base_url();
?>public/assets/bower_components/bootstrap/dist/js/bootstrap.min.
js"></script>
<!-- iCheck -->
<script src="<?php echo base_url();
?>public/assets/plugins/iCheck/icheck.min.js"></script>
<script>
  $(function () {
    $('input').iCheck({
      checkboxClass: 'icheckbox_square-blue',
      radioClass: 'iradio_square-blue',
      increaseArea: '20%' /* optional */
```

```
    });
  });
</script>
</body>
</html>
```

Controller – Login.php

Controller work as a messenger between the models and views. It gets the requests from view pass to model get responses.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Login extends CI_Controller {
     public function index()
     {
          // $this->load->view('includes/header');
          $this->load->view('user/login');
          // $this->load->view('includes/footer');
     }
     public function user_login(){
          //session_start();
          $username = $this->input->post('tbx_username');
          $password = md5($this->input->post('tbx_password'));
         $this->form_validation-
>set_rules('tbx_username','Username','required');
          $this->form_validation-
>set_rules('tbx_password','Password','required|max_length[20]');

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

                    $this->load->model('user_model', 'user');
                    $is_user = $this->user->check_login(
$username,$password );
```

```
if( $is_user == 'USER-OK' ){
                              // $this->load-
>view('includes/header');
                              // $this->load-
>view('includes/sidebar');
                              // $this->load-
>view('dashboard/index');
                              // $this->load-
>view('includes/footer');
                              redirect('/dashboard');

                         }else{
                              $this->load->view('user/login');
                         }
                    }else{

                         $this->load->view('user/login');
                    }
}
     public function user_logout(){
          $this->session->sess_destroy();
          $this->load->view('user/login');
     }
}
```

Model – User_model.php

Controller work as a messenger between the models and views. It gets the requests from view pass to model get responses.

```
<?php defined('BASEPATH') OR exit('No direct script access
allowed');

class User_model extends CI_Model {

    /**

     * Constructor

     */

    function __construct()

    {

        parent::__construct();

        $this->load->database();

    }

    function check_login( $username , $password ){


        $sql_get_user = "SELECT

                                *

                            FROM

                                tbl_user AS u

                            LEFT JOIN tbl_user_type ut ON
u.user_type = ut.user_type_id

                            WHERE u.user_name = '".$username."'
AND u.user_password = '".$password."' ";



        $query = $this->db->query( $sql_get_user );

        $result_user = $query->row();


        if( !empty( $result_user->first_name ) ){


            $_SESSION['LOGED-USER-ID'] = $result_user->user_id;

            $_SESSION['LOGED-USER-NAME'] = $result_user-
>first_name;
```

```
            $_SESSION['LOGED-USER-TYPE'] = $result_user->user_type;
            return 'USER-OK';
        }
        else{
            $this->session->set_flashdata('USER-LOGIN-NOT-OK','Invalid user name & password.Please try again using correct user name & password.');
            return 'USER-NOT-OK';
        }

    }
}
```

8 GLOSSARY

CSS (Cascading Style Sheet) - is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a mark-up language.

Database - is an organized collection of data for one or more purposes, usually in digital form.

Graphical User Interface - is a type of user interface that allows users to interact with electronic devices with images rather than text commands.

Internet - is a global system of interrelated computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide.

JavaScript - is a prototype-based, object-oriented client-side scripting language which is **dynamic**. It is also measured as a functional programming language.

JQuery - is a cross-browser JavaScript library designed to streamline the client-side scripting of HTML.

Object Oriented Development - is a typical method to software development based on objects and its occurrences

Structured Query Language - is a database computer declarative language considered for handling data in relational database management systems (RDBMS).

Unified Modeling Language (UML) - is a uniform general-purpose modeling language in the field of object-oriented engineering. This contains a set of graphic symbolization methods to create pictorial representations of object-oriented software-intensive systems.

Web Browser - is a software which allows the user to access WWW.