

**SEMI-AUTOMATED TIME SCHEDULING MANAGEMENT SYSTEM FOR
UNIVERSITY CLASSES**

BY

IKHTIAR AHMED

ID: 151-15-5257

AND

GRACE JOYDHAR

ID: 151-15-5075

AND

ABDULLAH-AL-MASUM

ID: 151-15-4952

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Saiful Islam

Lecturer

Department of CSE
Daffodil International University

Co-Supervised By

Subroto Nag Pinku

Lecturer

Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 09, 2018

APPROVAL

This Project titled “**Semi-Automated Time Scheduling Management System for University Classes**”, Submitted by Ikhtiar Ahmed, ID No:151-15-5257, Grace Joydhar, ID No: 151-15-5075 and Abdullah-Al-Masum, ID No:151-15-4952, to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 09 December, 2018.

BOARD OF EXAMINERS

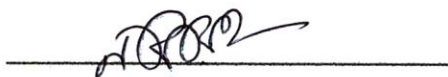


Dr. Syed Akhter Hossain

Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman

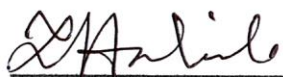


Narayan Ranjan Chakraborty

Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

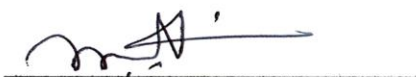


Md. Tarek Habib

Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Mohammad Shorif Uddin

Professor

Department of Computer Science and Engineering
Jahangirnagar University

External Examiner

DECLARATION

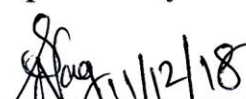
We hereby declare that all the information and statement true to our best of knowledge and this project has been done by us under the supervision of **Saiful Islam, Lecturer, and Department of CSE** Daffodil International University. We also notify that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



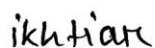
Saiful Islam
Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:



Mr. Subroto Nag Pinku
Lecturer
Department of CSE
Daffodil International University

Submitted by:



Ikhtiar Ahmed
ID: 151-15-5257
Department of CSE
Daffodil International University



Grace Joydhar
ID: 151-15-5075
Department of CSE
Daffodil International University



Abdullah-Al-Masum
ID: 151-15-4952
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

Primarily, we might want to express our true because of the Omnipotent ALLAH and Jesus Christ for the endowment of life, astuteness, and understanding he had given to us, a purpose behind our reality. In addition, to our families for the love furthermore, bolster they had given for an amazing duration.

In addition, Thanks to our beloved parents because they support us deeply Physically, Mentally and financially in the entire situation.

Secondly, we grateful and wish our profound our indebtedness to **Supervisor, Saiful Islam, Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Scheduling Algorithm*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

Thirdly, we also brightly appreciate to my respectable **Co-Supervisor, Subroto Nag Pinku**. Because his wisdom, intelligence, and idealness help our research. He also provided necessary data. We believe that this good work of this result is total of his excellent guidance and good assistance. Thanks to his recommendation and remarks.

Fourthly, we would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain**, Professor, and Head, Department of CSE, for his kind help to finish our project and to other faculty member and the staff of CSE department of Daffodil International University.

Finally, we would like special thanks to our beloved group mates because they play a crucial role in the important position. Without their hard-working, it is very impossible.

ABSTRACT

It already is known that every institution has a routine. On the other hand, a successful integrating management system plays a vital role in an organization. The developed system is a web-based application. This project was started from the experience of the teachers in the CSE department of the university is creating, managing and maintenance of class routine for every semester. The requirement specification was gathered from the practices of the department. The system was implemented HTML, CSS, and Bootstrap in the front-end part. On the other hand Laravel, PHP and Java script used in the back-end part. In addition, a mobile app on the Android platform will be developed in the near future. To find an exact and efficient solution, a genetic algorithm is used here. Because this algorithm maintains data according to the designed and checks all the errors efficiently. The system decreases the time of a student and teacher. Because the proposed system helps the student and teacher to find their routine dynamically. A teacher can view their schedule easily. The main purpose of this automated routine a student can access routine from anywhere and viewed teacher's details. This project gives an easy class schedule of all the university students. As a result, this project saves the time of all the students and teachers to find their schedule.

TABLE OF CONTENTS

CONTENTS	PAGE NO
Board of examiners	i
Declaration	ii
Acknowledgments	iii
Abstract	iv
CHAPTER	
CHAPTER 1: INTRODUCTION	1-4
1.1 Introduction	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Expected Outcomes	3
1.5 Report Layout	3-4
CHAPTER 2: BACKGROUND	5-7
2.1 Introduction	5
2.2 Related Works	5-6
2.3 Comparative studies	6
2.4 Scope of the problem	6
2.5 Challenges	6-7
CHAPTER 3: REQUIREMENTS SPECIFICATIONS	8-26
3.1 Business Process Model	8-9
3.2 Requirement Collection and Analysis	9
3.3 Use Case Model	10-11

3.4 Use Case Description	11-22
3.5 Entity Relationship Diagram	22-23
3.6 Logical Data Model	23
3.7 Data Flow Diagram	24-25
3.8 Design Requirements	26
CHAPTER 4: DESIGN SPECIFICATION	27-29
4.1 Front-End Design	27
4.2 Back-End Design	27-28
4.3 Interaction Design and UX	28-29
4.4 Implementation Requirements	29
CHAPTER 5: IMPLEMENTATION AND TESTING	30-49
5.1 Implementation of Database	30-32
5.2 Implementation of Front-End Design	33-47
5.3 Testing Implementation	47-49
5.4 Test Results and Report	49-50
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	51-52
6.1 Discussion and Conclusion	51
6.2 Scope of Future Development	51-52
APPENDICES	53-54
Appendix A: Project Reflection	53-54
Appendix B: Related Diagrams	54
REFERENCES	55-57

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1.1: The Business Proses Model	8
Figure 3.3.1: The Use-Case Diagram for the system	10
Figure 3.5.1: The Entity- Relationship Diagram	23
Figure 3.6.1: Logical Data Model	23
Figure 3.7.1: The Data Flow “Process” Notation	24
Figure 3.7.2: The Data Flow “Database” Notation	24
Figure 3.7.3: The Data Flow “External Entity” Notation	25
Figure 3.7.4: The Data Flow Notation	25
Figure 3.7.5: The Data Flow Diagram	25
Figure 5.2.1: Home Page of the System	33
Figure 5.2.2: View Routine Page of the System	34
Figure 5.2.3: About Us Page of the System	34
Figure 5.2.4: Login Page of the System	35
Figure 5.2.5: Admin Home Page of the System	35
Figure 5.2.6: Teacher Option Page	36
Figure 5.2.7: Import Teacher option Page	36
Figure 5.2.8: Add Teacher option Page	37
Figure 5.2.9: Modify Teacher option Page	37
Figure 5.2.10: Course Offer Option Page	38
Figure 5.2.11: Import Course option Page	38
Figure 5.2.12: Add Course option Page	39

Figure 5.2.13: Modify Course option Page	39
Figure 5.2.14: Classroom Option Page	40
Figure 5.2.15: Import Room option Page	40
Figure 5.2.16: Add Room option Page	41
Figure 5.2.17: Modify Room option Page	41
Figure 5.2.18: Time Option Page	42
Figure 5.2.19: Add Time option Page	42
Figure 5.2.20: Modify Time option Page	43
Figure 5.2.21: Static Routine option Page	43
Figure 5.2.22: Dynamic Routine option Page	44
Figure 5.2.23: Generate Routine option Page	44
Figure 5.2.24: View Routine option Page	45
Figure 5.2.25: Teacher Home Page	45
Figure 5.2.26: Room Booking option Page	46
Figure 5.2.27: Teacher's Routine Page	46
Figure 5.2.28: Teacher's Feedback option Page	47
Figure B.1: BPM Notations	54

LIST OF TABLES

TABLES	PAGE NO
Table 3.4.1: Use case description of Import Teacher's Details	12
Table 3.4.2: Use case description of Add Teacher's Details	12-13
Table 3.4.3: Use case description of Update Teacher's Details	13
Table 3.4.4: Use case description of Import Course Information	13-14
Table 3.4.5: Use case description of Add Course Details	14
Table 3.4.6: Use case description of Modify Course Details	14-15
Table 3.4.7: Use case description of Import Classes Details	15
Table 3.4.8: Use case description of Add Classroom Details	15-16
Table 3.4.9: Use case description of Modify Classroom Details	16
Table 3.4.10: Use case description of Import Section	16-17
Table 3.4.11: Use case description of Add Section Details	17
Table 3.4.12: Use case description of Add Time Details	17-18
Table 3.4.13: Use case description of Modify Time Details	18
Table 3.4.14: Use case description of View Routine	19
Table 3.4.15: Use case description of Set Static Routine	19-20
Table 3.4.16: Use case description of Generate Dynamic Routine	20
Table 3.4.17: Use case description of Book for Makeup Classes	21
Table 5.1.1: Database Structure for Admin Table	21-22
Table 5.1.2: Database Structure for Admin Details Table	22
Table 5.1.3: Database Structure for Course Table	30
Table 5.1.4: Database Structure for Room Table	30

Table 5.1.5: Database Structure for Section Table	31
Table 5.1.6: Database Structure for Takes Table	31
Table 5.1.7: Database Structure for Teacher Table	32
Table 5.1.8: Database Structure for Teacher Details Table	32
Table 5.1.9: Database Structure for Time Table	32

CHAPTER 1

INTRODUCTION

1.1 Introduction

A schedule management system is indispensable in each case of education. The manual and dynamic system is the most common and competitive for this management system. Manually works are time-consuming and need to ensure the data correctly. As opposed to, a dynamic scheduling system is a process, where the system modifies itself during runtime according to the requirement [17]. Nevertheless, we need a tiny help of the manual system, in that case, we named the system as a semi-automated system.

In this proposed project, we will present the system with three major sections. The first section is for Administrator. Administrators can always modify the information of the whole system. One Super Admin can create accounts of Admin with the default username and password. Every admin can change his/her personal details. The admin will have to give the required data for generating the routine.

The second section is for teachers. By default, teachers will get an activation URL with some information over Email. Teachers will request for activating their account with proper information [17]. They need to login to access the system with username and password they provided. They can only modify their information. Teachers will be able to book a room for extra classes and send the request to the admin for changing the class time, section or classroom number and so on. They will see their class routines in the system.

The last section and the most significant section is the user. Users can be anyone, who can only view or download the routine for printing or another academic purpose [11]. However, they need not create an account in the system.

In short words, Semi-Automated Time Scheduling Management System is the project, which will input the data from administrators and database after that it can generate a routine dynamically with the given data and little help of the manual system. However, it will ensure the accurate information and will be the user-friendly project.

1.2 Motivation

Firstly, when we admitted to the university, we saw that the routine management faculty preparing whole semester routine manually and it was very ticklish to create a routine manually. Every semester the entire student searches their class routine manually.

Secondly, we saw that the routine updates several times according to teachers requests or overlaps the time [17]. Multiple updates of routine make us confused about our class details. We also face difficulties to find makeup classes. Besides, we find several classes at the same time and in the same room.

Thirdly, managing time manually with so many constraints is not possible for any department especially like our Computer Science and Engineering Department.

Finally, we felt that it is so tough and decided to generate an automated routine. Then we started to study and research this system. After completing our initial research, we saw that it is possible to create a scheduling system using some data though it is an N-P Hard Problem. Then I decided to build the scheduling system.

1.3 Objectives

In this project, our main object is to develop a semi-automated time scheduling management system for university classes. Some specific objectives needed for this project.

- It is important to analyze the manual system. Then we need to collect the data, identify the system's requirements, and evaluate this data and requirements.
- Next is to design the model for this automated schedule management system. This application is web-based. So, firstly design the Front-end part for user interaction and next implement the system using PHP or other frameworks of PHP.
- The last part is to test and validate the system that to ensure that it runs successfully.

1.4 Expected Outcome

- We expected a complete system that generates the routine dynamically but due to some constraints, it is not possible to generate a fully automated routine.
- We expect that our proposed system will save time for management to generate routine manually.
- This proposed system will be effective for all three major users of the system, Administrator no need to generate manual routine, the teacher can view their individual routine and users or students are able to find a section/class wise routine.

1.5 Report Layout

Report Layout means specific information about every chapter of the project. As our project is a development project, we will describe our project in six chapters. The six chapters are Introduction, Background, Requirement Specification, Design Specification, Implementation, and Testing, and the last one is Conclusion and Future Scope.

In First Chapter, We just described the total view of our project including Introduction, Motivation, Objectives, Expected Outcomes and Report Layout.

In the Second Chapter, Illustrates information about the background of our project, like that related works, Comparative Studies, Scope of the problem and what kinds of challenges we were faced to create the routine.

In the Third Chapter, Describes some requirements and functionality of our project. The business process model and use case modeling were described how to work the routine exactly. In addition, requirement collection and design requirements were specified which types of data needed for this project.

In Chapter Four, Design Specification including Front-end Design, Back-end Design, Interaction Design and UX and Implementation Requirements. The types of language use in Front-end and Back-end that will also be described in this section.

In Chapter Five, after completing the Front-end and Back-end design then the next part is implementation and testing. How to implement using a database and how to

user interact the system given here. In addition, the testing result and report also viewed in this section.

In Chapter Six, Which is the last part of this project including two sections- Conclusion and Future Scope. The Future Scope section describes in detail about the features that can add in future with the system.

CHAPTER 2

BACKGROUND

2.1 Introduction

Every Educational institution has a routine to manage their classes. The university is a non-profitable well-known institution between Bangladesh. At first, it starts his journey only sixty students, but now the number of student at the university is among fifteenth thousand. The University provides under-graduate and master's program separately. The university already ensures their high-quality teaching ability.

An educational institution has different kinds of the department. In each year in every department has at most 3 semesters. Each department has one routine. They create routine manually. University routines depend on some condition, like that teacher, class time, number of periods and room [13]. It is very annoying to update routine manually in each semester.

As a Good university, it is more Significant to make routine automatically. The automated time scheduling management system to maintain the routine all the department of the university. Using an automated system to remove this problem. Then it is easier to update routine several times. In this system, students will see all information of the teachers. Automated routine managements system makes an educational institution is more effective and efficient.

2.2 Related Works

The work might be virtual calculation components, for example, strings, procedures or information streams, which are thus booked onto equipment assets. In this Process, there are huge complicated requirements. Most of the organization and institution have already tried to develop the routine [5]. Even in our institution also tried to develop the routine. However, they have faced the same problem and this was overlapping. We have some limitation on the overlap. When the teacher takes a class in one slot then he/she cannot take another class in the same section [17]. In addition, back-to-back classes not allowed for teachers but allowed for students. Besides

teacher cannot take more than three classes in a day. In the same, way we are confined the time and room overlapping.

2.3 Comparative studies

We know that every system has some quality and drawback. Our project name is “semi-automated time scheduling management system”. Our project is semi-automated because it is not possible to convert fully automated. That is the main deficiency of our project, besides overlapping is also a problem but it is possible to overcome.

On the other hand, there are many facilities for our project. We are able to remove the overlap of every section/room and teachers [11]. This system reduces the routine committee’s work and students are easily find their routine using a semester or course code. In addition, the student can print the routine easily and show the teacher’s information.

2.4 Scope of the problem

The project Semi-Automated time scheduling management system for university classes is helpful for every educational institution. Because our proposed system is very easy to find routine automatically. Easy to find routine teacher’s details in every semester. In this system also remain a manual system. However, our project is web-based, so the user interface of this system is easy to operate all the students [14]. The system easily creates an automated routine and maintain the system. We think the educational institution will be interested in our project, and they can create an automated system for their institution

2.5 Challenges

Our proposed system is to generate an automated routine. To develop, the system we are faced with some challenges. To generate the dynamic routine is so tough. We convert the static routine to dynamic. There are some challenges that we already faced in generating the dynamic routine [14]. Some challenges are –

- We are capable of every room must be used for every time slot.
- Book a room dynamically/powerfully.
- Not covering every section/room.
- None overlaps the Trainer/educators.
- Not persistent class more than three.
- To keep up the consecutive classes.
- To upload the teacher's information and initial.
- To alter the system as user requirements.
- To explore the routine based on teacher initial, course code and semester.

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 Business Process Model

The Elaboration of BPM is Business Process Model. Business Process Model refers to the graphical user interface [1]. It indicates how the process exactly works.

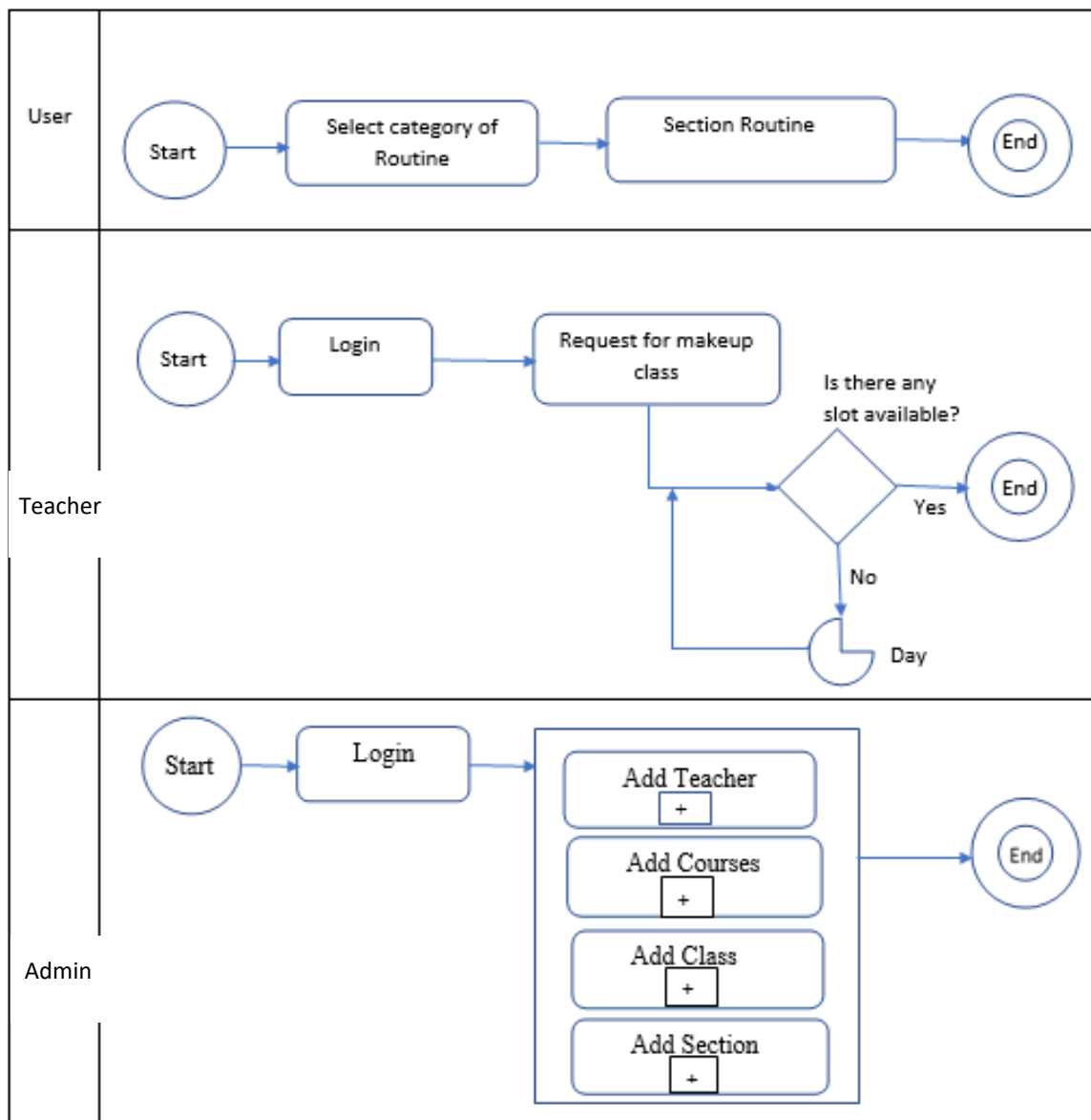


Figure 3.1.1: The Business Process Model for the Proposed Project.

BPM provides you a Clear thought about your idea. BPM likewise gives you a consistency of your application. It reduces the wasteful capacity. BPM helps the project to decrease the cost, increment speed and increment the nature of the process.

In the BPM, every process started at start node and finished the process at the end node. In this Figure 3.1.1, we can see that user, teacher, and admin started at start node and finished at end node. We can definitely say that Admin can maintain the full process, and teacher can request for makeup classes if any slot available. On the other hand, the user can view the routine easily without login. Login is mandatory for admin and teacher.

Therefore, Business Process Model easily describes the full project in a simple graphical interface.

3.2 Requirements Collection and Analysis

Semi-automated routine management system creates an environment, which helps the teacher, student, and everybody. Before taken this project at first we meet our project committee. We discuss the final year project more briefly. Then they suggested us about this topic. Because this project is more useful for university classes. Then we gone to our course teacher and suggested the same topic [6]. After all discussion, we started a study about dynamic routine, but we see that is so difficult to generate dynamically.

After analyzing the genetic algorithm, we knew that the genetic algorithm used for decision-making and problem-solving. The genetic algorithm also used in business purpose and organization to minimize the cost and time. The genetic algorithm works in six steps they are starting, evaluation, selection, crossover, mutation, test, and report. The genetic algorithm always gives an accurate result and so on[16]. Though some limitations stay there genetic algorithm is the best and easy way to solve any problem and for decision-making. So after analyzing the genetic algorithm, we can make sure that it is possible to generate the dynamic routine.

3.3 Use Case Model:

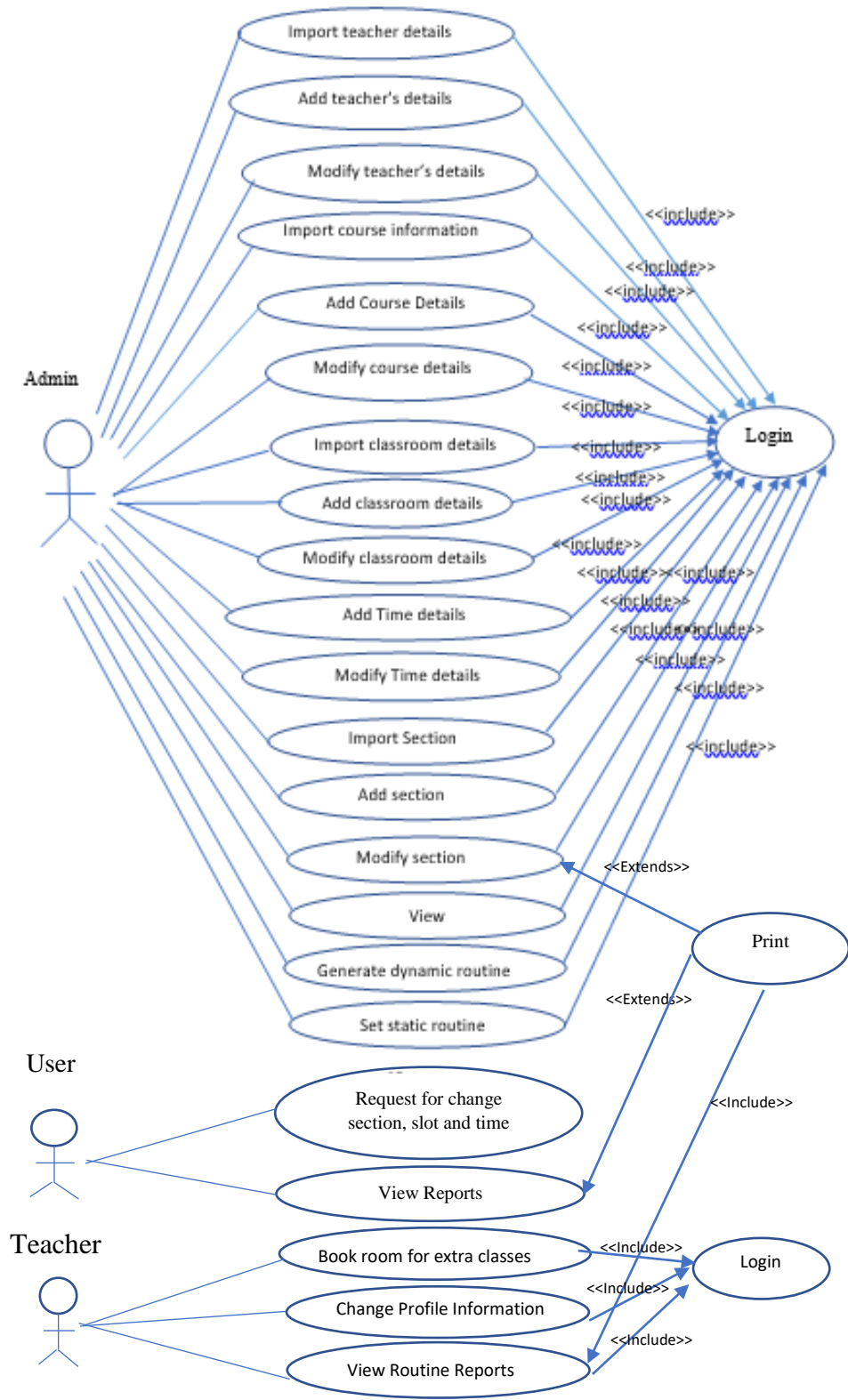


Figure 3.3.1: The Use-Case Diagram for the System.

Use case diagram is one kind of scenario, which in your system or framework connect with the general people, organization and external system. Normally use case diagram represents the relationship between the user and application in various circumstance. A use case can abridge the whole project for the user easily. We can see the entire project in a solitary diagram using a use case diagram. Use case diagram helps to develop the project. The oval shape represents the use case and stick figure represent the actor or user in the process. The proper use case diagram and activities used to reach the actual destination.

In this project, user or students have no required to login. They can search routine easily. Students can search routine as a semester wise or full routine. Students can also request for change section, slot and time. They can report having start and end time. Students can also view the full routine and print the routine.

In this project, to perform any type of activities the teacher must be login before. The teacher can request for makeup classes and change their password.

3.4 Use Case Description:

Normally, an actor to accomplish a complete project transaction performs the use case description [21]. Besides use case, description defines the actual task of the user or actor. At first, it has initiated by an actor or user and given the value of an actor. There are distinctive types of use case description given the underneath. If we focus on the table, we can say that there are various types of data given the table. They are at first, use case name, actor, precondition, user, internal path, exceptional path, and note. Use case name characterizes name of the procedure or use case. Actor or user indicate who will be the user of the system. Pre-condition are those things that should be set up before the utilization case begin. The internal path is the most successful path of the process. There are different kinds of task perform here. Predominantly the progression of the procedure. Exceptional is defined which types of error occurred of previous steps like that the internal path. Furthermore, the last one notes, it identifies who will create or update the process and when he updated the table that is included in the note part.

There is some model given the below.

Table 3.4.1: Use case description of Import Teacher's Details

1	Use Case Name:	Import Teacher's Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Select "Teacher" Option 4.2: Click on "Add Teacher" 4.3: Click the "Browse" button 4.4: Select file from drive to upload 4.5: Click on "Select" Button 4.6: Click the "Import" Button
5	Exceptional Path:	4.4.1: File type csv or excel and fill can't be empty 4.6.1: File should process according to requirement given.
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 20.11.17, 2:40 PM. by Abdullah-Al-Masum

Table 3.4.2: Use case description of Add Teacher's Details

1	Use Case Name:	Add Teacher details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on "Teacher" option 4.2: Click on "Add Teacher" 4.3: Enter the Full Name 4.4: Enter the teacher's Initial 4.5: Enter Employee ID 4.6: Enter E-Mail Address 4.7: Click the "Submit" button

5	Exceptional Path:	4.3.1: Name can contain the only character, dot (.), Hyphen (-). 4.4.1: Teacher initial contains the only Character 4.5.1: Employee ID contains the only integer. 4.6.1: Valid E-mail address as “example@domain”
6	Note:	Description created by Ikhtiar Ahmed on 23.08.17, 2:30 PM Modified on 23.07.18, 10:30 AM By Abdullah-Al-Masum

Table 3.4.3: Use case description of update Teacher’s Details

1	Use Case Name:	Update Teacher’s Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click the “Teacher” option 4.2: Click on “Modify Teacher” Button 4.3: Search the Teacher Initial 4.4: Select the desired Initial 4.5: Enter the Edit option 4.6: Enter Full Name 4.7: Enter E-Mail Address 4.8: Click the Update button
5	Exceptional Path:	4.6.1: Name can contain the only character, dot (.), Hyphen (-). 4.7.1: Valid E-mail address as “example@domain”
6	Note:	Description Created by Ikhtiar Ahmed on 23.08.17, 2:30 PM Modified on 17.07.18, 10:40 AM by Abdullah-Al-Masum

Table 3.4.4: Use case description of Import Course Information

1	Use Case Name:	Import Course Information
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on “Course Offer” Button 4.2: Click the “Import Course” button 4.3: Click the “Browse” button 4.4: Select file from drive to upload 4.5: Click on “Select” Button 4.6: Click the “Import” Button
5	Exceptional Path:	4.3.1: File type csv or excel and fill can’t be empty. 4.5.1: File should process according to requirement given.
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 17.07.18, 2:40 PM by Abdullah-Al-Masum

Table 3.4.5: Use case description of Add Course Details

1	Use Case Name:	Add Course Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on “Course Offer” Button 4.2: Select the “Add Course” option 4.3: Enter course-name 4.4: Enter course-code 4.5: Enter credit-hour 4.6: Enter the “Submit” 4.7: Select Course Code

		4.8: Enter Sections 4.9: Click “Submit” Button
5	Exceptional Path:	4.2.1: Name can contain only character 4.3.1: Can Contains character and integer 4.4.1: Contains only integer
6	Note:	Description Created by Ikhtiar Ahmed on 23.08.17, 2:40 PM Modified on 17.07.18, 11:40 AM by Abdullah-Al- Masum

Table 3.4.6: Use case description of Modify Course Details

1	Use Case Name:	Modify Course Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on “Course Offer” option 4.2: Click the “Modify Course” button 4.3: Select the “Edit” Link 4.4: Enter Course-name 4.5: Enter Course-code 4.6: Select Course-credit 4.7: Enter the “Update” Button
5	Exceptional Path:	4.4.1: Name can contain only character 4.5.1: Can Contains character and integer 4.6.1: Can Contains only Character
6	Note:	Description Created by Ikhtiar Ahmed on 23.08.17, 2:40 PM Modified on 17.07.18, 11:40 AM by Abdullah-Al- Masum

Table 3.4.7: Use case description of Import Classroom Details

1	Use Case Name:	Import Classroom Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on “Classroom” option 4.2: Click the “Import Room” button 4.3: Click the “Browse” button 4.4: Select file from drive to upload 4.5: Click on “Select” Button 4.6: Click the “Import” Button
5	Exceptional Path:	4.5.1: File type csv or excel and fill can’t be empty 4.6.1: File should process according to requirement given.
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 17.07.18, 2:40 PM by Abdullah-Al-Masum

Table 3.4.8: Use case description of Add Classroom Details

1	Use Case Name:	Add Classroom Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on “Classroom” option 4.2: Click the “Add Room” button 4.3: Enter Room-No 4.4: Enter Tower 4.5: Enter Class-Type 4.6: Enter the “Submit” Button
5	Exceptional Path:	4.3.1: Can contain character and integer 4.4.1: Can contain only integer

		4.5.1: Character only
6	Note:	Description Created by Ikhtiar Ahmed on 23.08.17, 12:30 PM Modified on 17.07.18, 11:50 AM by Abdullah-Al-Masum

Table 3.4.9: Use case description of Modify Classroom Details

1	Use Case Name:	Modify Classroom Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on “Classroom” option 4.2: Click the “Modify Room” button 4.3: Click the “Edit” link 4.4: Enter Room-No 4.5: Enter Tower 4.6: Enter Class-Type 4.7: Enter the “Modify” Button
5	Exceptional Path:	4.4.1: Can contain character and integer 4.6.1: Must Select Either “L” or “T”
6	Note:	Description Created by Ikhtiar Ahmed on 23.08.17, 12:30 PM Modified on 18.07.18, 11:50 AM by Abdullah-Al-Masum

Table 3.4.10: Use case description of Import Section

1	Use Case Name:	Import Section
2	Pre-Condition:	Login
3	Actor:	Admin

4	Internal Path:	4.1; Select the “Course Offer” option 4.2: Select “Import Course” Bottom 4.3: Enter the Browse button 4.4: Select file from drive to upload 4.5: Enter the “Import” Button
5	Exceptional Path:	4.3.1: File type csv or excel and fill cannot be empty. 4.4.1: File should process according to requirement given.
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 17.07.18, 2:40 PM by Abdullah-Al-Masum

Table 3.4.11: Use case description of Add Section Details

1	Use Case Name:	Add Section
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click on “Course Offer” option 4.2: Select “Add Course Option” 4.3: Select the Course Code 4.4: Enter the Section-Nam 4.5: Enter the “Submit”
5	Exceptional Path:	4.4.1: Name can contain only character
6	Note:	Description Created by Ikhtiar Ahmed on 23.08.17, 2:30 PM Modified on 17.07.18, 11:55 AM by Abdullah-Al-Masum

Table 3.4.12: Use case description of Add Time Details

1	Use Case Name:	Import Time Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Select the ‘Time’ option 4.2: Select the ‘Add Time’ Button 4.3: Enter Start-Time 4.4: Enter End-Time 4.5: Enter Duration 4.6: Enter Break Start-Time 4.7: Enter Break End-Time 4.8: Click the ‘Submit’ button
5	Exceptional Path:	4.3.1: Only Can contain Time 4.4.1: Only Can contain Time 4.5.1: Only Can contain Time 4.6.1: Only Can contain Time 4.7.1: Only Can contain Time
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 18.07.18, 2:40 PM by Abdullah-Al-Masum

Table 3.4.13: Use case description of Modify Time Details

1	Use Case Name:	Time Details
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Select the ‘Time’ option 4.2: Select the ‘Modify Time’ Button 4.3: Enter Start-Time 4.4: Enter End-Time

		4.5: Enter Duration 4.6: Enter Break Start-Time 4.7: Enter Break End-Time 4.8: Click the “Submit” button
5	Exceptional Path:	4.3.1: Only Can contain Time 4.4.1: Only Can contain Time 4.5.1: Only Can contain Time 4.6.1: Only Can contain Time 4.7.1: Only Can contain Time
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 18.07.18, 2:40 PM.

Table 3.4.14: Use case description of View Routine

1	Use Case Name:	View
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Select the Routine 4.2: Enter the Print option 4.3: Click the “view” option
5	Exceptional Path:	4.1.1: None
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 18.07.18, 2:40 PM.

Table 3.4.15: Use case description of Set Static Routine

1	Use Case Name:	Set Static Routine
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Enter set the static routine 4.2: Select “Add/Modify” Button 4.3: Enter the Semester 4.4: Enter the Department 4.5: Enter the select day 4.6: Enter the select time 4.7: Enter Routine Details 4.8: Select “Ok” Button
5	Exceptional Path:	4.3.1: can contain only character 4.5.1: Can contain character 4.3.1: Can contain time only
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 18.07.18, 2:40 PM.

Table 3.4.16: Use case description of Generate Dynamic Routine

1	Use Case Name:	Generate Dynamic Routine
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Click the Dynamic Routine 4.2: Enter the Semester 4.3: Enter the Department 4.4: Click on “Generate Routine” 4.4: Click the “View routine”
5	Exceptional Path:	4.1.1: Section can contain only character

		4.2.1: Can contain character 4.3.1: Can contain an integer
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 18.07.18, 2:40 PM.

Table 3.4.17: Use case description of Book for Makeup Class

1	Use Case Name:	Request For Makeup Class
2	Pre-Condition:	Login
3	Actor:	Admin
4	Internal Path:	4.1: Check the Empty place 4.2: Click on “Makeup Classes” option 4.3: Enter the “Book Now” for a Class 4.4: Enter the Information of Class 4.5: Click the “Submit” Button
5	Exceptional Path:	4.3.1: Can contain the only character
6	Note:	Description Created by Ikhtiar Ahmed on 24.08.17, 2:40 PM Modified on 18.07.18, 2:40 PM.

3.5 Entity Relationship Diagram

Entity relationship diagram is one kind of summary of all data structure of system [22]. For this system, the entity relationship diagram represents the database that is important for this system. The relationship between tables is also shown with the entities in this diagram.

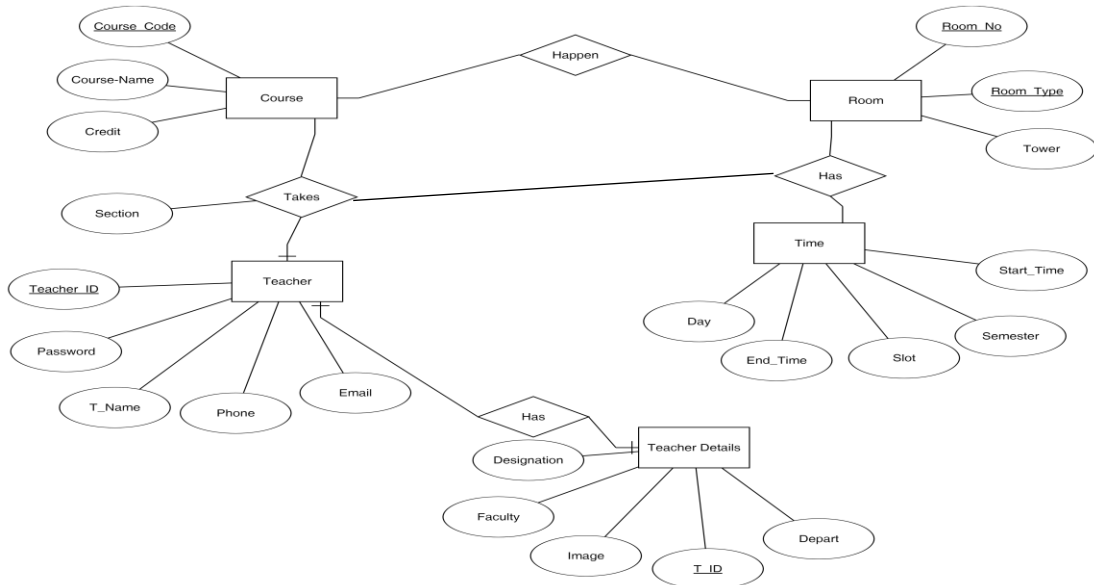


Figure 4.5.1: Entity Relationship Diagram

3.6 Logical Data Model

The logical data model is the process of representing data architecture and organization in a graphical way [23]. Logical data Model provides all the entity relation of a system. It is specific problem domain expressed independently of a particular database management product or storage technology.

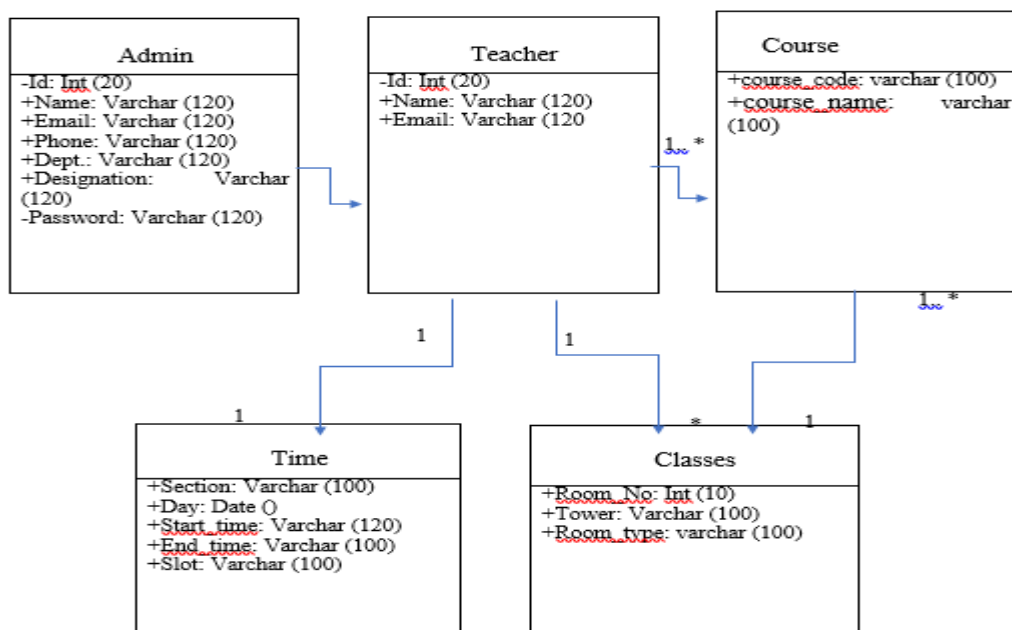


Figure 3.6.1: Logical Data Model of the System

3.7 Data Flow Diagram

A Data Flow Diagram is one kind of chart, which represent the flow of data through an information system [8]. Data Flow Diagram is the initial steps to create an overview of the system without going to a brief description.

DFD utilizes characterized images like square shapes, circles, and bolts alongside content marks [12]. A DFD indicates what types of data will be input and output from the process, how the information will progress through the system, and where the information will put away.

There are four types of notation of Data Flow Diagram:

- Process
- Database/File
- External Entity
- Data Flow

Process: An activity that changes or transforms data flows. Since they transform incoming data into outgoing data, all processes must have inputs and outputs on a DFD.[9]

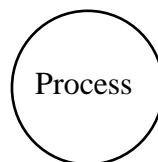


Figure 3.7.1: Data Flow Process Notation

Database/File: Database or File simply used to store the information or data of the transforming system. It is also known as rectangular.

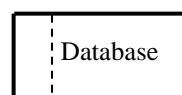


Figure 3.7.2: Data Flow Process Notation

External Entity: Also known as, actors, and terminators, external entities produce and consume data that flows between the entity and the process is a figure. These data flows are the inputs and outputs of the DFD.

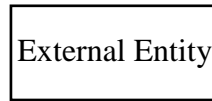


Figure 3.7.3: Data Flow Process Notation

Data Flow: Data Flow Used to movement of data between external entities, processes and data stores [8]. It also represented with an arrow symbol, which indicates the direction of flow.



Figure 3.7.4: Data Flow Process Notation

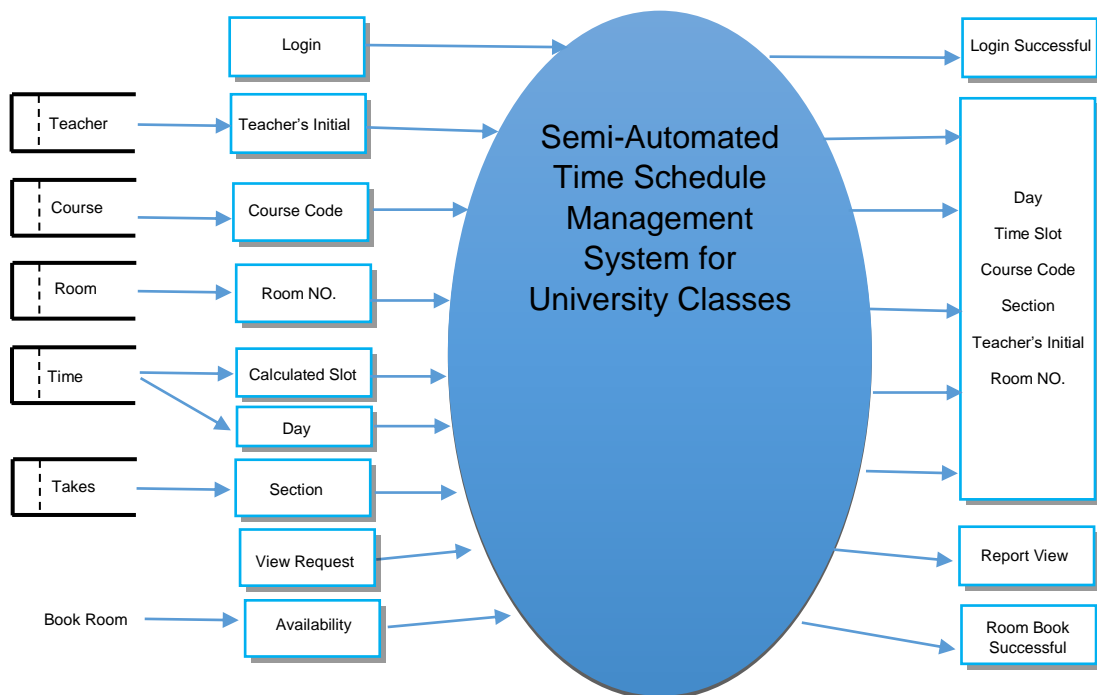


Figure 3.7.5: Data Flow Diagram

3.8 Design Requirements

There are some requirements to generate the routine management system. They are- Hardware Requirements and Software Requirements.

Hardware Requirements:

Hardware requirements or specification means what kinds of hardware needed to operate the project.[25]

- Processor: Platinum III or Higher
- RAM: 512 Mb or more
- Hard Disk:500 Mb or more
- Screen Resolution at least 640 * 480 or above.
- Support for printer
- Required Internet Connection

Software Requirements:

Though the system is a web-based application, so the client side is required-

- Web server: Apache Server (XAMPP)
- Operating system: Windows/Linux
- Database: MYSQL
- Sublime Text for Editing
- Internet Browser For View

Language:

- HTML
- CSS
- Bootstrap
- PHP
- Laravel
- Java

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

The Front-end is one kind of application that user interacts the application directly. Front-end convert the data to the graphical interface for the user to see and interact the data easily.

Front-end is the viewer site of the user. Our Project “Semi-Automated Routine Management System” is one kind of web-based application. This Application also has a two part. One is front end and another is back-end.

Front-end part including some language. They are HTML, CSS, and JavaScript. HTML refers to the HyperText Markup Language and CSS refer to the Cascading style sheet. On the other side, JavaScript is one kind of scripting language.

HTML and CSS used for creating a structure for the application [3].HTML can derive the header, footer. In addition, how the text display in the application and how the media and images appear.

HTML and CSS is the main structure of the website[3]. If you have no knowledge about HTML and CSS, you cannot create a website. JavaScript added a more functionality to your website.

After completing, the front-end part user can interact the site, clicking some button and input something like that your name, mail, etc.

4.2 Back-end Design

The back-end is also a program that supports indirectly to the front-end site. The back-end also called the server-side. Because it helps to how the site works, updates and changes. The back-end is a program that user cannot see the browser.

We already told that Our Project name is “Semi-Automated Routine Management System” is a web-based application. The application works dynamically. Without back-end, program automation is very impossible [16]. Some people who are work in the back-end they called to the back-end programmer.

Back-end usually consists of three parts. They are an application, server, and a database. In this system, all information stored in the database and then create a server.

Back-end also controls the security of the application. Dynamic sites are always changing updates in real time. For dynamics sites, a database is required to work properly.

PHP is server-side scripting language. PHP refers to the hypertext preprocessor. In this project, we have also used PHP. Because PHP runs different kinds of the platform and supports many databases like that MySQL, Oracle etc [2].

4.3 Interaction Design and UX

Goal Driven Design: User research is very important to know the real scenarios and related problems [19]. That is the actual person who was used the system. What we want to achieve through our persona should be well defined by user stories. User stories give logic to our thoughts making the design perfect.

User Expectations: This is the same case when a user comes to an application and performs some actions expecting some reactions from the product. If a user fills some form and clicks ‘Done’, he expects some confirmation alert or an error message if something is wrong with any of the fields he expects some alert or error message.

Usability: There should be less attention to figure out how to use the application so that the user can focus on completing the task. The product should be very easy to use and easy to understand its functions.

Utility: It is the most important factor focusing on “Does user find the function useful?” or “Is the function important or is it designed to give high-quality UI factor only?[19]’ The function should solve some problem or purpose then only it is useful to the user. There should be a very logical reasoning behind the function to complete a valuable task.

Affordance and Signifier: Function that speaks for itself and suggests its own use is affordance. Signifier hints at the affordance of the particular function and suggests how to use.

4.4 Implementation Requirements

Requirements for implementing Front End: This system is implemented with Web programming. For this, I have used Sublime Text to write the code and the languages are HTML, CSS and PHP.

Requirements for implementing Back End: To implement the database I have used XAMPP server to connect the database with PHP and implement database design with MySql database.

Summary of languages, Frameworks, and tools

- HTML
- CSS
- PHP
- MySql
- JavaScript
- Laravel 5
- Bootstrap
- XAMPP

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

Data should be stored in a database for reuse the data. To implement the database, I have used XAMPP Apache and MySQL server to run the web application by localhost [24]. The Database Tables are Given Below:

Table 5.1.1: Database Structure for Admin Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id		int(100)	No	None			Change Drop More
<input type="checkbox"/>	2	password		varchar(255)	latin1_swedish_ci	No	None		Change Drop More
<input type="checkbox"/>	3	name		varchar(255)	latin1_swedish_ci	No	None		Change Drop More
<input type="checkbox"/>	4	Email		varchar(255)	latin1_swedish_ci	No	None		Change Drop More

Table 5.1.1 shows the admin table which has four entities where id is the primary key and used to login to the system with another entity named password.

Table 5.1.2: Database Structure for Admin Details Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id		int(100)	No	None			Change Drop More
<input type="checkbox"/>	2	designation		varchar(255)	latin1_swedish_ci	No	None		Change Drop More
<input type="checkbox"/>	3	department		varchar(255)	latin1_swedish_ci	No	None		Change Drop More
<input type="checkbox"/>	4	cell		varchar(100)	latin1_swedish_ci	No	None		Change Drop More
<input type="checkbox"/>	5	image		longblob	No	None			Change Drop More

Table 5.1.2 shows the admin details table which has five entities where id is the foreign key, used to show the details about the admin.

Table 5.1.3: Database Structure for Course Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	code	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 2	name	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	credit	int(10)			No	None			Change Drop More

Table 5.1.3 shows the Course table which has three entities where the code is the primary key and shows the name of the course and credit with code.

Table 5.1.4: Database Structure for Room Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	number	int(10)			No	None			Change Drop More
<input type="checkbox"/> 2	type	varchar(10)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	tower	varchar(20)	latin1_swedish_ci		No	None			Change Drop More

Table 5.1.4 shows the room table which has three entities. Used to see the building name, room type with the specific room number.

Table 5.1.5: Database Structure for Section Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	C_code	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 2	Sections	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Table 5.1.5 shows the Section table which has two entities where C_code is foreign key and shows the sections with its code.

Table 5.1.6: Database Structure for Takes Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	section	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 2	t_id	int(100)			No	None			Change Drop More
<input type="checkbox"/> 3	c_code	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Table 5.1.6 shows that, Takes table has three entities where the section is primary key with other two foreign keys.

Table 5.1.7: Database Structure for Teacher Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(100)			No	None			Change Drop More
<input type="checkbox"/> 2	name	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	initial	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 4	password	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Table 5.1.7 shows the teacher table has four entities where id and teacher initial both are the primary key and use to log in with a specific password.

Table 5.1.8: Database Structure for Teacher Details Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(100)			No	None			Change Drop More
<input type="checkbox"/> 2	designation	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	department	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 4	cell	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 5	email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 6	priority	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 7	image	longblob			No	None			Change Drop More

Table 5.1.8 shows that Teacher Details table has seven entities where ID is a primary key, cell and email is unique, and used to show the details of the teacher.

Table 5.1.9: Database Structure for Time Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	start_time	time			No	None			Change Drop More
<input type="checkbox"/> 2	end_time	time			No	None			Change Drop More
<input type="checkbox"/> 3	duration	time			No	None			Change Drop More
<input type="checkbox"/> 4	Break_start	time			No	None			Change Drop More
<input type="checkbox"/> 5	Break_End	time			No	None			Change Drop More
<input type="checkbox"/> 6	slot	int(10)			No	None			Change Drop More
<input type="checkbox"/> 7	day	varchar(120)	latin1_swedish_ci		No	None			Change Drop More

Table 5.1.9 shows that timetable has seven entities where the day is unique. Used to show the duration between start and end time. It also shows the duration of break time with the slot type.

5.2 Implementation of Front-End Design

We use to implement the front-end section with HTML and CSS and Bootstrap [6]. To run in localhost server we need XAMPP server and one browser [24]. The following figure described all graphical user interfaces.

Home Page:

In this figure 5.2.1: show the diagram of The Homepage of “Semi-Automated Time Scheduling Management System for University Classes”. In this picture, we can see that, here is four button like that Documentation, View Routine, About Us and Login. Students can view their routine from the “view routine” button. Students have no required for login. Only Admin and teacher can use “login” option for their work.

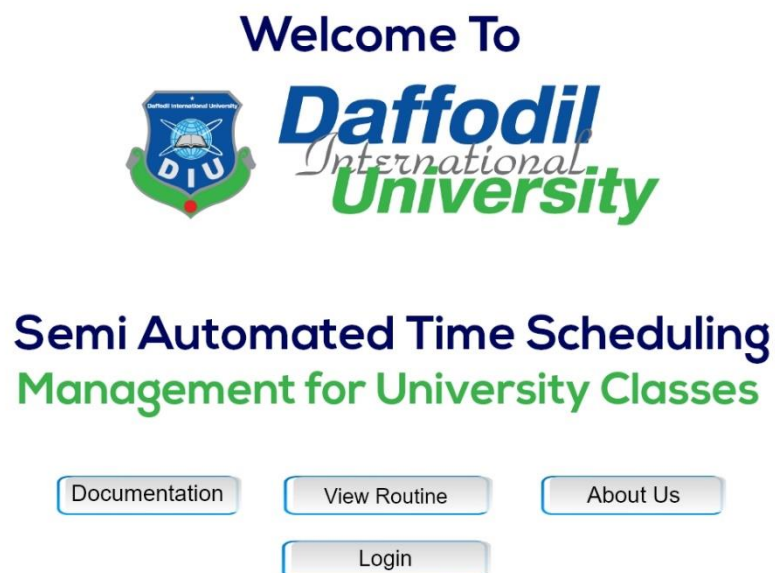


Figure 5.2.1: Home Page of the system

View Routine Page:

Figure 5.2.2 is View Routine page; in this section, all can view routine from a different viewpoint by selecting semester, level & term and so other options which are optional.

Semi Automated Time Scheduling Management for University Classes

The screenshot shows a web interface for viewing a routine. At the top, there is a button labeled "View Routine". Below it is a form with five input fields, each with a label and a placeholder text:

- Semester:** Select Semester
- Level:** Select Level & Term (Optional)
- Section:** Select Section (Optional)
- Course:** Select Course (Optional)
- Teacher:** Select Teacher's Initial (Optional)

At the bottom of the form is a blue button labeled "Submit".

Figure 5.2.2: View Routine page of the system

About Us Page:

Figure 5.2.3 is About Us page of the system that simply shows the images and short description of all supervisors and developers.

Semi Automated Time Scheduling Management for University Classes

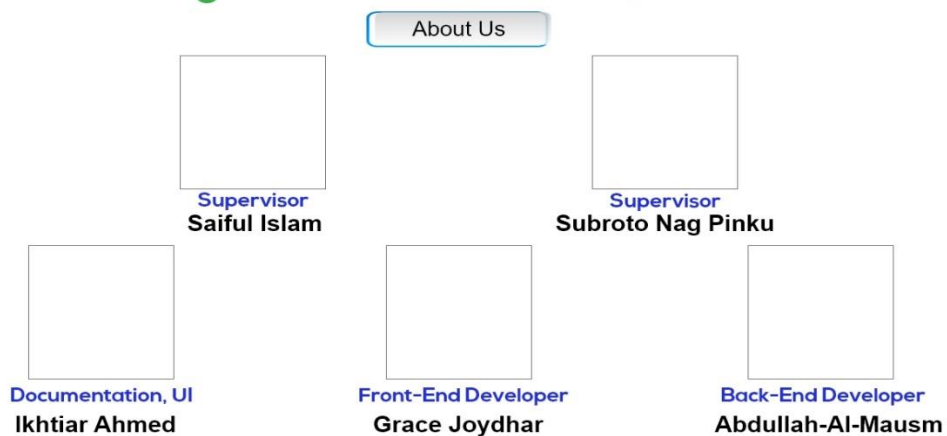


Figure 5.2.3: About Us page of the system

Login Page:

Figure 5.2.4 is a single Login Page for both admin and teacher only can access by giving valid user ID and Password.



Figure 5.2.4: Login page of the system

Admin Panel:

Figure 5.2.5: Admin home page shows the details of the admin with the image. There is the option of editing admin profile and also the option of change password.

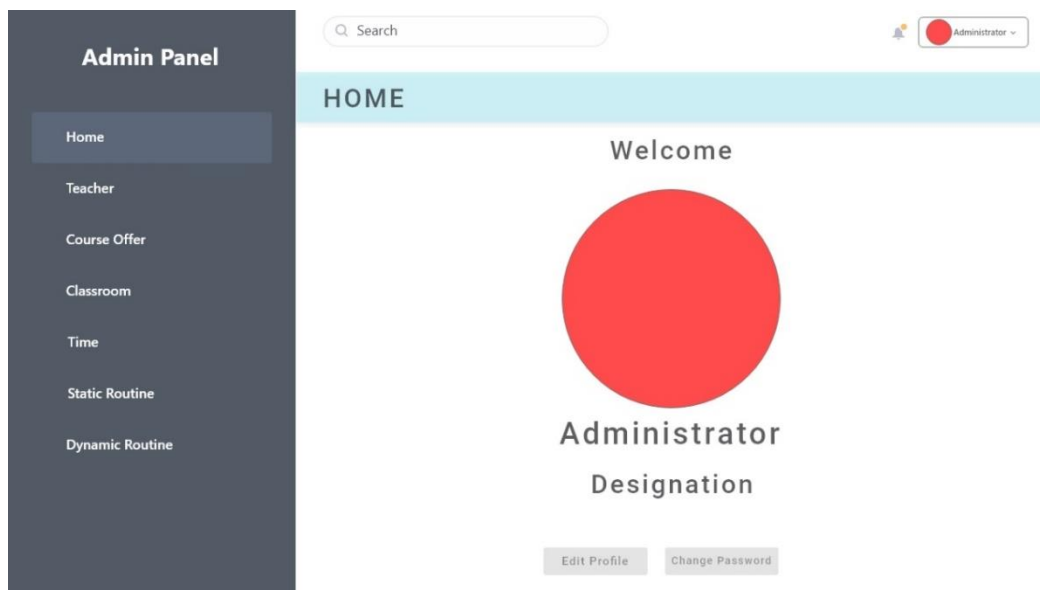


Figure 5.2.5: Admin home page.

Figure 5.2.6: Teacher option page gives the facility to import or add new teachers. Also has an option to modify the details of existing teacher.

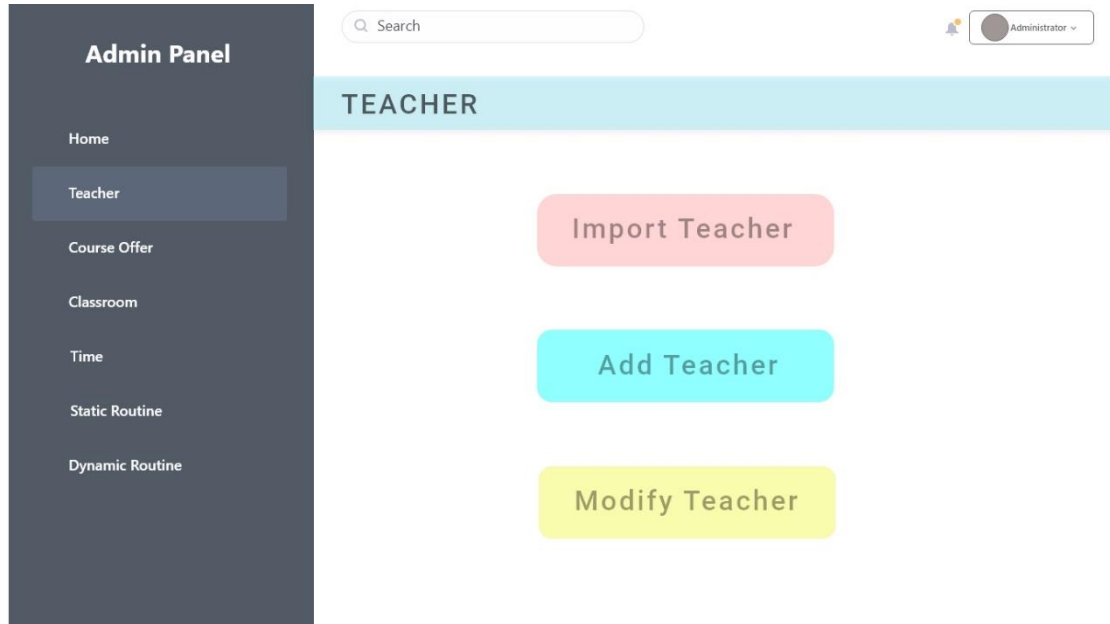


Figure 5.2.6: Teacher option page.

Figure 5.2.7: Import Teacher option page use to browse any csv or xls file for uploading. But if the admin is not interested to upload the selected file, there is an option to clear the file. Otherwise, the file will be uploaded by import file.

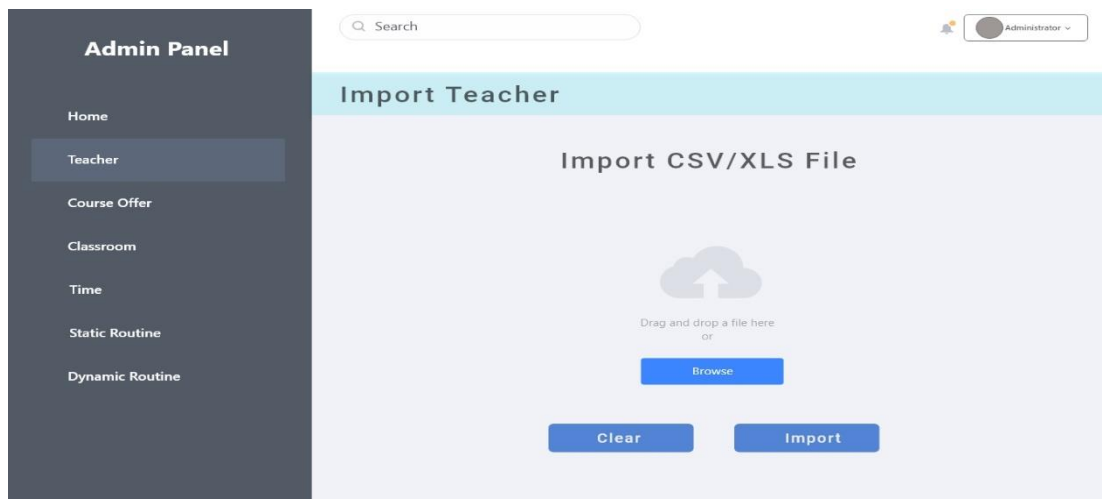


Figure 5.2.7: Import Teacher option page.

Figure 5.2.8: Add Teacher option page send a mail address to the teacher. Or there is an option to add the teacher manually by inputting the mail address and name of the teacher. After that, the details will be submitted.

Figure 5.2.8: Add Teacher option page.

Figure 5.2.9: Modify Teacher option page use to change any details about the teacher. Like for changing teacher initial, ID, name, email address or other information, the modify option will be used.

TI	ID	Name	Email	More	Modify
SI	700001	Saiful Islam	Saiful.cse@diu.edu.bd	More	Edit Delete
SNP	700002	Subroto Nag Pinku	pinku.cse@diu.edu.bd	More	Edit Delete
DMR	700003	Dewan Mamun Reza	mamun.cse@diu.edu.bd	More	Edit Delete
FF	700004	Fahad Faisal	fahad.cse@diu.edu.bd	More	Edit Delete
MJA	700005	Mohammad Jahangir Alam	jahangir.cse@diu.edu.bd	More	Edit Delete

Figure 5.2.9: Modify Teacher option page

Figure 5.2.10: Course Offer option page use to import, add or modify the existing course in the database.

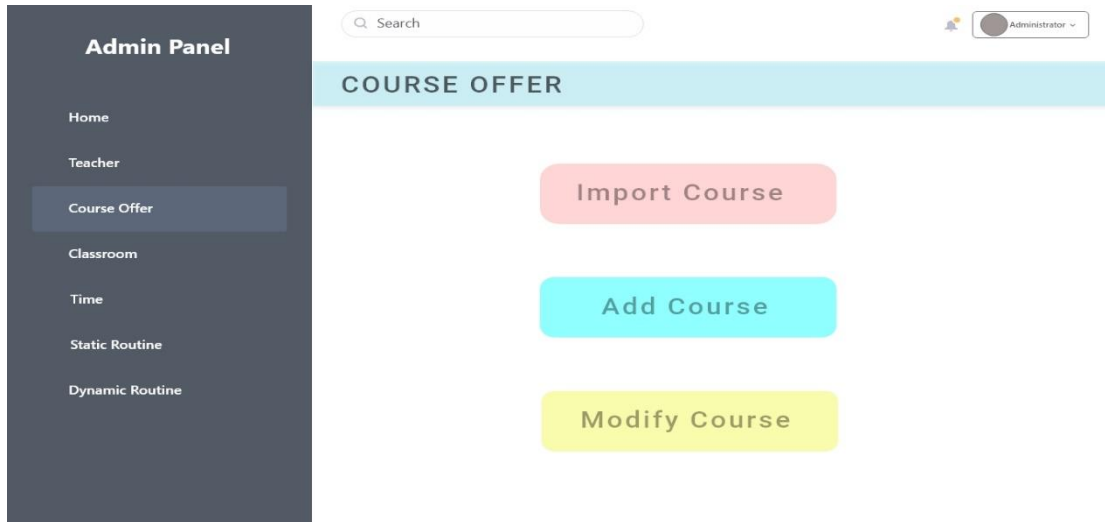


Figure 5.2.10: Course Offer option page.

Figure 5.2.11: Import Course option page use to browse any csv or xls file for uploading. But if the admin is not interested to upload the selected file, there is an option to clear the file. Otherwise, the file will be uploaded by import file.

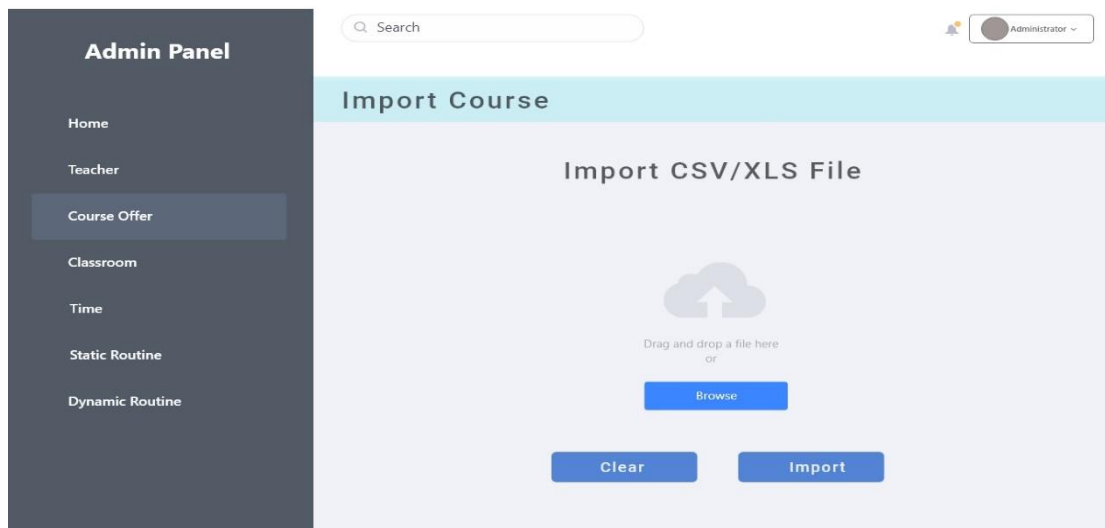


Figure 5.2.11: Import Course option page.

Figure 5.2.12: Add Course option page use to add the course and section both manually. For course, submit the course name with the course code and course credit. And for the section, section name with the code have to submit.

The screenshot shows the 'Admin Panel' on the left with a sidebar menu containing: Home, Teacher, **Course Offer**, Classroom, Time, Static Routine, and Dynamic Routine. The main content area has a search bar and a user profile for 'Administrator'. Below this is the 'Add Course' section, which includes a sub-section 'Add Course Manually' with three input fields: 'Code: Input Course Code', 'Credit: Input Course Credit', and 'Course Name: Input Course Name'. There are 'Clear' and 'Submit' buttons below these fields. The 'Add Section' section below it has two input fields: 'Code: Select Course Code' (a dropdown) and 'Section: Input Section', with 'Clear' and 'Submit' buttons.

Figure 5.2.12: Add Course option page.

Figure 5.2.13: Modify Course and Section option page use to change any information about the section or course. For example, if admin or authority wants to change the section name or modify the course name then it can be done by modifying option.

The screenshot shows the 'Admin Panel' on the left with a sidebar menu containing: Home, Teacher, **Course Offer**, Classroom, Time, Static Routine, and Dynamic Routine. The main content area has a search bar and a user profile for 'Administrator'. Below this is the 'Modify Section' section, which contains a table with the following data:

Level & Term	Code	Course Name	Credit	Section	Modify
L1 T1	CSE 112	Computer Fundamentals	3	15 View	Edit Delete
L1 T1	MAT 111	Mathematics 1	3	13 View	Edit Delete
L1 T1	ENG 113	Basic Functional English	3	13 View	Edit Delete
L1 T1	PHY113	Physics 1	3	15 View	Edit Delete
L1 T2	CSE 122	Programming & Problem Solving	3	17 View	Edit Delete
L1 T2	CSE 123	Programming & Problem Solving Lab	1	17 View	Edit Delete
L1 T2	PHY 123	Physics II	3	18 View	Edit Delete
L1 T2	PHY 124	Physics II Lab	1	19 View	Edit Delete
L1 T2	ENG 123	English II	3	19 View	Edit Delete
L1 T2	MAT 121	Mathematics-II	3	17 View	Edit Delete

Figure 5.2.13: Modify Course and Section option page.

Figure 5.2.14: Classroom option page use to import, add or modify the existing classroom details in the database.

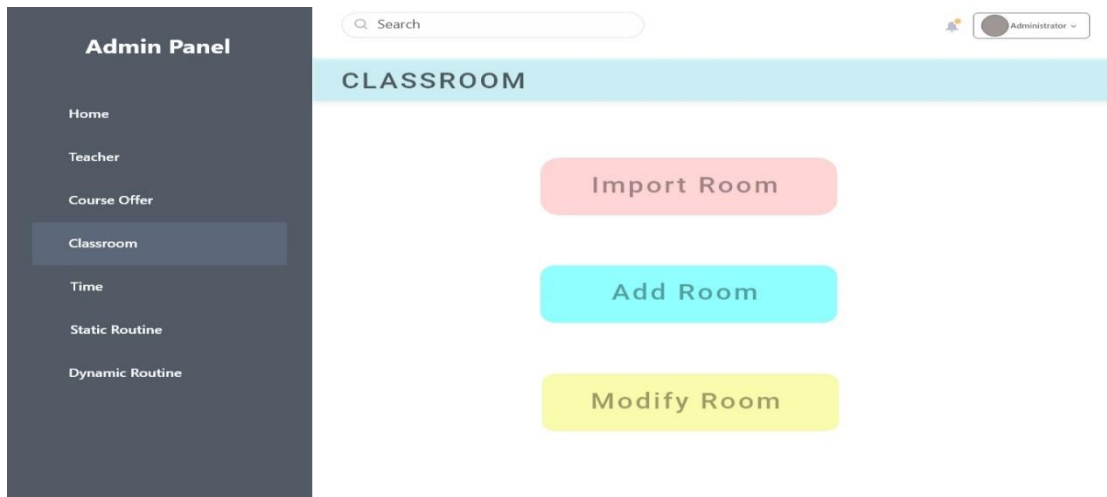


Figure 5.2.14: Classroom option page.

Figure 5.2.15: Import Room option page use to browse any csv or xls file for uploading. But if the admin is not interested to upload the selected file, there is an option to clear the file. Otherwise, the file will be uploaded by import file.

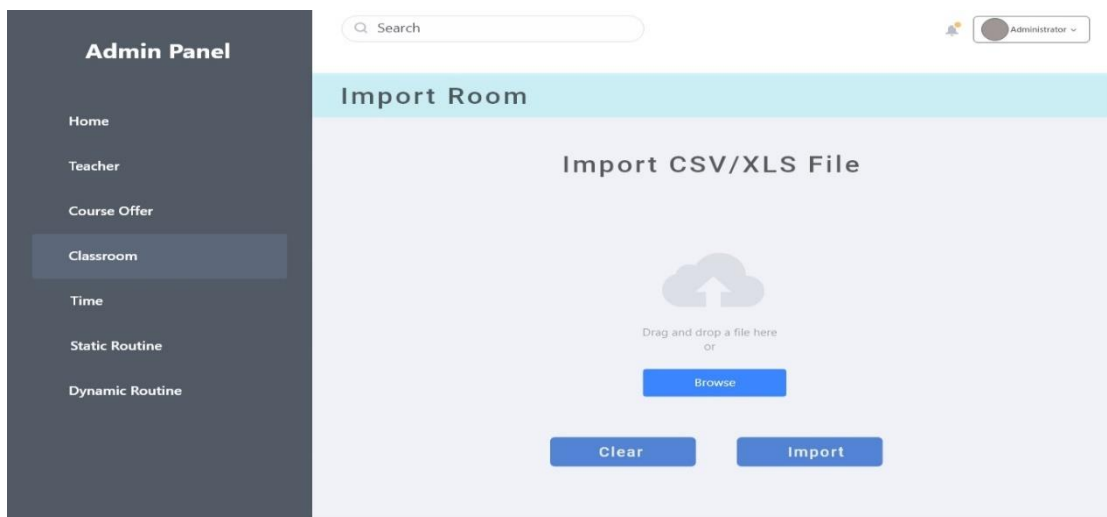


Figure 5.2.15: Import Room option page.

Figure 5.2.16: Add Room option page, room details can be added by selecting the types of room (lab or theory), room no and also the name of the building where the class will happen.

Figure 5.2.16: Add Room option page.

Figure 5.2.17: Modify Room option page, room no or building/tower can be changed. So in that case, the admin will modify the database by giving updated information.

Room Type	Room NO.	Tower Name	Modify
T	101	DTS	Edit Delete
T	102	DTS	Edit Delete
T	103	DTS	Edit Delete
T	104	DTS	Edit Delete
T	105	DTS	Edit Delete
T	106	DTS	Edit Delete
T	107	DTS	Edit Delete
L	201	CSEB	Edit Delete
L	202	CSEB	Edit Delete
L	601	CSEB	Edit Delete

Figure 5.2.17: Modify Room option page.

Figure 5.2.18: Time option page used for adding class time with the start and end time with the break time. Also for variable reasons, if the time changed then it needs to be modified.

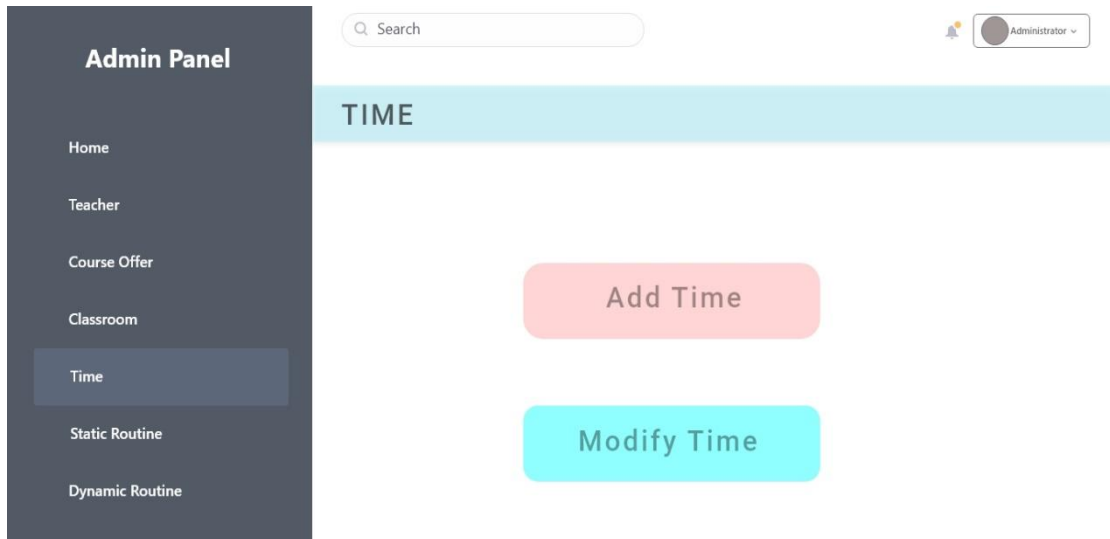


Figure 5.2.18: Time option page.

Figure 5.2.18: Add Time option page use to add time details with start and end time with the duration, also need to input start and break time.

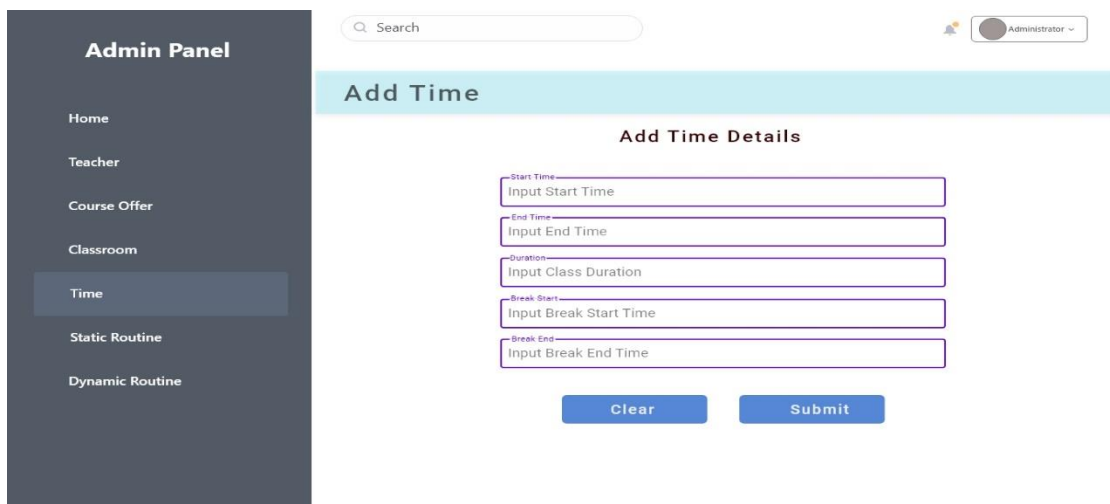


Figure 5.2.18: Add Time option page.

Figure 5.2.18: Modify Time option page used to modify time details and slot will be generated here. In the modify time, there is an option, delete. If we don't need any time, we can delete that time.

Admin Panel

- Home
- Teacher
- Course Offer
- Classroom
- Time**
- Static Routine
- Dynamic Routine

Search Administrator

Modify Time

Time Details	Value	Modify
Start Time	08:30	Edit Delete
End Time	17:30	Edit Delete
Duration	01:30	Edit Delete
Break Start Time	00:00	Edit Delete
Break End Time	00:00	Edit Delete

Slot Generated

08:30-10:00	10:00-11:30	11:30-13:00
13:00-14:30	14:30-16:00	16:00-17:30

Figure 5.2.18: Modify Time option page.

Figure 5.2.19: Static Routine option page, use to add some routine details manually at the first time where the start and end time, slot, teacher details, section, and other details will be included into the routine.

Admin Panel

- Home
- Teacher
- Course Offer
- Classroom
- Time
- Static Routine**
- Dynamic Routine

Search Administrator

STATIC ROUTINE

SATURDAY						
ROOM	08:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	16:00-17:30
L 901 CSEB	C: S: T: Add Modify	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 902 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 801 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 802 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 701 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 702 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:

Figure 5.2.19: Static Routine option page.

Figure 5.2.20: Dynamic Routine option page can generate a new routine with full details and also can use to view the routine fully.

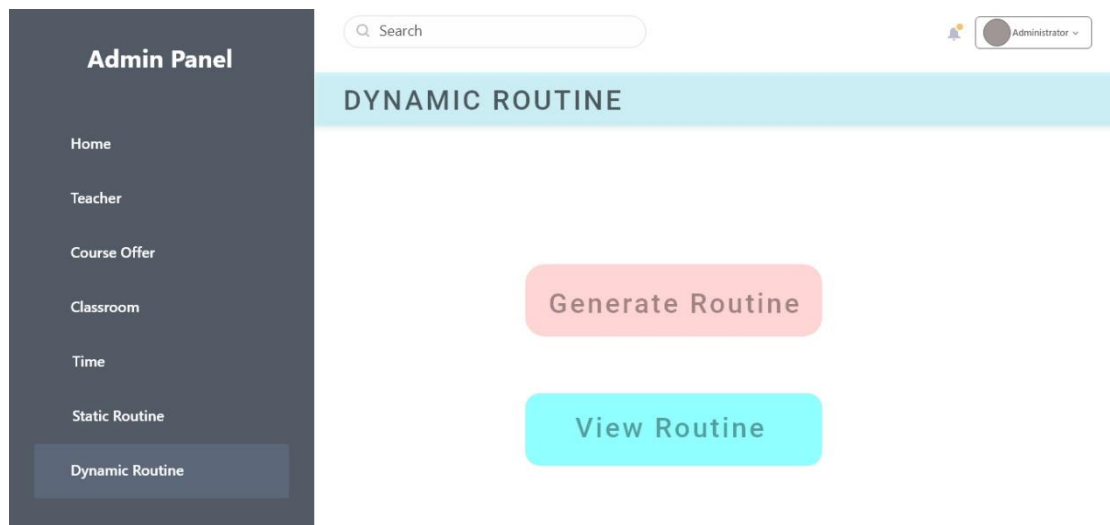


Figure 5.2.20: Dynamic Routine option page.

Figure 5.2.21: Generate Routine option page mainly shows the final result after generating the full routine. If the routine information is true then it will show 'Generation Successful'. Otherwise, some errors will be shown.

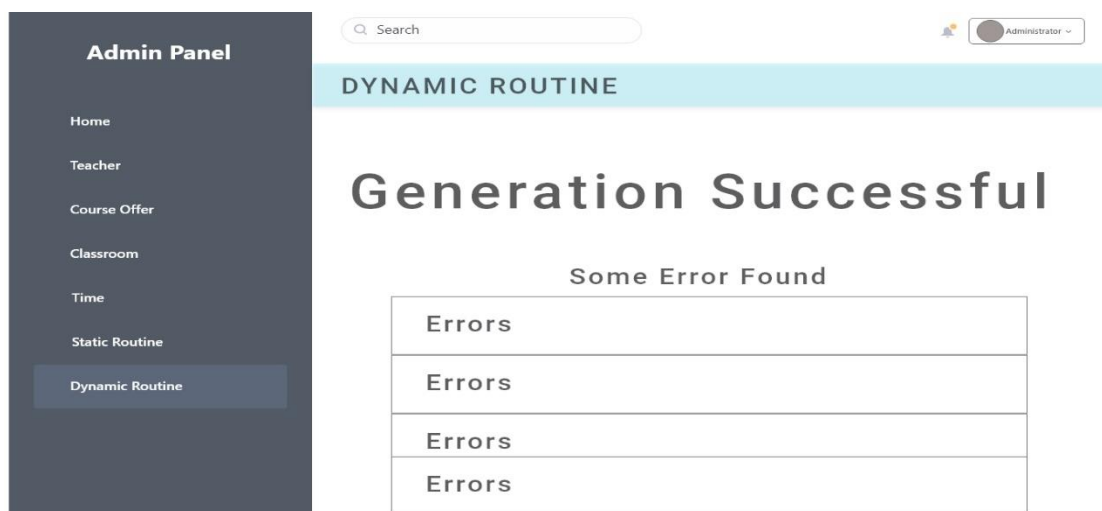


Figure 5.2.21: Generate Routine option page.

Figure 5.2.22: View Routine option page use to view the generated routine with all details of the routine, like room no, start and end time and so on.

SATURDAY						
ROOM	08:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	16:00-17:30
L 901 CSEB	C: S: T: Add Modify	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 902 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 801 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 802 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 701 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:
L 702 CSEB	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:	C: S: T:

Figure 5.2.22: View Routine option page

Teacher Panel:

Figure 5.2.23: Teacher Homepage shows the details of the teacher with the image. There is the option of editing teacher profile and also the option of change password.

Figure 5.2.23: Teacher Homepage

Figure 5.2.24: Room Booking Option page used for booking a room. The teacher can see the available room, time and also the day details and then they can book the room.

The screenshot shows the 'Room Booking' page. On the left is a 'Teacher's Panel' sidebar with options: Home, Room Booking (selected), My Routine, and Change Request. The main content area has a search bar and a user profile dropdown labeled 'Teacher'. Below this is a light blue header 'ROOM BOOKING' and a sub-header 'Available Rooms'. A table lists available rooms:

DAY	TIME	ROOM	BOOKING
Saturday	13:00-14:30	105 DT	Book Now
Monday	16:00-17:30	105 DT	Book Now
Thursday	10:00-11:30	901 CSEB	Book Now
Sunday	08:30-10:00	802 CSEB	Book Now

Figure 5.2.24: Room Booking Option page

Figure 5.2.25: Teacher's Routine page use to show the class details for a teacher. It will show the off day or the everyday class time. But it will possible after login to the profile with a specific id and password.

The screenshot shows the 'My Routine' page. On the left is a 'Teacher's Panel' sidebar with options: Home, Room Booking, My Routine (selected), and Change Request. The main content area has a search bar and a user profile dropdown labeled 'Teacher'. Below this is a light blue header 'MY ROUTINE'. A table shows the routine for each day:

	08:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	16:00-17:30
Saturday						
Sunday						
Monday						
Tuesday						
Wednesday						
Thursday	OFF DAY	OFF DAY	OFF DAY	OFF DAY	OFF DAY	OFF DAY

Figure 5.2.25: Teacher's Routine page

Figure 5.2.26: Teacher's Feedback option page use to send any feedback or request about changing class time, slot or day. Teacher needs to write the details about the request.

The screenshot shows a web interface for a teacher's panel. On the left is a dark grey sidebar with the title 'Teacher's Panel' and four menu items: 'Home', 'Room Booking', 'My Routine', and 'Change Request'. The 'Change Request' item is highlighted. At the top of the main content area is a search bar and a user profile dropdown labeled 'Teacher'. Below this is a light blue header with the text 'CHANGE REQUEST'. The main form area contains a 'Select' dropdown menu with the text 'Select Request For', a 'Details' section with the text 'Details About your Request' and a large empty text area for input, and a blue 'Send' button at the bottom right.

Figure 5.2.26: Teacher's Feedback option page

5.3 Testing Implementation

A systems quality assurance is a matter of concern. It is very important to assure that the product quality is meeting its goal. The only way to assure the system stability and to assure that the quality is meeting the goal is to test the system more than once. By using different testing methods, a tester can identify the errors and problem. In addition, it helps the developers to solve the problems with further development. Testing is also important to make the developers know is there any lack of something to implement that is necessary to satisfy the users of a system.

A systems quality assurance is a matter of concern. It is very important to assure that the product quality is meeting its goal. The only way to assure the system stability and to assure that the quality is meeting the goal is to test the system more than once. By using different testing methods, a tester can identify the errors and problem. In

addition, it helps the developers to solve the problems with further development. Testing is also important to make the developers know is there any lack of something to implement that is necessary to satisfy the users of a system.

There are several types of testing that we have already maintained. They are-

- Unit Testing
- Integration Testing
- System Testing
- Validation Testing
- Acceptance Testing
- Regression Testing

Unit Testing: Unit testing represents them, isolate each part of the process [30]. And also test every unit of the process. Normally unit testing performed by developers before the setup is transferred over to the testing team to formally run the test case. We also test each part of the process, like that Import Process, Database posting, Single Addition, and Modification.

List of unit test to the system-

- Input parameter
- Variable data types
- Data overflowing
- Data receive a response from the server
- Data hash mapping
- Filtering the desired result is working or not
- Database connection
- Session starting and ending
- Exception Handling

Integration Testing: Integration testing is one kind of testing which linked parts of the system to visualize if they function correctly [30]. Integration testing can be done in two different ways-

- Bottom-up Integration
- Top-down Integration

Bottom-up Integration: The Bottom-up Integration starts with unit testing, trailed by the trial of logically more elevated amount mixes of units called modules or assemblies.

Top-Down Integration: In this testing, firstly tested of highest-level modules and after that tested lower-level modules.

System Testing: This type of testing tests the whole system. When all components are integrated, the application as the entire process is tested letter by letter to specify the quality [30]. This type of testing accomplished by the specialized testing team. System testing is important because it performs several ways. The first step is development life cycle, secondly, it verifies the functionality and thirdly verify and validate the requirements as well as the processor architecture.

Validation Testing: Validation Testing is also another important testing of the test system. Because it measured the durability of the system. We are checking this system with various input in a different field and incorrect use id, password to login to the system.

Acceptance Testing: Probably this is the most significant type of testing in the test system. The testing directed by the quality assurance team. It also finds out the actual accuracy of the system. Acceptance testing finds out the spelling mistakes, interface gaps but also search any bugs in the system.

Regression Testing: Regression testing used to verify the fixed bug that hasn't resulted in another requirement. When the testing finds any fault of the system it does not publish the result it's being uncovered in the process. Regression testing also reduces the gaps when testing, verify the new changes, moderate risk when testing is performed and also increase the speed to market the product.

5.4 Test Results and Report

There are different types of error that we had to face while measuring the performance of the system [30]. We got various errors like that library error, input error, mailing error, message error, data type error, syntax error. We were tried to solve the plenty of error but we were not able to solve all of this error.

List of major errors are-

- Not declaring the variable type as they meant to be
- Lower-case and Upper-case values
- Data Parsing
- Lots of Syntax error
- Exception handling

We tries to solve this entire problem to get this well-made system and to ensure the best of it to the users.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

We have developed a “Semi-Automated Time Scheduling Management System for University Classes” successfully. It is so difficult to manage full routine manually. As a result, occurring many problems like that overlap section, overlap teacher, multiple class in same time [17]. For this reason, we took a decision to manage routine dynamically. After developing the system, teacher and student easily cooperate with their routine. Super admin can add a normal admin. Entering the system teacher and admin have to log in. If Admin changes anything in the system, its fall effects of the full system.

To generate the dynamic routine some information must be entered in this process. This process completing by an administrator. Like that teacher information, section, classroom, tower, time etc. Admin can also set the rules for a dynamic routine like that how many courses a teacher able to take. Another restriction is back to back class is not allow for the teacher, the minimum time gap is required between two classes [17]. At last, Database Resource Management System helps to generate the routine dynamically.

This automated system is helpful for the teacher, student and university management. Because this system reduces their time, workload, teacher, and student found their routine easily.

We hope that our automated system will use in every educational institution to solve their problem.

6.2 Scope for Further Development

We think that Our Developed system is for use for the educational institution because it is future scope.

There are many scopes of this system in the near future. Some are –

- MySQL will be utilized in the back-end database.
- We would like to implementation for the entire department.
- To enhance the ease of use and capability.
- Increase more functionality for the user's requirements.
- To build up the examination routine for the entire department.

APPENDICES

Appendix A: Project Reflection

N-P Heard Problem: We show that the problem of finding an optimal schedule for a set of jobs is *NP*-complete even in the following two restricted cases. All jobs require one time unit. All jobs require one or two time units, and there are only two-processor resolving. Therefore, the General preemptive scheduling problem is also *NP*-complete.

HTML - Hyper Text Markup Language, HTML is the standard markup language for creating Web pages.

CSS - Cascading Style Sheets. CSS is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

Laravel - Laravel is a free, open-source PHP web framework, created by Taylor Orwell and intended for the development of web applications following the model–view–controller architectural pattern and based on Symphony.

PHP - Hypertext Preprocessor is a server-side scripting language designed for Web development, and used as a general-purpose programming language.

Java script - JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language that is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

URL - A Uniform Resource Locator, colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.

MySQL - MySQL is an open source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

UI & UX - User interface design or user interface engineering is the design of user and UX design refers to **user experience** design, while UI design stands for user interface design. Both of these are crucial to an IT product and need to work closely together.

Bootstrap - Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation, and other interface components, as well as optional JavaScript extensions.

Project Link:

https://github.com/masumjim/F_Project

Appendix B: Related Diagrams

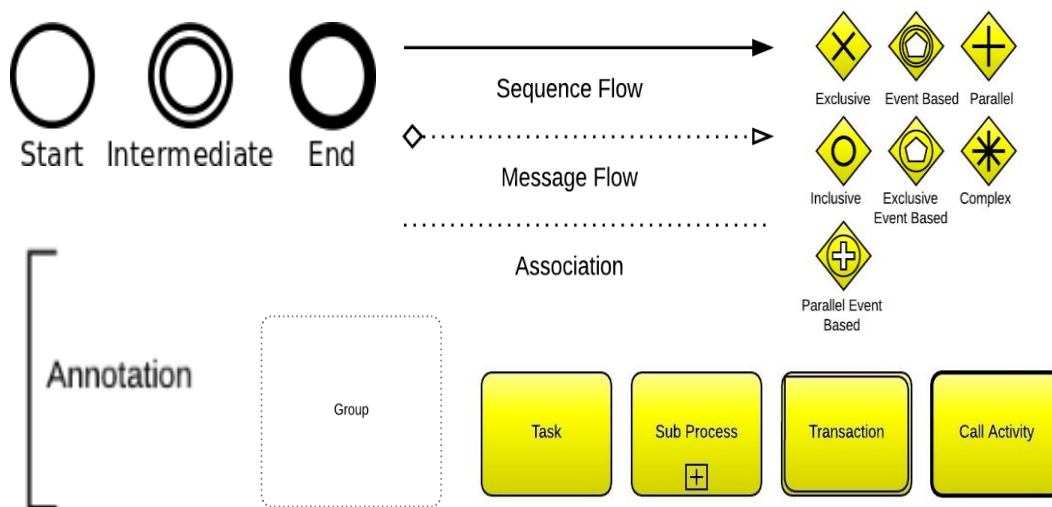


Figure B.1: BPM Notations

REFERENCES

- [1] Business_Process_Model_and_Notation.[online].Available at <<https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation/>> Last accessed on 20.11.2018 at 10:20 PM
- [2] Php-tutorial.[online].Available at << <https://www.codingunit.com/php-tutorial-language-introduction>>> Last accessed on 20.11.2018 at 12:16 PM
- [3] front-end-developer.[online].Available at << <https://skillcrush.com/2016/02/11/skills-to-become-a-front-end-developer/>>> Last accessed on 20.11.2018 at 12:30 PM
- [4] Class routine management system.[online].Available at <<<http://www.assignmentpoint.com/business/management/routine-management-system-for-department-of-computer-science-and-engineering.html>>>Last accessed on 20.11.2018 at 12:35 PM
- [5] Routine management system.[online]. Available at <<http://soft-project.blogspot.com/2009/01/routine-management-system.html>>> Last accessed on 20.11.2018 at 1:00 PM
- [6] Requirements specification.[online].Available at <<<https://www.scribd.com/doc/48111565/Software-Requirements-Specification-for-online-student-management-system>>> Last accessed on 20.11.2018 at 12:16 PM
- [7] Software Requirements specification.[online].Available at: <<http://www.academia.edu/24648944/software_requirements_specification_srs_for_student_information_management_system>> Last accessed on 20.11.2018 at 12:16 PM
- [8] Types of Data Flow diagram.[online].Available at <<https://www.lucidchart.com/pages/data-flow-diagram/data-flow-diagram-symbols>>> Last accessed on 20.11.2018 at 12:16 PM
- [9] Data flow diagram.[online].Available at: <https://en.wikipedia.org/wiki/Data_flow_diagram>> Last accessed on 20.11.2018 at 12:10 PM
- [10] *D.B Stewart and p.k khosla "Real time scheduling of sensor based control systems" vol-24. Department of Electrical and Computer Engineering and the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, PP-27-29*
- [11] *Saiful islam and Thanjida akhter. "design and depeloment of integrated class routine information and management system":Daffodil International University /Library, P03883,pp-1-50.*
- [12] *Md. Farukul Islam "Smart Attendance System": Daffodil International University /Library*
- [13] *Sifat E Jahan, Ruksana Akter Brishte and Sadia Mehrose Islam "Intelligent Routine Management",School of Engineering and Computer Science BRAC University.*

- [14] Anil Lama "ROUTINE MANAGEMENT SYSTEM": DWIT College, DEERWALK INSTITUTE OF TECHNOLOGY, pp-27-29.
- [15] IM020-09 "An Automated System for Patient Record Management" vol- 22: Faculty of computing and information technology, Makerere University, June 2009
- [16] OGBIBE NKECHI AGNES "AUTOMATED HOSPITAL MANAGEMENT SYSTEM" vol-18 : Department of Electronic Engineering, University of Nigeria, Nsukka, Nigeria, , pp-5-7, May 2011
- [17] Mohammad Majbah UddinMd. Nurul Huda ANDMd. Ekramul Hoqu "Dynamic Routine Management System"vol-3:daffodil international university/library, p06718,January pp:1-50,2016,
- [18] Mugdha Kishor Patil , Rakhe Shruti Subodh , Prachi Ashok Pawar and Naveena Narendrasingh Turkar "Web Application for Automatic Time Table Generation" vol-4 International Journal of Current Engineering and Technology,June 2014
- [19] Interaction Design.[online].Available at <<<https://medium.com/strategic-designs/5-pillars-of-interaction-design-ixd-every-designer-must-know-about-a-very-big-ux-factor-5350e1706b0>>> Last accessed on 20.11.2018 at 1:50PM
- [20] Use Case Diagram,[online]. Available at << <https://www.lucidchart.com/pages/uml-use-case-diagram>>> Last accessed on 21.11.2018 at 1:50PM
- [21] Use Case Description.[online]. Available at << Available at: <https://www.dummies.com/business/business-strategy/how-to-create-use-case-description-for-your-business-analysis-report/>>> Last accessed on 21.11.2018 at 12:16 PM
- [22] ER-diagram. Available at << <https://www.smartdraw.com/entity-relationship-diagram/>>> Last accessed on 21.11.2018 at 12:16 PM
- [23] Logical data model.[online]. Available at << https://en.wikipedia.org/wiki/Logical_data_model>> Last accessed on 21.11.2018 at 10:11 PM
- [24] XAMPP.[online]Available at << <https://www.apachefriends.org/index.html>>> Last accessed on 17.11.2018 at 11:16 PM
- [25] windows.[online].Available at << <https://www.microsoft.com/en-us/windows>>> Last accessed on 22.11.2018 at 10:40 PM
- [26]Ubuntu.[online].Available at << <https://www.ubuntu.com/#download>>>Last accessed on 22.11.2018 at 4:20PM
- [27] web browser-Mozilla Firefox.[online].Available at << <https://www.mozilla.org/en-US/firefox/new/>>> Last accessed on 22.11.2018 at 12:16 PM
- [28] Bootstrap.[online]. Available at << <https://getbootstrap.com/>>> Last accessed on 23.11.2018 at 5:20PM
- [29] Laravel.[online]. Available at <<<https://laravel.com/docs/5.7/installation>>> Last accessed on 10.11.2018 at 2:20 PM

[30] software testing overview. Available at

<<https://www.tutorialspoint.com/software_testing/software_testing_overview.htm>> Last accessed on 10.11.2018 at 2:30 PM

[31] Project and Intern. Available at

<<<https://sites.google.com/daffodilvarsity.edu.bd/picsediu/downloads>>> Last accessed on 23.11.2018 at 2:30 PM

Document Viewer

Turnitin Originality Report

Processed on: 24-Nov-2018 18:31 +06

ID: 1044025353

Word Count: 10710

Submitted: 1

151-15-4952 By Abdullah-al-Masum

Similarity Index 6%	Similarity by Source Internet Sources: 6% Publications: 0% Student Papers: 7%
--	---

[include quoted](#)
[include bibliography](#)
[excluding matches < 1%](#)
[download](#)
[refresh](#)
[print](#)
 mode:

2% match (student papers from 05-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [941584057](#)

1% match (Internet from 21-Sep-2018) ✕

<http://dspace.library.daffodilvarsity.edu.bd:8080>

1% match (student papers from 02-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [939620869](#)

1% match (Internet from 27-Jun-2017) ✕

<https://www.lucidchart.com/pages/data-flow-diagram/data-flow-diagram-symbols>

1% match (student papers from 17-Nov-2015) ✕

[Submitted to InSite Institution Placeholder on 2015-11-17](#)

1% match (student papers from 04-Apr-2018) ✕

Class: Article 2018

Assignment: Journal Article

Paper ID: [940919861](#)