



Daffodil
International
University

Timetable Displayer

Smart Class routine maker

Submitted by

Kanak Kumar Sarker

ID: 171-35-2043

Department of Software Engineering

Daffodil International University

Supervised by

Ms. Fatama Binta Rafiq

Lecturer

Department of Software Engineering

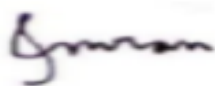
Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of
Bachelor of Science in Software Engineering.

APPRPVAL

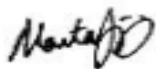
This project titled on “Timetable Displayer”, submitted by Kanak Kumar Sarker (ID: 171-35-2043) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



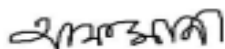
Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



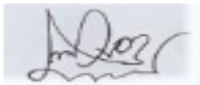
Internal Examiner 1

Dr. Md. Mostafijur Rahman
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



Internal Examiner 2

Afsana Begum
Lecturer (Senior)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

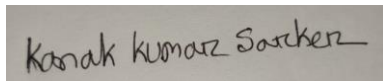


External Examiner

Professor Dr. Mohammed Nasir Uddin
Professor and Chairman
Department of Computer Science and Engineering
Jagannath University

DECLARATION

I hereby declare that project titled “Timetable Displayer” has been completed by me under the supervision of Ms. Fatama Binta Rafiq, Lecturer, Department of Software Engineering, Daffodil International University for the purpose of achieving degree of Bachelor of Science from Daffodil International University. This is also declared by me that neither this project nor any part of this project has been used or submitted elsewhere for any kind of degree or awards.

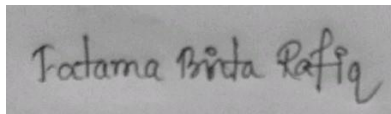


.....
Kanak Kumar Sarker

ID: 171-35-2043

Department of Software Engineering

Daffodil International University



.....
Ms. Fatama Binta Rafiq

Lecturer

Department of Software Engineering

Daffodil International University

Supervisor

ACKNOWLEDGMENT

I thank our creator Almighty God from deep of my heart and show much gratitude towards him for his heavenly blessing as we are able to submit our final year project successfully.

I also show my heartiest gratitude and accept our utmost liability to our supervisor **Ms. Fatama Binta Rafiq, Lecturer**, Department of Software Engineering, Daffodil International University, Dhaka. My supervisor's vast knowledge in the area of web application and her enthusiastic approach toward my project has deeply encouraged us to complete this project. Her strength, her forbearance, her scholastic counsel, her unparalleled leadership, her passion, her profound reflection on each rough sheet, and her mistakes and helpfully correcting them every time has formed my attempts into a complete project.

I want to share my appreciation and gratitude to **Dr. Imran Mahmud, Associate Professor and Head In-Charge**, Department of Software Engineering, for his gracious look towards my project which has made it possible to complete it properly. I also thank other faculty members and staff of the Software Engineering department of Daffodil International University for their kindness and helpful attitude towards us.

I would like to thank all of my classmates at Daffodil International University, who took part in this discussion while completing the course work.

At last, I must acknowledge with due respect the constant support and patience of my parents.

ABSTRACT

Timetable Displayer is a web based application. The system is used to build and manipulate an educational institution's class routine. A timetable for an educational institute is a temporary arrangement of a series of lectures and classrooms in which all the restrictions specified are met.

Through automating this process with a computer-assisted timetable display, admin who is interested in designing and maintaining course timetables will save a lot of valuable time. Because each institution has its own timetabling problem, the software packages that are commercially available cannot meet any educational institution's needs.

I have therefore created a realistic approach to developing a timetabling system for lecture courses, which can be modified to suit any timetabling problem for educational institutions. This project implements a functional timetabling algorithm, used in an integrated timetabling method, able to easily take care of both strong and weak constraints. Until they are finalized for a given term, each student can access their timetable, but they cannot update it.

Table of Contents

APPRPVAL	i
BOARD OF EXAMINERS	i
DECLARATION	ii
ACKNOWLEDGMENT	iii
ABSTRACT	iv
LIST OF TABLES	x
LIST OF FIGURES	x
CHAPTER 1	1
INTRODUCTION	1
1.1 Project Overview	1
1.2 Project Purpose	1
1.2.1 Background	1
1.2.2 Benefits & Beneficiaries	2
1.2.3 Goals	2
1.3 Stakeholders	2
1.4 Project Schedule	2
1.4.1 Gantt Chart	3
1.4.2 Project Timeline	4
CHAPTER 2	5
SOFTWARE REQUIREMENT SPECIFICATION	5
2.1 Function Requirement: Admin	5
2.1.1 Log in	5
2.1.2 Periods manage	5
2.1.3 Room manage	5
2.1.4 Course manage	6
2.1.5 Course Teachers manage	6
2.1.6 Classes manage	6
2.1.7 Create Timetable	6
2.2 Function Requirement: Student	7
2.2.1 Registration	7
2.2.2 Login	7
2.2.3 View Timetable	7

2.2.4 Download PDF	7
2.3 Performance Requirements	8
2.3.1 Capacity Requirements	8
2.4 Dependability Requirements	8
2.4.1 Reliability Requirements	8
2.4.2 Fault-Tolerance Requirements	8
2.4.3 Safety-Critical Requirements	9
2.5 Maintainability and Supportability Requirements	9
2.5.1 Maintenance Requirements	9
2.5.2 Supportability Requirements	9
2.5.3 Adaptability Requirements	9
2.6 Security Requirements	9
2.6.1 Access Requirements	9
2.6.2 Integrity Requirements	10
2.6.3 Privacy Requirements	10
2.7 Usability and Human-Interaction Requirements	10
2.7.1. Ease of Use Requirements	10
2.7.2 Understandability Requirements	10
2.7.3 Accessibility Requirements	10
2.7.4 User Documentation Requirements	10
2.7.5 Training Requirements	10
CHAPTER 3	11
SYSTEM ANALYSIS & DESIGN	11
3.1 Use Case Diagram	11
3.2 Use Case Description	12
3.2.1 Admin Login	12
3.2.2 Admin Manage Periods	13
3.2.3 Admin Manage rooms	14
3.2.4 Admin Manage courses	15
3.2.5 Admin Manage course Teachers	16
3.2.6 Admin Manage Classes	17
3.2.7 Admin Create New Timetable	18

3.2.8 Student Registration	19
3.2.9 Student Login	20
3.2.10 View Timetable	21
3.2.11 Print PDF	21
3.3 Activity Diagram	22
3.3.1 Admin Login Activity	22
3.3.2 Admin Manage Periods Activity	23
3.3.3 Admin Manage Rooms Activity	24
3.3.4 Admin Manage Courses Activity	25
3.3.5 Admin Manage Course Teacher Activity	26
3.3.6 Admin Manage Classes Activity	27
3.3.7 Admin Create Timetable Activity	28
3.3.8 Student Registration	29
3.3.9 Student Login	30
3.3.10 View Timetable Activity	31
3.3.11 Print PDF activity	32
3.4 System Sequence Diagram	33
3.4.1 Admin Login	33
3.4.2 Admin Manage Periods	34
3.4.3 Admin Manage Rooms	35
3.4.4 Admin Manage Courses	36
3.4.5 Admin Manage Course Teachers	37
3.4.6 Admin Manage Classes	38
3.4.7 Admin Create Timetable	39
3.4.8 Student Registration	40
3.4.9 Student Login	41
3.4.10 View Timetable	42
3.4.11 Print PDF	43
3.5 System Entity Relation Diagram	44
3.7 Design and Implementation Constraints	45
3.7.1 Framework or Software Language:	45
3.8 Development Tools and Technology:	45

3.9 Project Organization	45
3.9.1 Software Process Model	45
CHAPTER 4	47
SYSTEM TESTING	47
4.1 Testing Features	47
4.1.1 Features to be tested	47
4.2 Testing Strategies	47
4.2.1 Test Approach	47
4.2.2 Test Category	47
4.2.3 Success/Failed Criteria	47
4.3 Testing Environment	48
4.4 Test Cases	48
Testing Case No-1 (Integration Testing)	48
Testing Case No-2 (Module Testing)	52
CHAPTER 5	54
USER MANUAL	54
5.1 Admin Login Page	54
5.2 Admin Dashboard	54
5.3 Admin Manage Periods	55
5.3.1 View Periods	55
5.3.2 Add New Timeslots	55
5.3.3 Update Timeslot	56
5.3.4 Delete Timeslots	56
5.4 Admin Manage Rooms	57
5.4.1 View Rooms	57
5.4.2 Add Room	57
5.4.3 Update Rooms	58
5.4.4 Delete Room	58
5.5 Admin Manage Courses	59
5.5.1 View Courses	59
5.5.2 Add Course	59
5.5.3 Update Course	60

5.5.4 Delete Course	60
5.6 Admin Manage Teachers	61
5.6.1 View Teachers	61
5.6.2 Add Teacher	61
5.6.3 Update Teacher	62
5.6.4 Delete Teacher	62
5.7 Admin Manage Classes	63
5.7.2 Add Class	63
5.7.3 Update Class	64
5.7.4 Delete Class	64
5.8 Admin Create Timetable	65
5.9 Student Sign Up	65
5.10 Student Login	66
5.11 Student Dashboard	66
5.12 Student Profile	67
5.13 View Timetable and Download	67
5.14 PDF Format	68
CHAPTER 6	69
PROJECT SUMMARY	69
6.1 GitHub Link	69
6.2 Limitations	69
6.3 Obstacles and Achievements	69
6.4 Future Scope	69
6.5 References	70
6.6 Plagiarism Report	71

LIST OF TABLES

CHAPTER 1

Table 1.4.2: Project Timeline	4
-------------------------------	---

CHAPTER 3

Table 3.2.1: Admin Login	12
Table 3.2.2: Admin Manage Periods	13
Table 3.2.3: Admin Manage Rooms	14
Table 3.2.4: Admin Manage Courses	15
Table 3.2.5: Admin Manage Course Teachers	16
Table 3.2.6: Admin Manage Classes	17
Table 3.2.7: Admin Manage Create New Timetable	18
Table 3.2.8: Student Registration	19
Table 3.2.9: Student Login	20
Table 3.2.10: View Timetable	21
Table 3.2.11: Print PDF	21

CHAPTER 4

Table 4.4.1: Testing Case (Admin & Student Login)	48
Table 4.4.2: Testing Case (Add Teacher)	52

LIST OF FIGURES

CHAPTER 1

Figure 1.4.1: Gantt chart	3
---------------------------	---

CHAPTER 3

Figure 3.1: Use case for Timetable Displayer	11
Figure 3.3.1: Admin Login Activity Diagram	22
Figure 3.3.2: Admin Manage Periods Activity Diagram	23
Figure 3.3.3: Admin Manage Rooms Activity Diagram	24
Figure 3.3.4: Admin Manage Courses Activity Diagram	25
Figure 3.3.5: Admin Manage Course Teacher Activity Diagram	26
Figure 3.3.6: Admin Manage Classes Activity Diagram	27
Figure 3.3.7: Admin Create Timetable Activity Diagram	28
Figure 3.3.8: Student Registration Activity Diagram	29

Figure 3.3.9: Student Login Activity Diagram	30
Figure 3.3.10: View Timetable Activity Diagram	31
Figure 3.3.11: Print PDF activity Diagram	32
Figure 3.4.1: Admin Login Sequence Diagram	33
Figure 3.4.2: Admin Manage Periods Sequence Diagram	34
Figure 3.4.3: Admin Manage Rooms Sequence Diagram	35
Figure 3.4.4: Admin Manage Courses Sequence Diagram	36
Figure 3.4.5: Admin Manage Course Teachers Sequence Diagram	37
Figure 3.4.6: Admin Manage Classes Sequence Diagram	38
Figure 3.4.7: Admin Create Timetable Sequence Diagram	39
Figure 3.4.8: Student Registration Sequence Diagram	40
Figure 3.4.9: Student Login Sequence Diagram	41
Figure 3.4.10: View Timetable Sequence Diagram	42
Figure 3.4.11: Print PDF Sequence Diagram	43
Figure 3.5: Entity Relation Diagram	44
Figure 3.9.1: Waterfall Model	46
CHAPTER 4	
Figure 4.4.1.1: Blank Admin Login Attempt	49
Figure 4.4.1.2: Blank Student Login Attempt	49
Figure 4.4.1.3: Admin Login Attempt with Correct Data	50
Figure 4.4.1.4: Student Login Attempt with Correct Data	50
Figure 4.4.1.5: Admin Login Successful with Authentication	51
Figure 4.4.1.6: Student Login Successful with Authentication	51
Figure 4.4.2.1: Blank filled submit	53
Figure 4.4.2.2: Teacher Data Insert Successfully	53
CHAPTER 5	
Figure 5.1 UI (Admin Login Page)	54
Figure 5.2 UI (Admin Dashboard Page)	54
Figure 5.3.1 UI (View Periods Page)	55
Figure 5.3.2 UI (Add New Timeslots Page)	55
Figure 5.3.3 UI (Update Timeslot Page)	56

Figure 5.3.4 UI (Delete Timeslots Page)	56
Figure 5.4.1 UI (View Rooms Page)	57
Figure 5.4.2 UI (Add Room Page)	57
Figure 5.4.3 UI (Update Room Page)	58
Figure 5.4.4 UI (Delete Room Page)	58
Figure 5.5.1 UI (View Courses Page)	59
Figure 5.5.2 UI (Add Course Page)	59
Figure 5.5.3 UI (Update Course Page)	60
Figure 5.5.4 UI (Delete Course Page)	60
Figure 5.6.1 UI (View Teachers Page)	61
Figure 5.6.2 UI (Add Teacher Page)	61
Figure 5.6.3 UI (Update Teacher Page)	62
Figure 5.6.4 UI (Delete Teacher Page)	62
Figure 5.7.1 UI (View Classes Page)	63
Figure 5.7.2 UI (Add Class Page)	63
Figure 5.7.3 UI (Update Class Page)	64
Figure 5.7.4 UI (Delete Class Page)	64
Figure 5.8 UI (Create Timetable Page)	65
Figure 5.9 UI (Student Sign up Page)	65
Figure 5.10 UI (Student Login Page)	66
Figure 5.11 UI (Student Dashboard Page)	66
Figure 5.12 UI (Student Profile Page)	67
Figure 5.13 UI (View Timetable and Download Page)	67
Figure 5.14 UI (PDF format Page)	68

CHAPTER 1

INTRODUCTION

1.1 Project Overview

There is a routine for each educational institution to handle its delegated classes and time. Actually, a single Branch of the company is manually designing these rituals. When designing these timetables, the conditions of teacher's time, deadline, and class time, etc. For creating timetable are often performed manually. Often, over time, habits alter. Thus, because of the shift in the importance of the parameters, their handling becomes more dynamic. It would help to work easier if the above thorough process of developing and managing a schedule can be achieved using an automated tool. In the absence of such an organized system to create and maintain class timetables, there is a need to innovate timetable management frameworks. The timetable display is an integrated management system that helps educational institutions to play an important role in the development and maintenance of timetables.

1.2 Project Purpose

One of the most schedules and error-prone applications is schedules timetables. Currently, the access device manually creates or produces a series of timetables, but much of the time it has trouble producing a clash-free and complete timetable. There are also major challenges, such as the generation of high-cost time tables during scheduling, and these issues are often replicated.

Most educational institutions have resorted to the manual generation of their schedules, which takes a lot of time to get done and optimum, according to statistics. There are always a few disagreements even at the optimum point of the manually generated timetable, and it is the professor who takes a competing course who sorts out the course's logistics to prevent the dispute. There is also a strong need for an application that distributes the course equally and without collisions. Our mission here is to create a quick, clearly interpreted, powerful and compact framework that can produce high-quality timetables automatically in seconds.

1.2.1 Background

There are routines for the administration of classes and time allotted to each educational institution in Bangladesh. Actually, a separate department in the organization manually creates these rituals.

When creating these routines, criteria such as the time of teachers, number of periods, class time, etc. for creating routines are often manually held in mind. Over time, routines still keep evolving. This makes their handling much more complicated due to the adjustment in value/state parameters. If an integrated device may be used for the above harsh method of establishing and managing the schedule, then it would allow an institution's department to loosen the pressure and improve productivity.

Due to the lack of such an automated framework for the development and maintenance of class routines, it is important to invent the Timetable Displayer. Timetable Displayer is an integrated management system composed of both smartphone and online software used for the efficient management of routine development and maintenance by educational institutions.

1.2.2 Benefits & Beneficiaries

We have beneficial aspects of this solution. They are pointed out below:

- Time consuming
- Reduce from extra hassle
- Digital Solution
- Student can easily view per semester class routine
- Easy to manage

1.2.3 Goals

The main purpose of this system is to create a routine in all educational instructions. If the just gives input through our system then the system will automatically create a routine. The student will be able to download a separate routine for each semester.

1.3 Stakeholders

There are two types of Stakeholders in our Solutions. They are:

- Admin: Admin can access all the features.
- Student: Student can view their profile class routine and download PDF format.

1.4 Project Schedule

Project Schedule is important for every project to complete on time.

1.4.1 Gantt Chart

It is a graphic view of overtime-planned tasks. It's a very efficient way to show on a particular day what work is planned to be completed. It also helps to demonstrate the start and finish times of my project.

Activities		W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15
Planning	Ideas															
	Problem definition															
	Proposal planning															
Requirements	Requirement Specification															
	Requirement analysis															
QA -1	Quality assurance															
System Design	Design specification															
	Interface design															
	Database design															
Development	Development system modules															
	Integrate system modules															
QA -2	Test Cases															
Testing	Unit testing															
	Black box testing															
Resolve Issues	Resolve issues found															
Release	Software release															

Figure 1.4.1: Gantt chart

1.4.2 Project Timeline

The Project timeline record is given below:

Task	Date
Topic Selection Seminar	18/06/2020
Brainstorming Project Topic & Name Selection	20/06/2020 - 29/06/2020
Submit Project Proposal	30/06/2020
Requirement Specification	15/07/2020 - 27/07/2020
Mid Term Defense	04/11/2020
System Analysis & Design	10/08/2020 - 25/08/2020
Development System	02/09/2020 - 25/12/2020
System Testing	20/12/2020 - 29/12/2020
Complete Project	30/12/2020

Table 1.4.2: Project Timeline

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Function Requirement: Admin

2.1.1 Log in

Requirement ID	FR.A.1
Requirement Name	Log in
Description	Admin can log in using Email and password
Stakeholder	Admin

2.1.2 Periods manage

Requirement ID	FR.A.2
Requirement Name	Periods manage
Description	Admin can manage time slots.
Stakeholder	Admin

2.1.3 Room manage

Requirement ID	FR.A.3
Requirement Name	Room manage
Description	Admin can manage all room's details.
Stakeholder	Admin

2.1.4 Course manage

Requirement ID	FR.A.4
Requirement Name	Course manage
Description	Admin can manage all course details.
Stakeholder	Admin

2.1.5 Course Teachers manage

Requirement ID	FR.A.5
Requirement Name	Course Teacher manage
Description	Admin can manage all teacher's details.
Stakeholder	Admin

2.1.6 Classes manage

Requirement ID	FR.A.6
Requirement Name	Classes manage
Description	Admin can manage all semester and section.
Stakeholder	Admin

2.1.7 Create Timetable

Requirement ID	FR.A.7
Requirement Name	Create Timetable
Description	Every semester routine can be created by admin.
Stakeholder	Admin

2.2 Function Requirement: Student

2.2.1 Registration

Requirement ID	FR.S.1
Requirement Name	Registration
Description	Students may create their own accounts.
Stakeholder	Student

2.2.2 Login

Requirement ID	FR.S.2
Requirement Name	Login
Description	Student can log in using Email and password.
Stakeholder	Student

2.2.3 View Timetable

Requirement ID	FR.S.3
Requirement Name	View timetable
Description	Admin and Students can see all Semester routines.
Stakeholder	Admin/Student

2.2.4 Download PDF

Requirement ID	FR.S.4
Requirement Name	PDF Download
Description	Admin and Students can download routine in a PDF format.
Stakeholder	Admin/Student

2.3 Performance Requirements

The requirement of dependability encompasses reliability, protection, security and availability, but reliability is the main issue. These specifications are also necessary.

2.3.1 Capacity Requirements

The system must be capable of handling all types of user data.

CR-1	System will handle many data
Description	System need to handle many type of data
Stakeholder	Admin

2.4 Dependability Requirements

Dependability is measured by four dimensions. Such as: Availability, Reliability, Safety, and Security. So, our system should fulfil these four types of dimension.

2.4.1 Reliability Requirements

Reliability means the probability that the system will operate without failure

RR-1	System must be available 24/7
Description	System must be available all the time and updated and malware free
Stakeholder	N/A

2.4.2 Fault-Tolerance Requirements

To ensure fault-tolerance to the users, it is important to ensure 0% crash and must show accurate results.

FTR-1	System operate all the user access without system error
Description	Both user can access our system at a time, system must be handle request without any errors
Stakeholder	N/A

2.4.3 Safety-Critical Requirements

There is no safety-critical requirement in my project.

2.5 Maintainability and Supportability Requirements

It is very important to provide after support or service to the end users.

2.5.1 Maintenance Requirements

MR-1	System helps to create routine and delete routine
Description	It is very important.
Stakeholder	Admin

2.5.2 Supportability Requirements

There is some dimension of supportability requirements. They are:

- Maintainability
- Configurability
- Compatibility
- Serviceability

2.5.3 Adaptability Requirements

There are no adaptability requirements of my project.

2.6 Security Requirements

Security requirements for system solutions is very important. It should be functional requirements. Software Security set on security of an application system. There are some security related requirements. They are:

- Sign in a admin or student
- Get access according to logged in user
- Sign out as a admin or student

2.6.1 Access Requirements

There is no access requirement in my project.

2.6.2 Integrity Requirements

There is no integrity requirement in my project.

2.6.3 Privacy Requirements

It is important to have privacy requirements in every system. To ensure privacy, every user will enter the system by verifying their information and they will use their accessibility features.

2.7 Usability and Human-Interaction Requirements

The main target of developing any system solution is to make it user friendly and easy for the users.

2.7.1. Ease of Use Requirements

Our Solution is easy to use and understandable

EUR-1	System must be usable & easy for the user
Description	This solution is easy for user to manage system
Stakeholder	Admin, student

2.7.2 Understandability Requirements

Here no specific understandably requirements in my project.

2.7.3 Accessibility Requirements

Here no specific accessibility requirements in my project.

2.7.4 User Documentation Requirements

No user documentation requirements in my project.

2.7.5 Training Requirements

Don't need any training requirements in my project.

CHAPTER 3

SYSTEM ANALYSIS & DESIGN

3.1 Use Case Diagram

There are two actors in our use case diagram. This diagram will refine my project in details

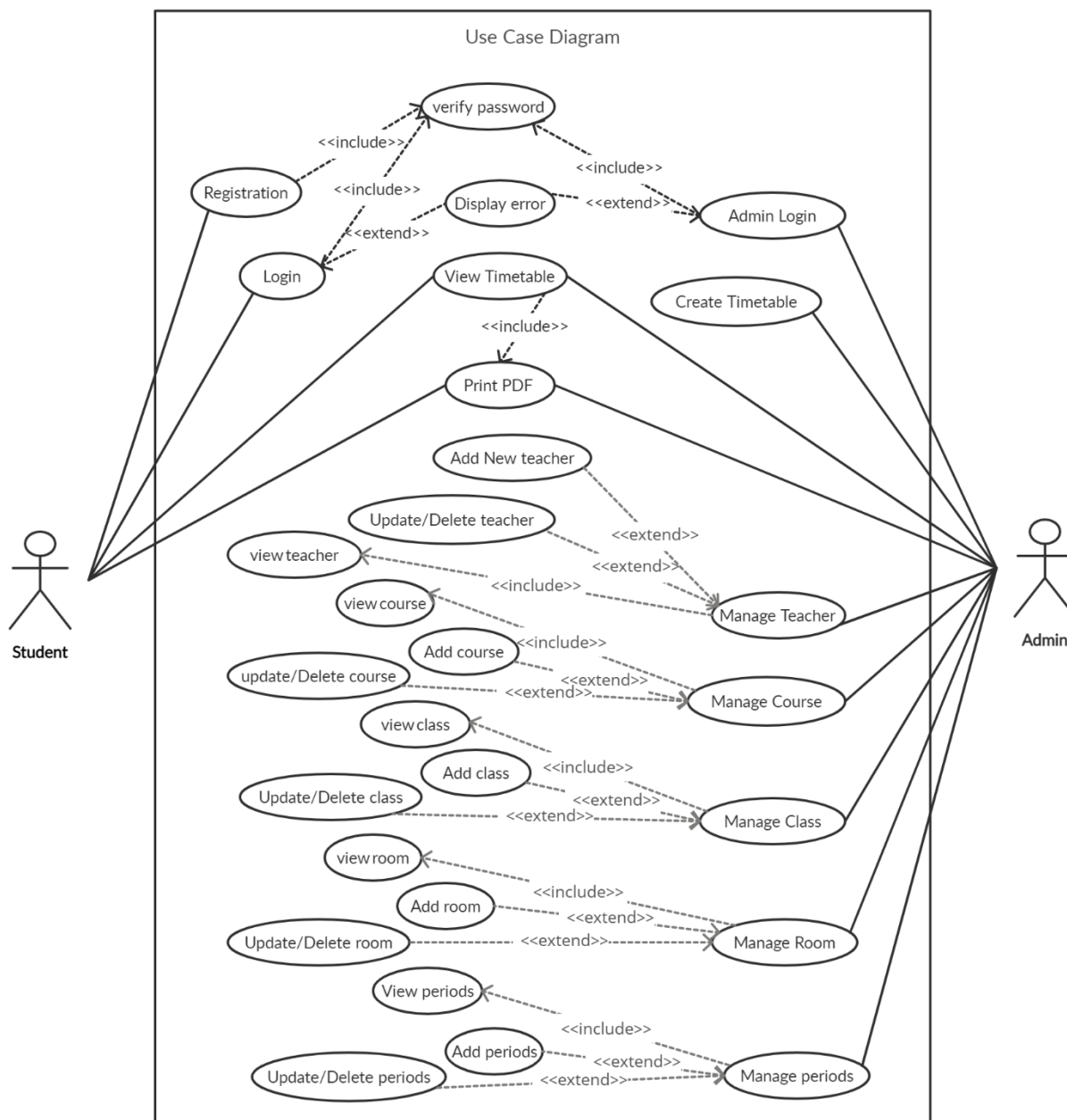


Figure 3.1: Use case for Timetable Displayer [1]

3.2 Use Case Description

3.2.1 Admin Login

Use case	Log In								
Goal	This will only be the selected person login for this system.								
Pre-condition	Its fixed account for login There is no need to register								
Primary actors Secondary actors	Admin								
Trigger	This will be text box and button in a card								
Description/ Main success	<table><thead><tr><th>Step</th><th>Action</th></tr></thead><tbody><tr><td>1.</td><td>Enter the website</td></tr><tr><td>2.</td><td>Enter email & password</td></tr><tr><td>3.</td><td>Logged In</td></tr></tbody></table>	Step	Action	1.	Enter the website	2.	Enter email & password	3.	Logged In
Step	Action								
1.	Enter the website								
2.	Enter email & password								
3.	Logged In								
Alternative flows	N/A								
Quality requirement	N/A								

Table 3.2.1: Admin Login

3.2.2 Admin Manage Periods

Use case	Manage periods														
Goal	Create new timeslots & view timeslots list, update and delete timeslots.														
Pre-condition	Must be logged in														
Primary actors Secondary actors	Admin														
Trigger	This will be button & form														
Description/ Main success	<table><thead><tr><th>Step</th><th>Action</th></tr></thead><tbody><tr><td>1.</td><td>login the system</td></tr><tr><td>2.</td><td>Click into “Periods” from sidebar</td></tr><tr><td>3.</td><td>Show timeslot’s list.</td></tr><tr><td>4.</td><td>Click ‘Add new timeslot’s’</td></tr><tr><td>5.</td><td>Entry time slots like 8:30 am- 10:00 am.</td></tr><tr><td>6.</td><td>Click Submit button</td></tr></tbody></table>	Step	Action	1.	login the system	2.	Click into “Periods” from sidebar	3.	Show timeslot’s list.	4.	Click ‘Add new timeslot’s’	5.	Entry time slots like 8:30 am- 10:00 am.	6.	Click Submit button
Step	Action														
1.	login the system														
2.	Click into “Periods” from sidebar														
3.	Show timeslot’s list.														
4.	Click ‘Add new timeslot’s’														
5.	Entry time slots like 8:30 am- 10:00 am.														
6.	Click Submit button														
Alternative flows	<table><tbody><tr><td>1.</td><td>Click edit button</td></tr><tr><td>2.</td><td>Edit all details and click update button</td></tr><tr><td>3.</td><td>Click delete button</td></tr><tr><td>4.</td><td>Delete timeslot and click delete button</td></tr></tbody></table>	1.	Click edit button	2.	Edit all details and click update button	3.	Click delete button	4.	Delete timeslot and click delete button						
1.	Click edit button														
2.	Edit all details and click update button														
3.	Click delete button														
4.	Delete timeslot and click delete button														
Quality requirement	N/A														

Table 3.2.2: Admin Manage Periods

3.2.3 Admin Manage rooms

Use case	Manage rooms														
Goal	Admin can new room & view room list, update and delete rooms.														
Pre-condition	Must be logged in														
Primary actors Secondary actors	Admin														
Trigger	This will be button & form														
Description/ Main success	<table> <thead> <tr> <th>Step</th><th>Action</th></tr> </thead> <tbody> <tr> <td>1.</td><td>login the system</td></tr> <tr> <td>2.</td><td>Click into “Rooms” from sidebar</td></tr> <tr> <td>3.</td><td>Show room’s list.</td></tr> <tr> <td>4.</td><td>Click ‘Add new room’</td></tr> <tr> <td>5.</td><td>Entry room name and size.</td></tr> <tr> <td>6.</td><td>Click Submit button</td></tr> </tbody> </table>	Step	Action	1.	login the system	2.	Click into “Rooms” from sidebar	3.	Show room’s list.	4.	Click ‘Add new room’	5.	Entry room name and size.	6.	Click Submit button
Step	Action														
1.	login the system														
2.	Click into “Rooms” from sidebar														
3.	Show room’s list.														
4.	Click ‘Add new room’														
5.	Entry room name and size.														
6.	Click Submit button														
Alternative flows	<table> <tbody> <tr> <td>1.</td><td>Click edit button</td></tr> <tr> <td>2.</td><td>Edit all details and click update button</td></tr> <tr> <td>3.</td><td>Click delete button</td></tr> <tr> <td>4.</td><td>Delete timeslot and click delete button</td></tr> </tbody> </table>	1.	Click edit button	2.	Edit all details and click update button	3.	Click delete button	4.	Delete timeslot and click delete button						
1.	Click edit button														
2.	Edit all details and click update button														
3.	Click delete button														
4.	Delete timeslot and click delete button														
Quality requirement	N/A														

Table 3.2.3: Admin Manage Rooms

3.2.4 Admin Manage courses

Use case	Manage courses														
Goal	Admin can add new course & view course details list, update and delete course.														
Pre-condition	Must be logged in														
Primary actors Secondary actors	Admin														
Trigger	This will be button & form														
Description/ Main success	<table border="1"> <thead> <tr> <th>Step</th><th>Action</th></tr> </thead> <tbody> <tr> <td>1.</td><td>login the system</td></tr> <tr> <td>2.</td><td>Click into “Courses” from sidebar</td></tr> <tr> <td>3.</td><td>Show course list.</td></tr> <tr> <td>4.</td><td>Click ‘Add new course’</td></tr> <tr> <td>5.</td><td>Entry course name, course code, course teacher</td></tr> <tr> <td>6.</td><td>Click Submit button</td></tr> </tbody> </table>	Step	Action	1.	login the system	2.	Click into “Courses” from sidebar	3.	Show course list.	4.	Click ‘Add new course’	5.	Entry course name, course code, course teacher	6.	Click Submit button
Step	Action														
1.	login the system														
2.	Click into “Courses” from sidebar														
3.	Show course list.														
4.	Click ‘Add new course’														
5.	Entry course name, course code, course teacher														
6.	Click Submit button														
Alternative flows	<table border="1"> <tbody> <tr> <td>1.</td><td>Click edit button</td></tr> <tr> <td>2.</td><td>Edit all details and click update button</td></tr> <tr> <td>3.</td><td>Click delete button</td></tr> <tr> <td>4.</td><td>Delete timeslot and click delete button</td></tr> </tbody> </table>	1.	Click edit button	2.	Edit all details and click update button	3.	Click delete button	4.	Delete timeslot and click delete button						
1.	Click edit button														
2.	Edit all details and click update button														
3.	Click delete button														
4.	Delete timeslot and click delete button														
Quality requirement	N/A														

Table 3.2.4: Admin Manage Courses

3.2.5 Admin Manage course Teachers

Use case	Manage course Teachers														
Goal	Admin can add new Teacher & view teacher details list, update and delete teacher.														
Pre-condition	Must be logged in														
Primary actors Secondary actors	Admin														
Trigger	This will be button & form														
Description/ Main success	<table border="1"> <thead> <tr> <th>Step</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>login the system</td> </tr> <tr> <td>2.</td> <td>Click into “course Teacher” from sidebar</td> </tr> <tr> <td>3.</td> <td>Show course Teacher’s list.</td> </tr> <tr> <td>4.</td> <td>Click ‘Add new Teacher’</td> </tr> <tr> <td>5.</td> <td>Entry name, email, courses, unavailable periods</td> </tr> <tr> <td>6.</td> <td>Click Submit button</td> </tr> </tbody> </table>	Step	Action	1.	login the system	2.	Click into “course Teacher” from sidebar	3.	Show course Teacher’s list.	4.	Click ‘Add new Teacher’	5.	Entry name, email, courses, unavailable periods	6.	Click Submit button
Step	Action														
1.	login the system														
2.	Click into “course Teacher” from sidebar														
3.	Show course Teacher’s list.														
4.	Click ‘Add new Teacher’														
5.	Entry name, email, courses, unavailable periods														
6.	Click Submit button														
Alternative flows	<table border="1"> <tbody> <tr> <td>1.</td> <td>Click edit button</td> </tr> <tr> <td>2.</td> <td>Edit all details and click update button</td> </tr> <tr> <td>3.</td> <td>Click delete button</td> </tr> <tr> <td>4.</td> <td>Delete timeslot and click delete button</td> </tr> </tbody> </table>	1.	Click edit button	2.	Edit all details and click update button	3.	Click delete button	4.	Delete timeslot and click delete button						
1.	Click edit button														
2.	Edit all details and click update button														
3.	Click delete button														
4.	Delete timeslot and click delete button														
Quality requirement	N/A														

Table 3.2.5: Admin Manage Course Teachers

3.2.6 Admin Manage Classes

Use case	Manage classes														
Goal	Admin can add new class & view class details list, update and delete class.														
Pre-condition	Must be logged in														
Primary actors Secondary actors	Admin														
Trigger	This will be button & form														
Description/ Main success	<table border="1"> <thead> <tr> <th>Step</th><th>Action</th></tr> </thead> <tbody> <tr> <td>1.</td><td>login the system</td></tr> <tr> <td>2.</td><td>Click into “classes” from sidebar</td></tr> <tr> <td>3.</td><td>Show Classes list.</td></tr> <tr> <td>4.</td><td>Click ‘Add new class’</td></tr> <tr> <td>5.</td><td>Entry name, Multiple course, semester, class per week, people, unavailable room</td></tr> <tr> <td>6.</td><td>Click Submit button</td></tr> </tbody> </table>	Step	Action	1.	login the system	2.	Click into “classes” from sidebar	3.	Show Classes list.	4.	Click ‘Add new class’	5.	Entry name, Multiple course, semester, class per week, people, unavailable room	6.	Click Submit button
Step	Action														
1.	login the system														
2.	Click into “classes” from sidebar														
3.	Show Classes list.														
4.	Click ‘Add new class’														
5.	Entry name, Multiple course, semester, class per week, people, unavailable room														
6.	Click Submit button														
Alternative flows	<table border="1"> <tbody> <tr> <td>1.</td><td>Click edit button</td></tr> <tr> <td>2.</td><td>Edit all details and click update button</td></tr> <tr> <td>3.</td><td>Click delete button</td></tr> <tr> <td>4.</td><td>Delete timeslot and click delete button</td></tr> </tbody> </table>	1.	Click edit button	2.	Edit all details and click update button	3.	Click delete button	4.	Delete timeslot and click delete button						
1.	Click edit button														
2.	Edit all details and click update button														
3.	Click delete button														
4.	Delete timeslot and click delete button														
Quality requirement	N/A														

Table 3.2.6: Admin Manage Classes

3.2.7 Admin Create New Timetable

Use case	Create New Timetable										
Goal	Admin can create new Timetable.										
Pre-condition	Must be logged in										
Primary actors Secondary actors	Admin										
Trigger	This will be button & form										
Description/ Main success	<table><thead><tr><th>Step</th><th>Action</th></tr></thead><tbody><tr><td>1.</td><td>login the system</td></tr><tr><td>2.</td><td>Click into “Create New Timetable” from dashboard</td></tr><tr><td>3.</td><td>Enter Timetable name, select semester, select days</td></tr><tr><td>4.</td><td>Click Submit button</td></tr></tbody></table>	Step	Action	1.	login the system	2.	Click into “Create New Timetable” from dashboard	3.	Enter Timetable name, select semester, select days	4.	Click Submit button
Step	Action										
1.	login the system										
2.	Click into “Create New Timetable” from dashboard										
3.	Enter Timetable name, select semester, select days										
4.	Click Submit button										
Alternative flows	N/A										
Quality requirement	N/A										

Table 3.2.7: Admin Manage Create New Timetable

3.2.8 Student Registration

Use case	Student Registration										
Goal	Every student can create an account.										
Pre-condition	Register with student name, email, password & confirm password.										
Primary actors Secondary actors	Student										
Trigger	This will be text box and button in a card										
Description/ Main success	<table><thead><tr><th>Step</th><th>Action</th></tr></thead><tbody><tr><td>1.</td><td>Enter the website</td></tr><tr><td>2.</td><td>Click into 'sign up'</td></tr><tr><td>3.</td><td>Enter name, email, password & confirm password and click submit button</td></tr><tr><td>4.</td><td>Successfully register.</td></tr></tbody></table>	Step	Action	1.	Enter the website	2.	Click into 'sign up'	3.	Enter name, email, password & confirm password and click submit button	4.	Successfully register.
Step	Action										
1.	Enter the website										
2.	Click into 'sign up'										
3.	Enter name, email, password & confirm password and click submit button										
4.	Successfully register.										
Alternative flows	N/A										
Quality requirement	N/A										

Table 3.2.8: Student Registration

3.2.9 Student Login

Use case	Student Log In								
Goal	This will only be the registered student login for this system.								
Pre-condition	There is need to register								
Primary actors Secondary actors	Student								
Trigger	This will be text box and button in a card								
Description/ Main success	<table><thead><tr><th>Step</th><th>Action</th></tr></thead><tbody><tr><td>1.</td><td>Enter the website</td></tr><tr><td>2.</td><td>Enter email & password</td></tr><tr><td>3.</td><td>Logged In</td></tr></tbody></table>	Step	Action	1.	Enter the website	2.	Enter email & password	3.	Logged In
Step	Action								
1.	Enter the website								
2.	Enter email & password								
3.	Logged In								
Alternative flows	N/A								
Quality requirement	N/A								

Table 3.2.9: Student Login

3.2.10 View Timetable

Use case	View Timetable								
Goal	User can view all timetable list.								
Pre-condition	Must have to login.								
Actors	Admin & Student								
Trigger	This will be show dashboard.								
Description/ Main success	<table><tr><td>Step</td><td>Action</td></tr><tr><td>1.</td><td>Login The System</td></tr><tr><td>2.</td><td>Click into “dashboard” from sidebar</td></tr><tr><td>3.</td><td>Show all timetable list.</td></tr></table>	Step	Action	1.	Login The System	2.	Click into “dashboard” from sidebar	3.	Show all timetable list.
Step	Action								
1.	Login The System								
2.	Click into “dashboard” from sidebar								
3.	Show all timetable list.								
Alternative flows	N/A								
Quality requirement	N/A								

Table 3.2.10: View Timetable

3.2.11 Print PDF

Use case	Print PDF						
Goal	User can download PDF						
Pre-condition	Must have to login.						
Actors	Admin & Student						
Trigger	This will be button						
Description/ Main success	<table><tr><td>Step</td><td>Action</td></tr><tr><td>1.</td><td>Login The System</td></tr><tr><td>2.</td><td>Click into “Print PDF” from view timetable</td></tr></table>	Step	Action	1.	Login The System	2.	Click into “Print PDF” from view timetable
Step	Action						
1.	Login The System						
2.	Click into “Print PDF” from view timetable						
Alternative flows	N/A						
Quality requirement	N/A						

3.3 Activity Diagram

3.3.1 Admin Login Activity

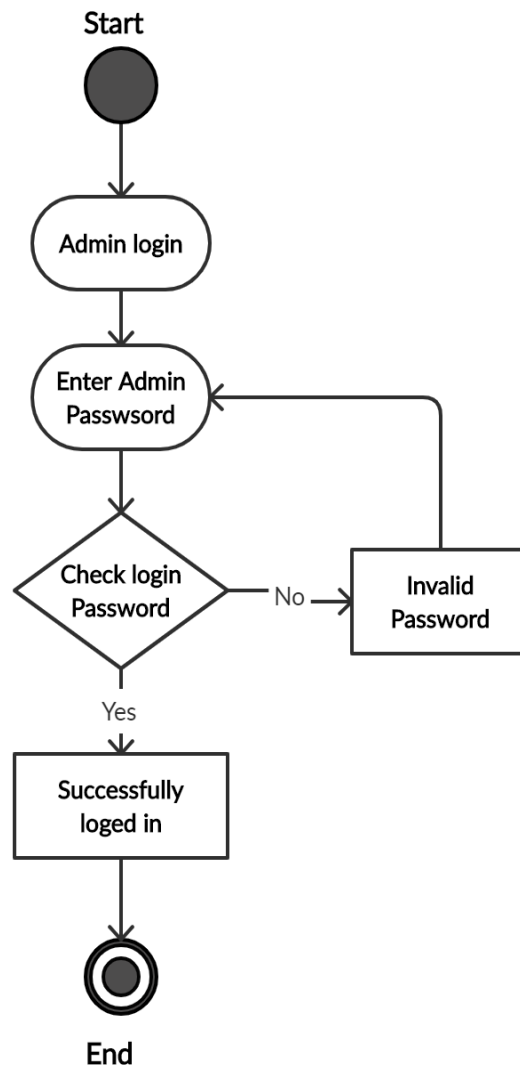


Figure 3.3.1: Admin Login Activity Diagram [1]

3.3.2 Admin Manage Periods Activity

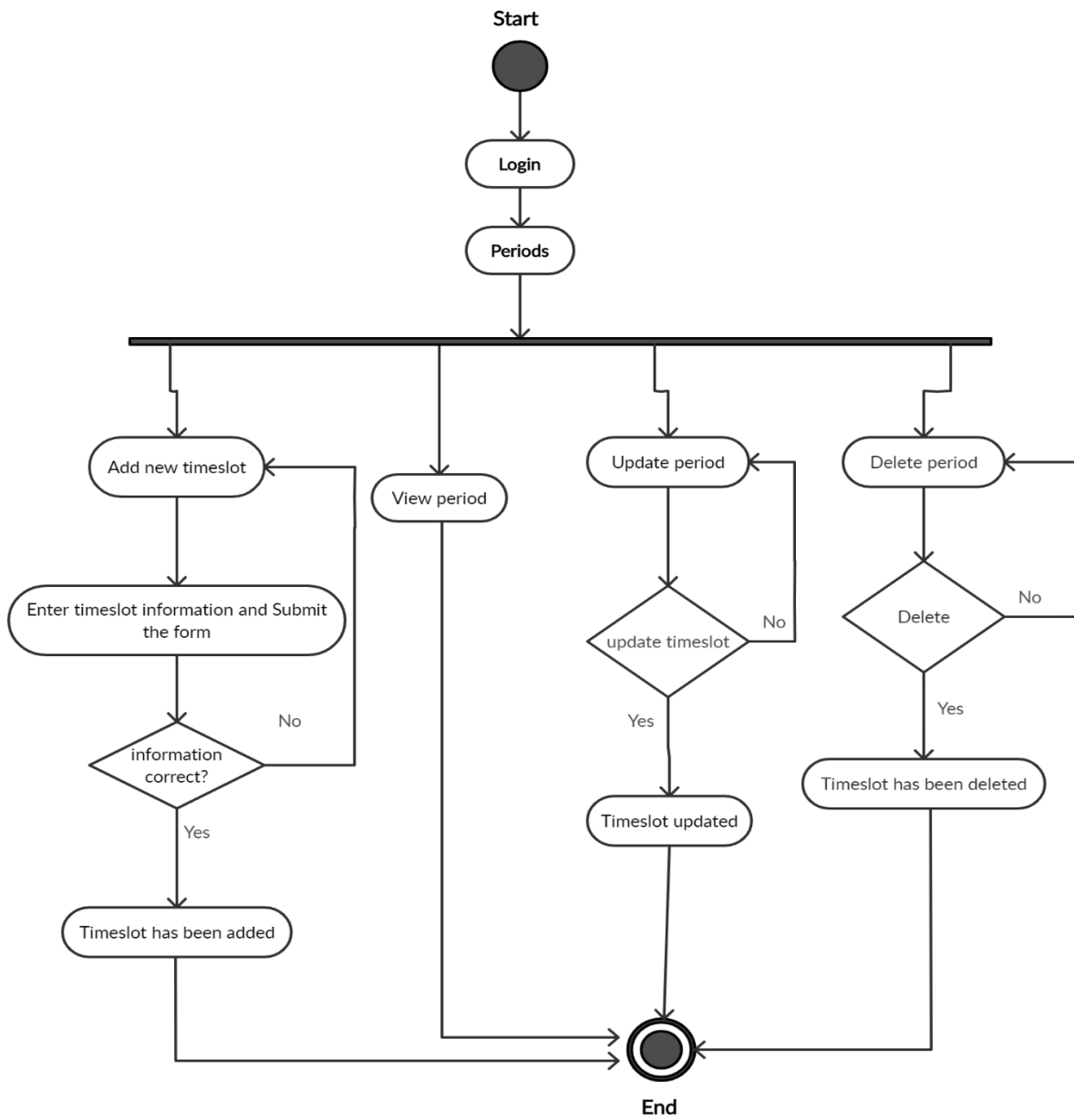


Figure 3.3.2: Admin Manage Periods Activity Diagram

3.3.3 Admin Manage Rooms Activity

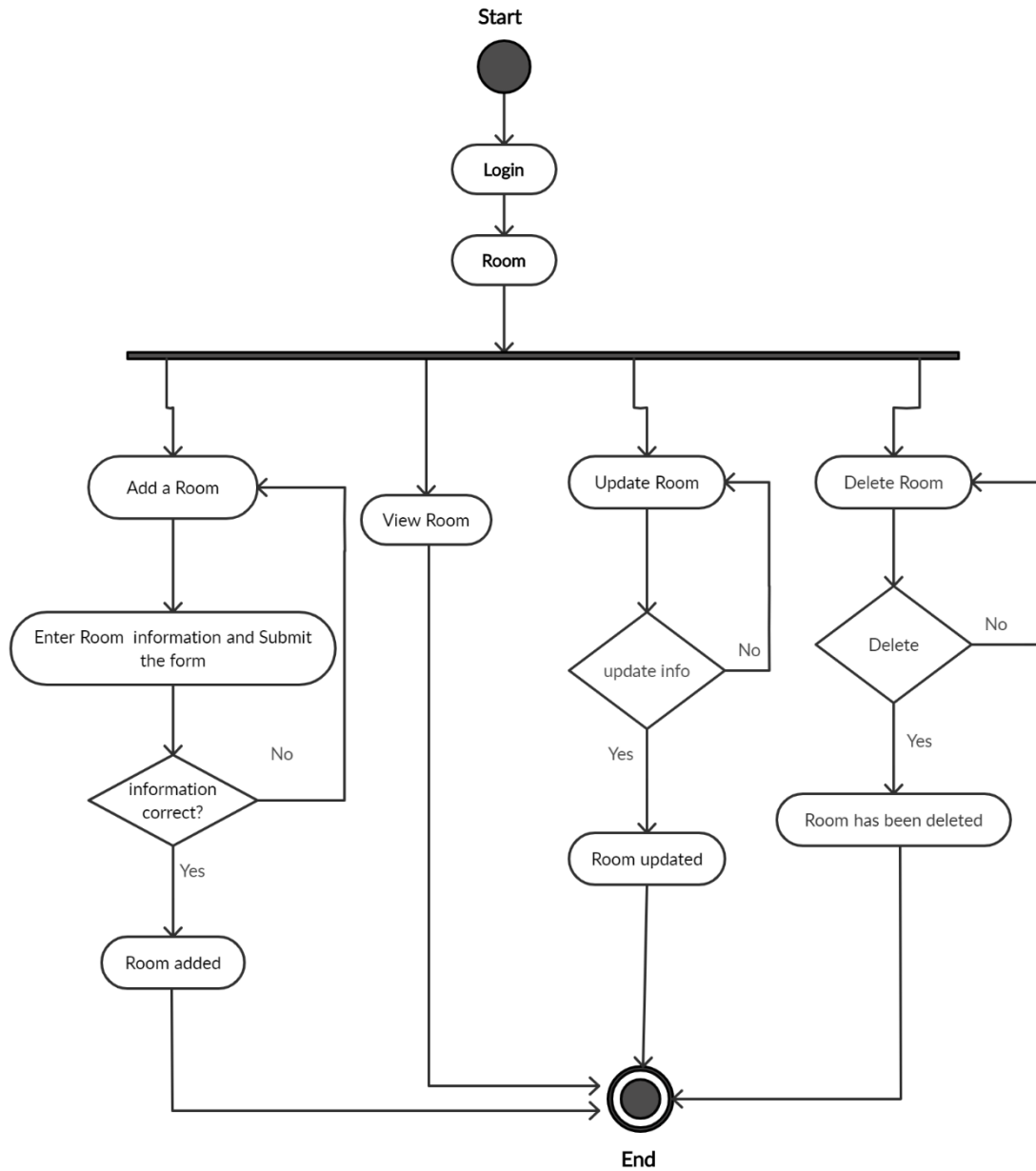


Figure 3.3.3: Admin Manage Rooms Activity Diagram

3.3.4 Admin Manage Courses Activity

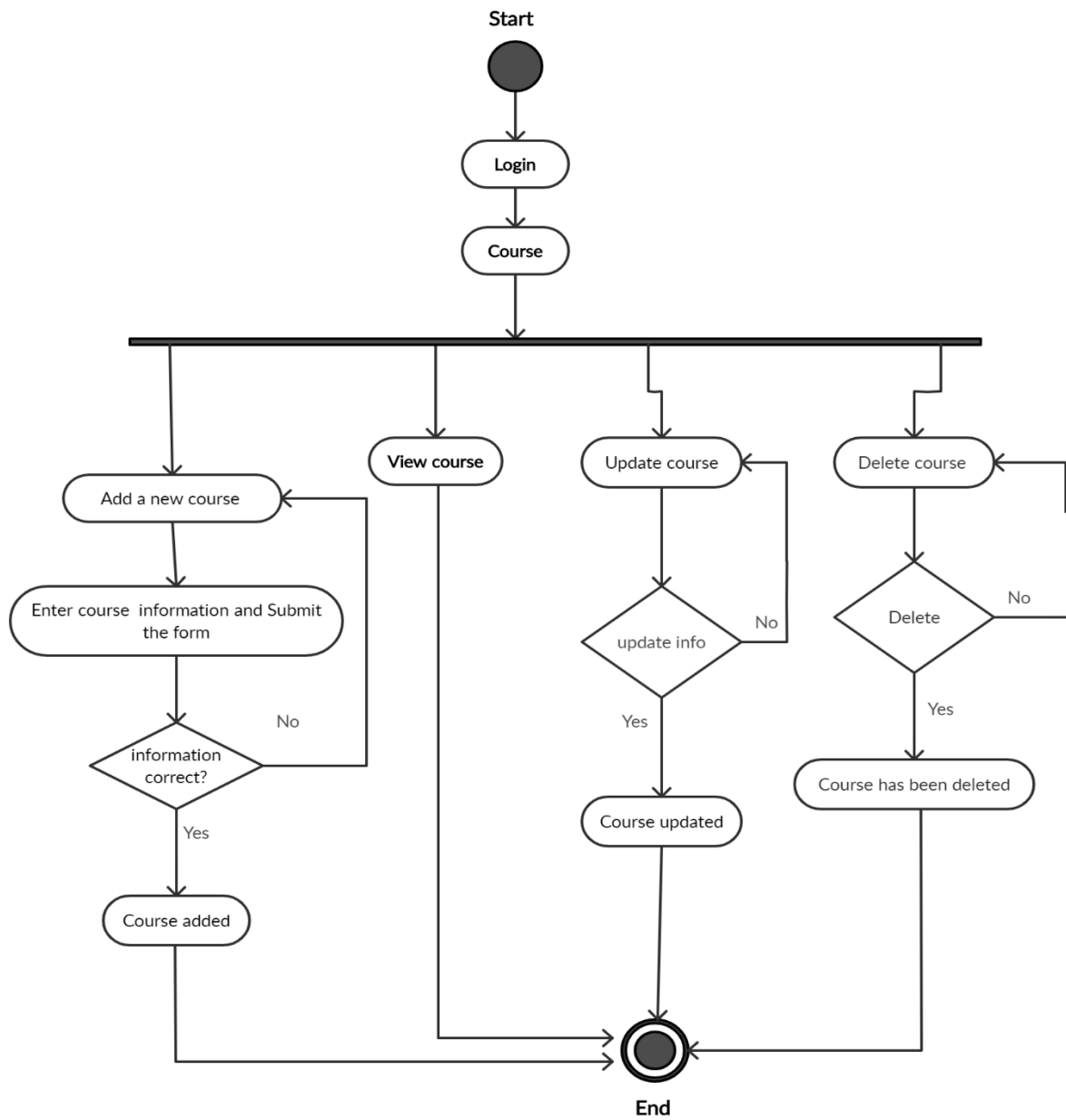


Figure 3.3.4: Admin Manage Courses Activity Diagram

3.3.5 Admin Manage Course Teacher Activity

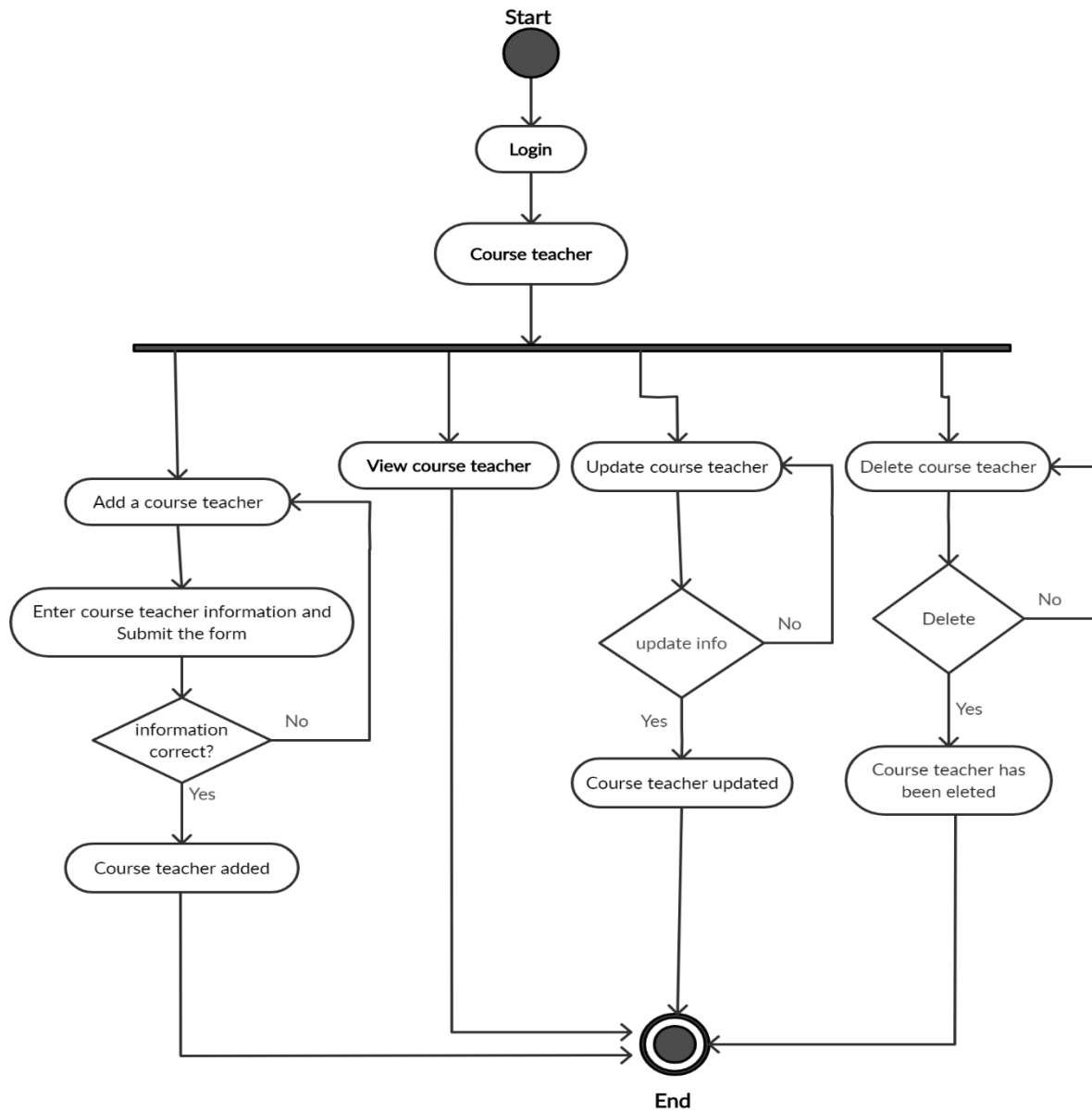


Figure 3.3.5: Admin Manage Course Teacher Activity Diagram

3.3.6 Admin Manage Classes Activity

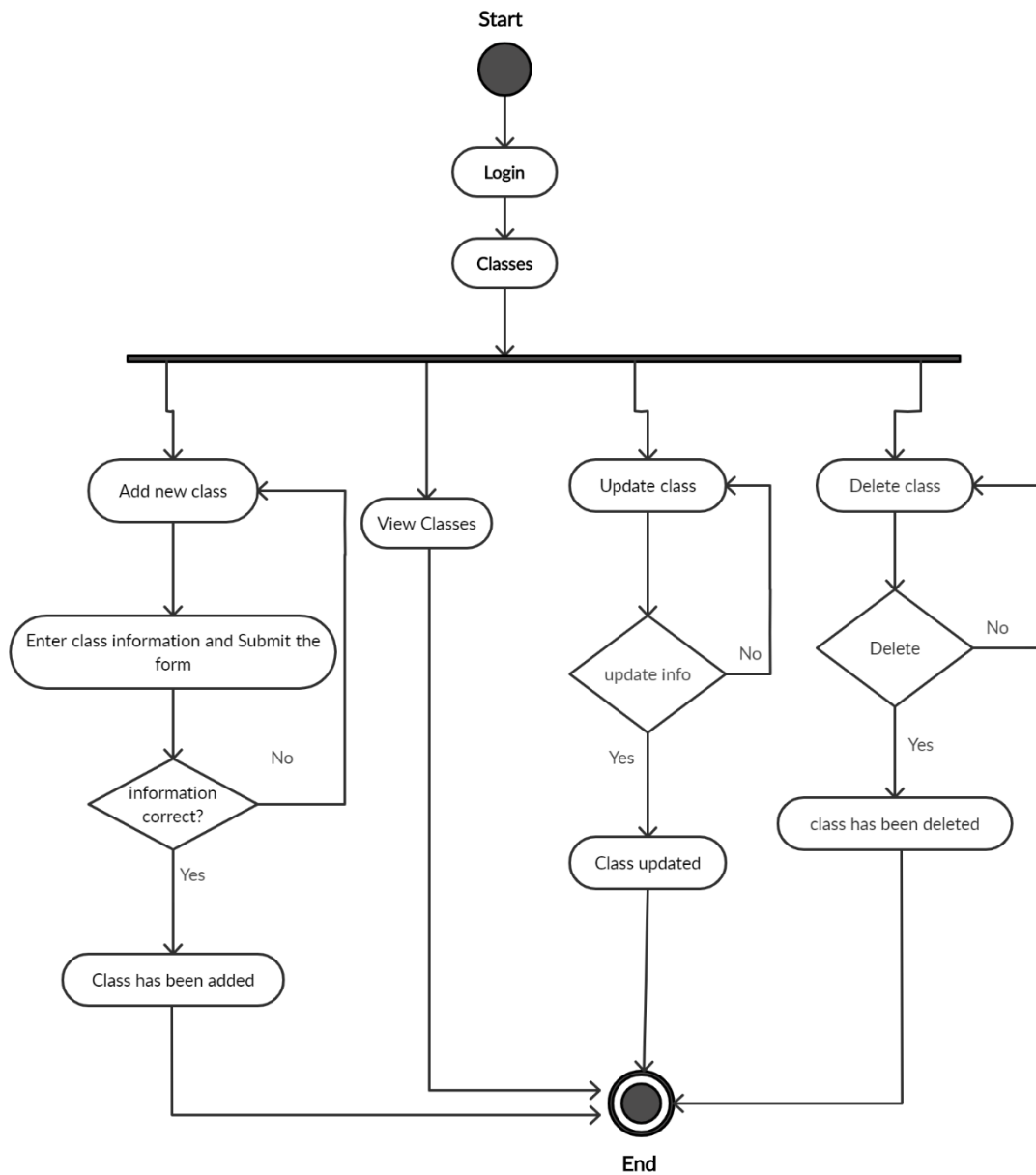


Figure 3.3.6: Admin Manage Classes Activity Diagram

3.3.7 Admin Create Timetable Activity

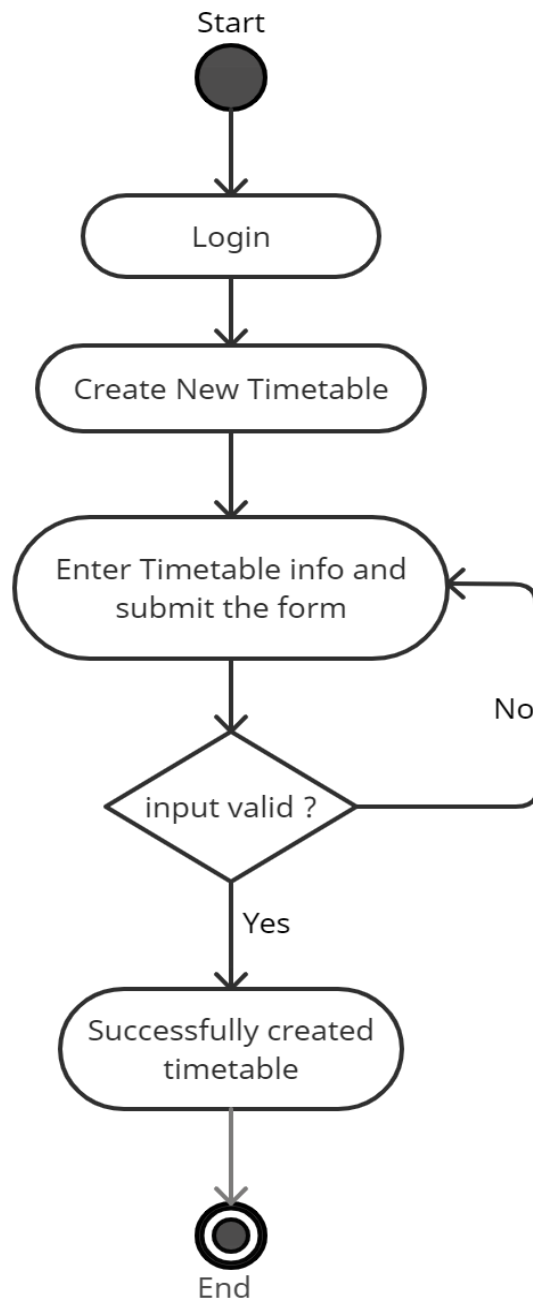


Figure 3.3.7: Admin Create Timetable Activity Diagram

3.3.8 Student Registration

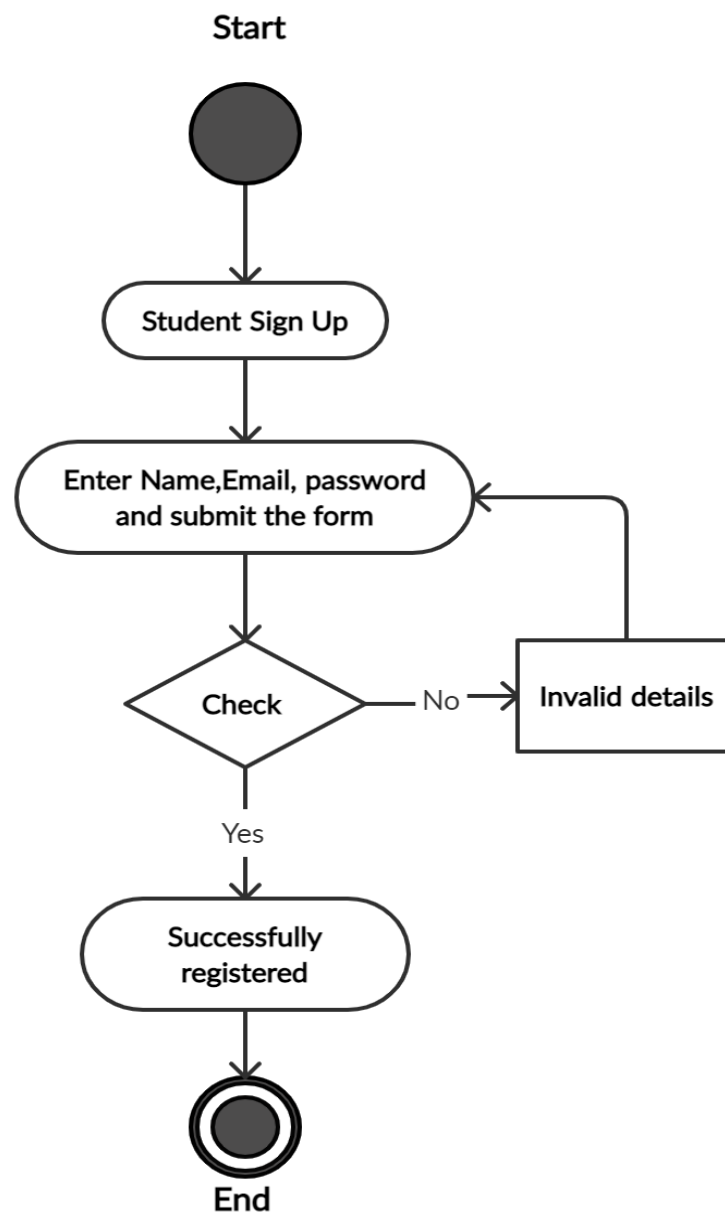


Figure 3.3.8: Student Registration Activity Diagram

3.3.9 Student Login

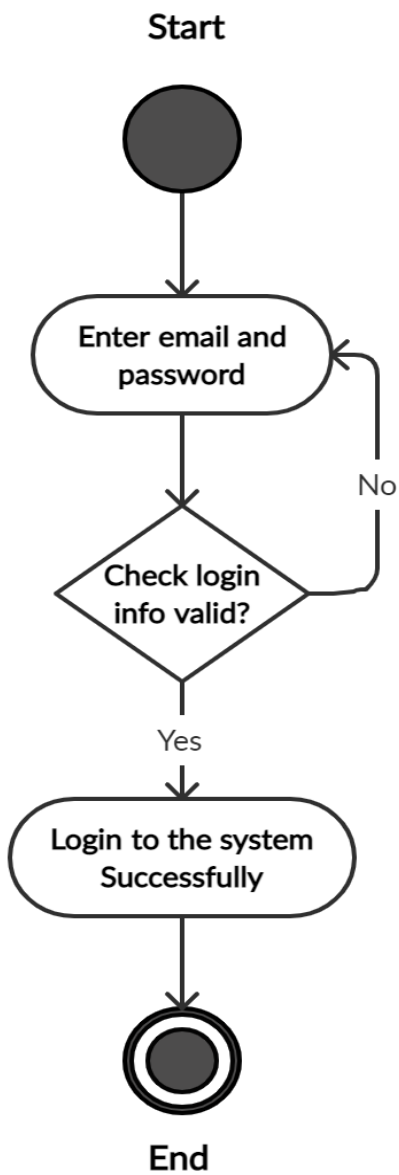


Figure 3.3.9: Student Login Activity Diagram

3.3.10 View Timetable Activity

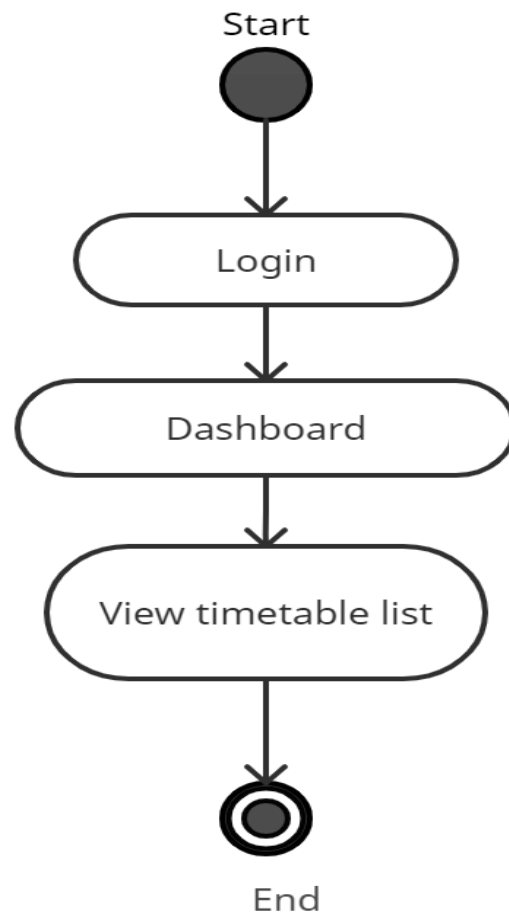


Figure 3.3.10: View Timetable Activity Diagram

3.3.11 Print PDF activity

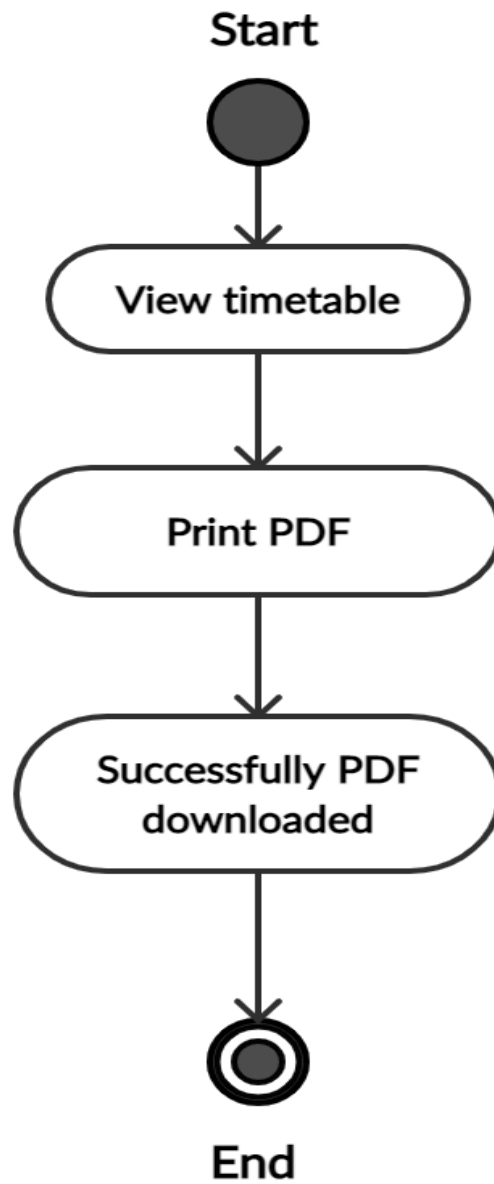


Figure 3.3.11: Print PDF activity Diagram

3.4 System Sequence Diagram

3.4.1 Admin Login

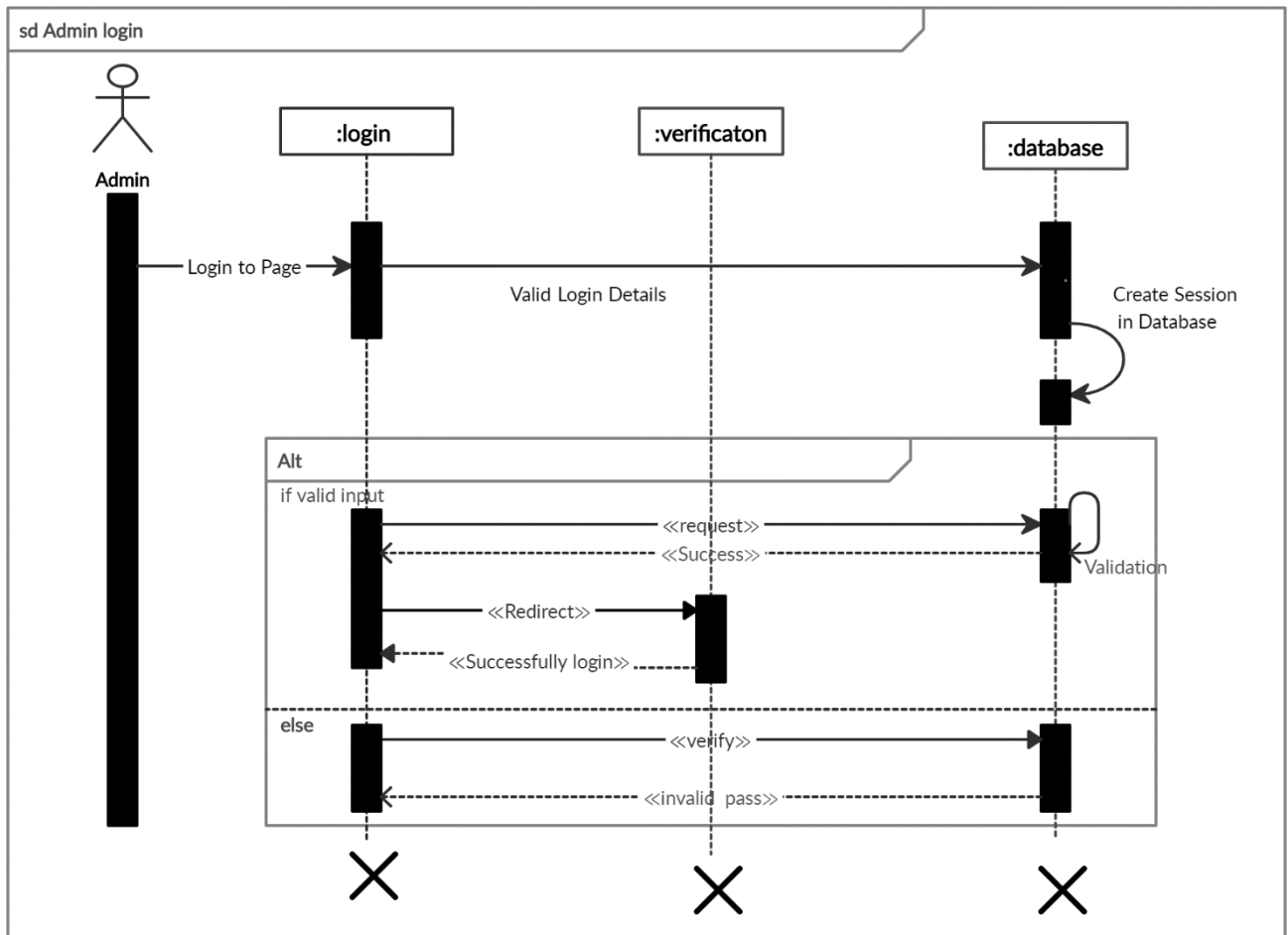


Figure 3.4.1: Admin Login Sequence Diagram

3.4.2 Admin Manage Periods

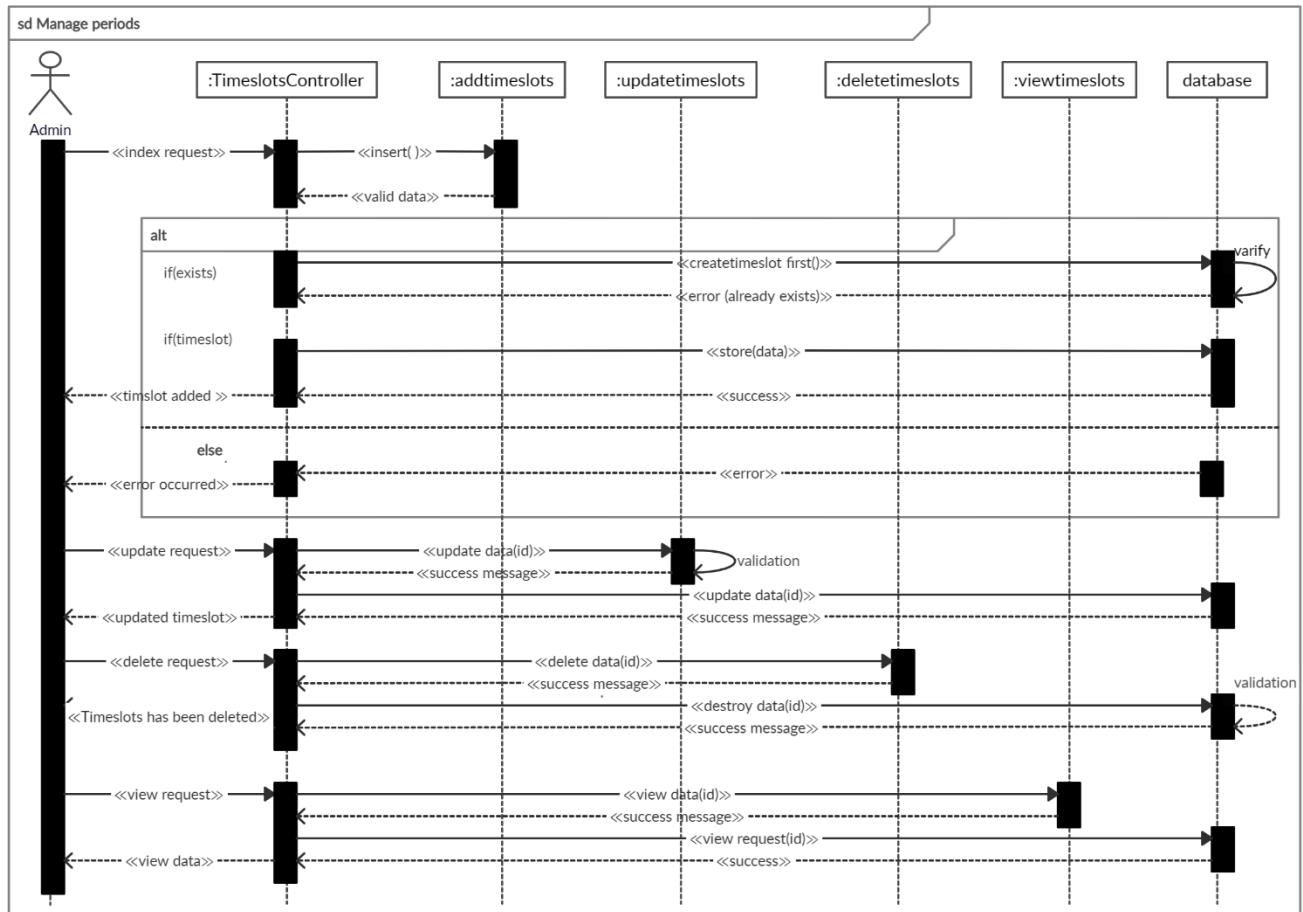


Figure 3.4.2: Admin Manage Periods Sequence Diagram

3.4.3 Admin Manage Rooms

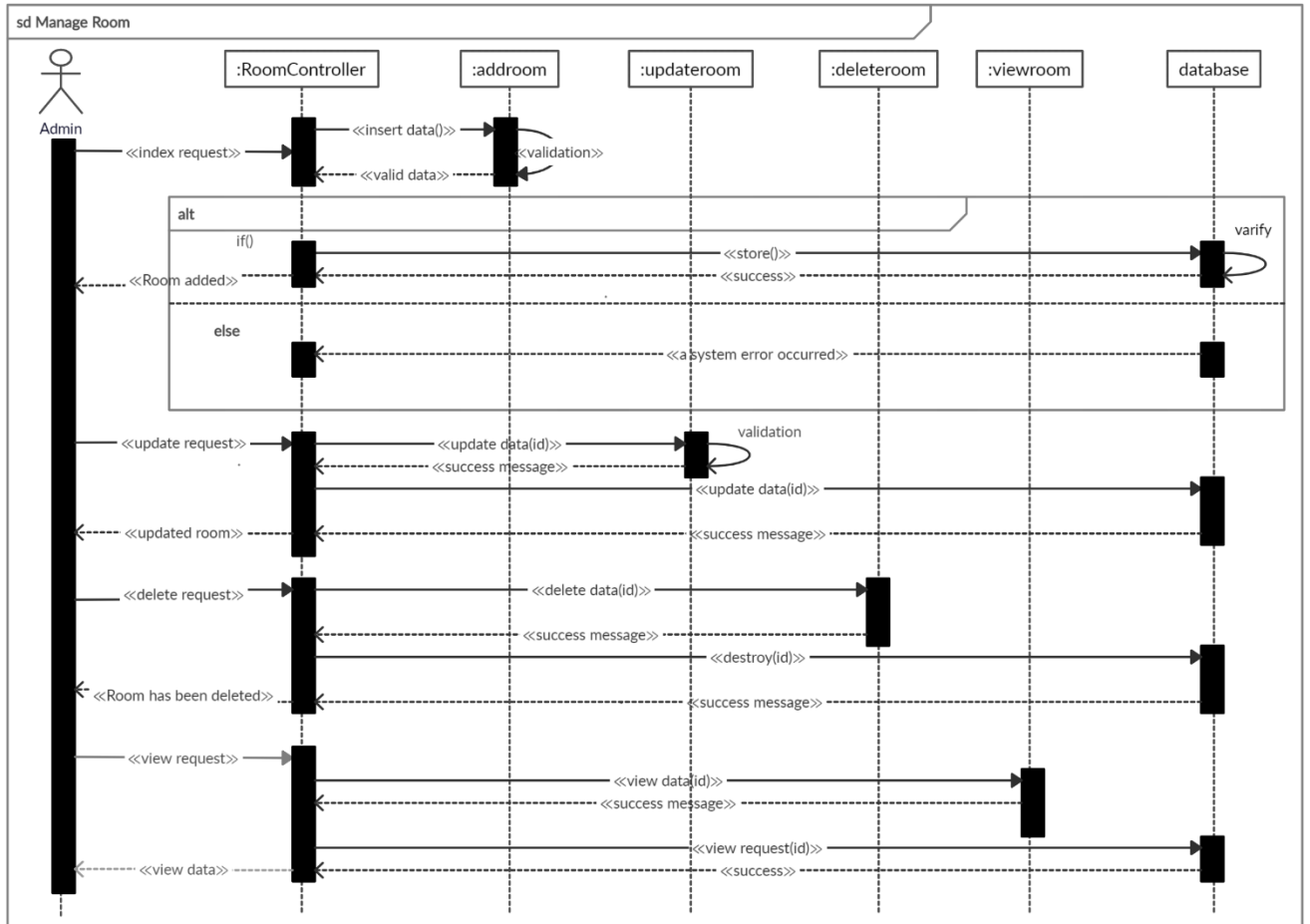


Figure 3.4.3: Admin Manage Rooms Sequence Diagram

3.4.4 Admin Manage Courses

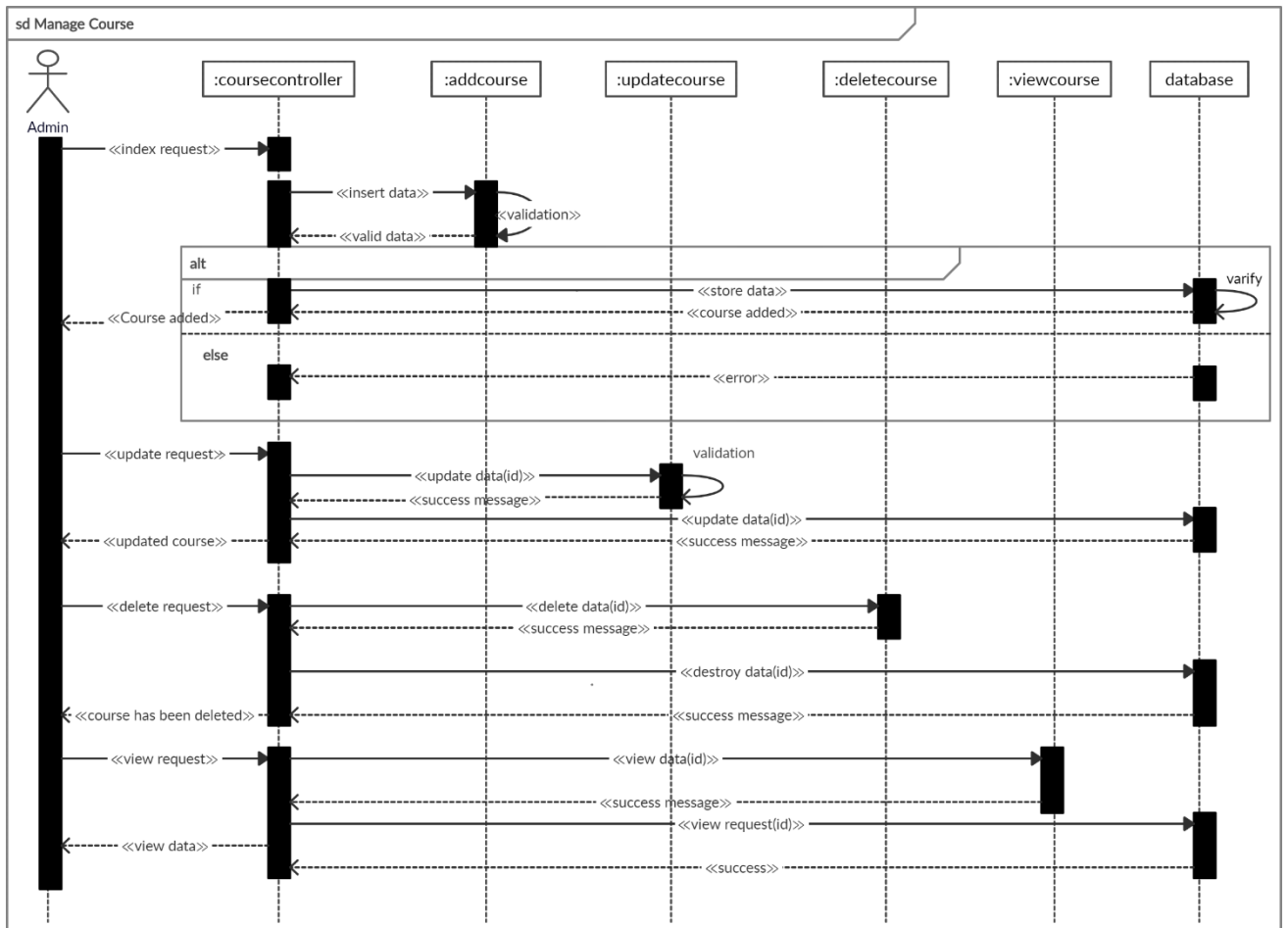


Figure 3.4.4: Admin Manage Courses Sequence Diagram

3.4.5 Admin Manage Course Teachers

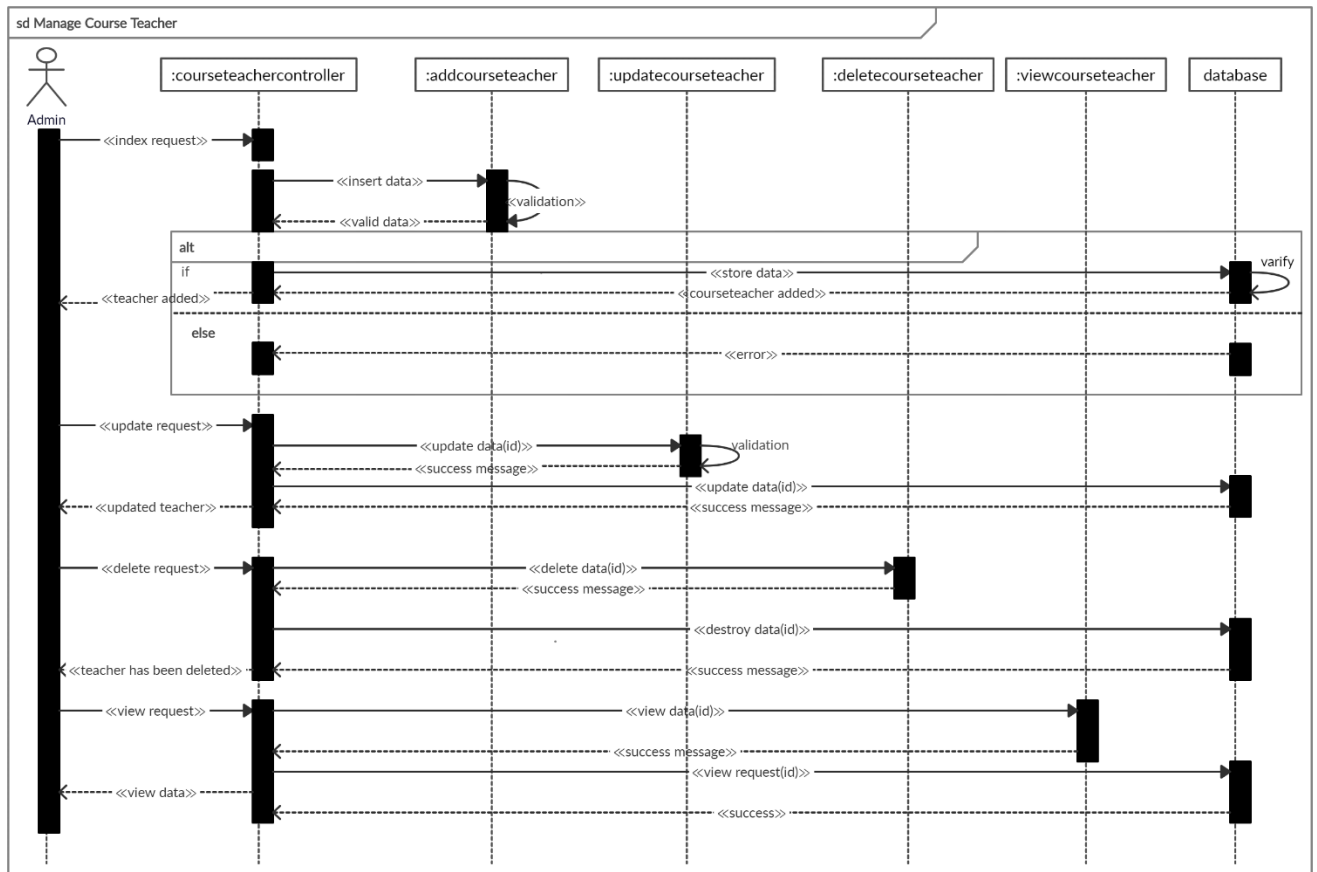


Figure 3.4.5: Admin Manage Course Teachers Sequence Diagram

3.4.6 Admin Manage Classes

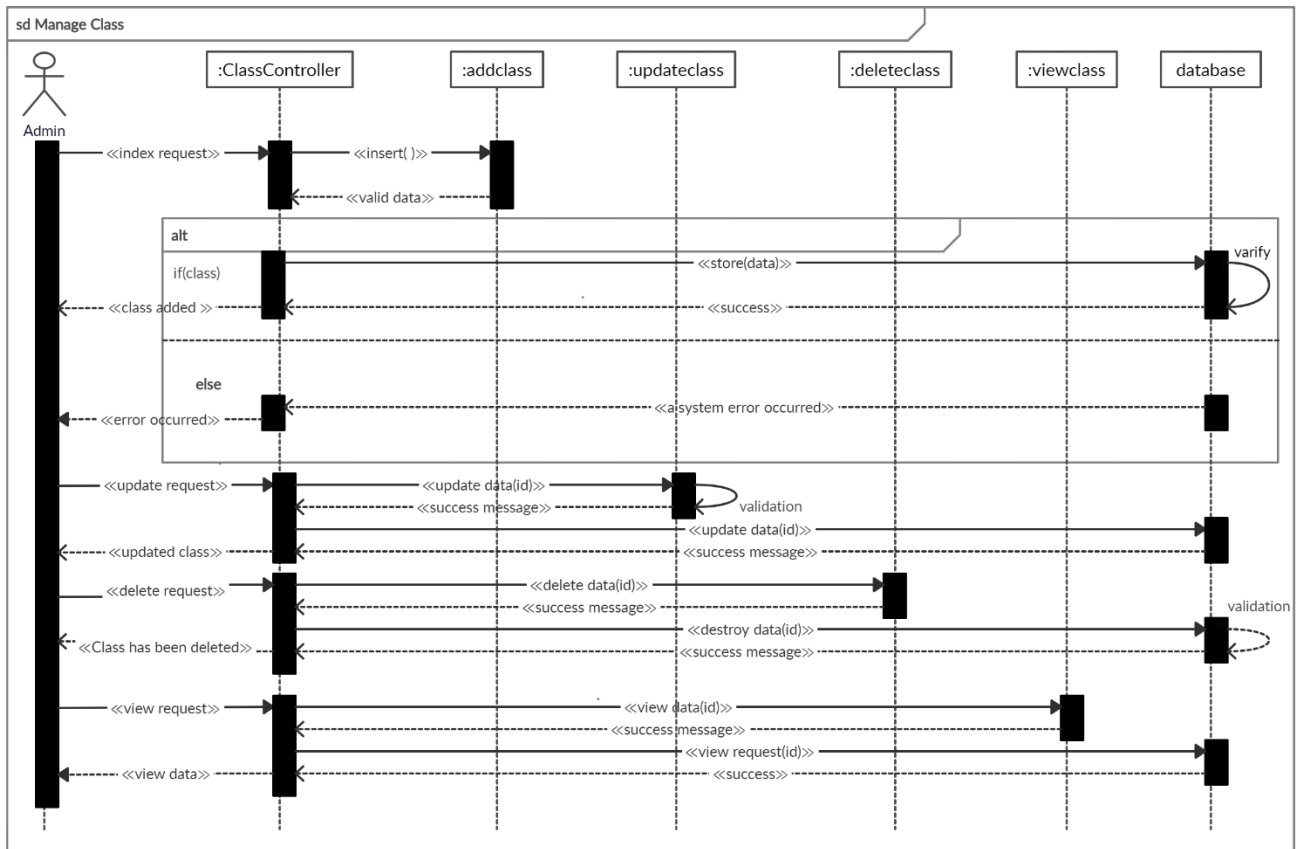


Figure 3.4.6: Admin Manage Classes Sequence Diagram

3.4.7 Admin Create Timetable

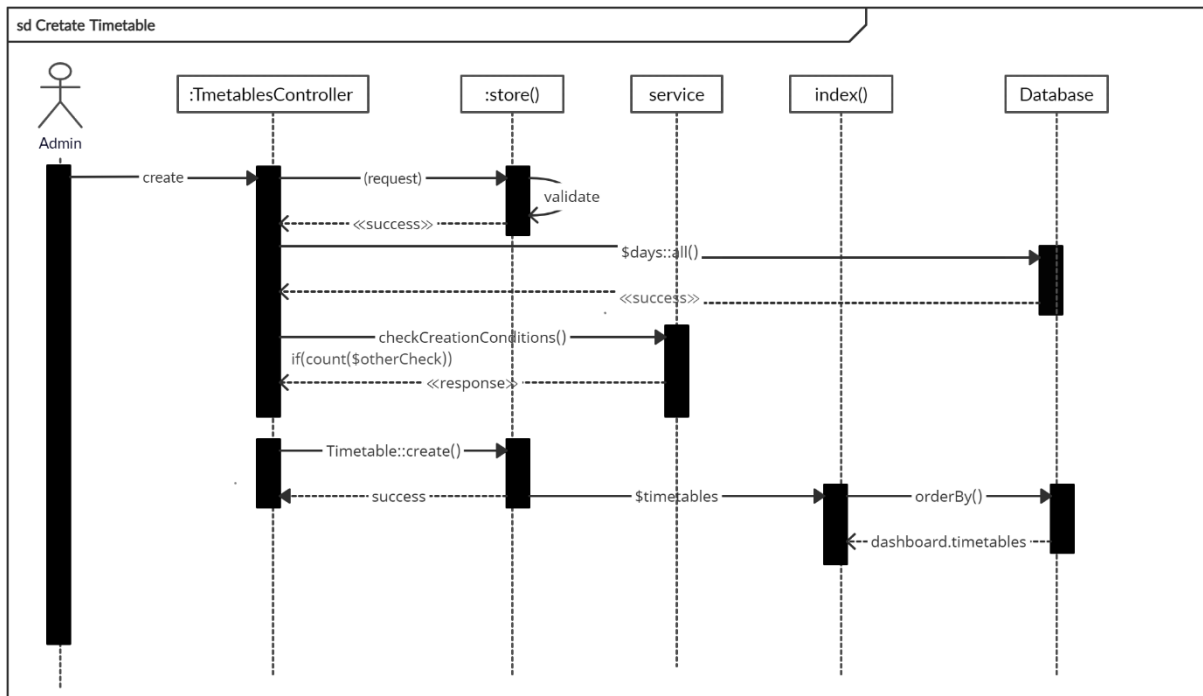


Figure 3.4.7: Admin Create Timetable Sequence Diagram

3.4.8 Student Registration

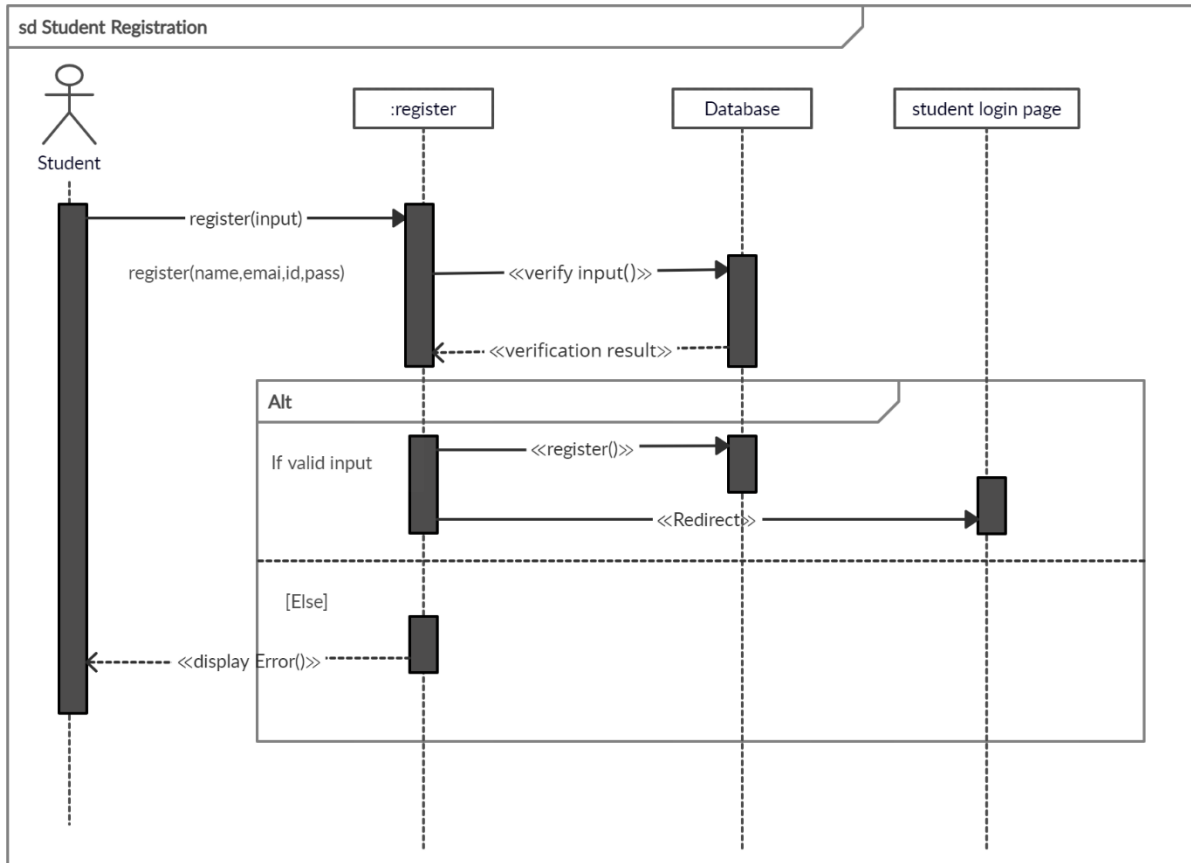


Figure 3.4.8: Student Registration Sequence Diagram

3.4.9 Student Login

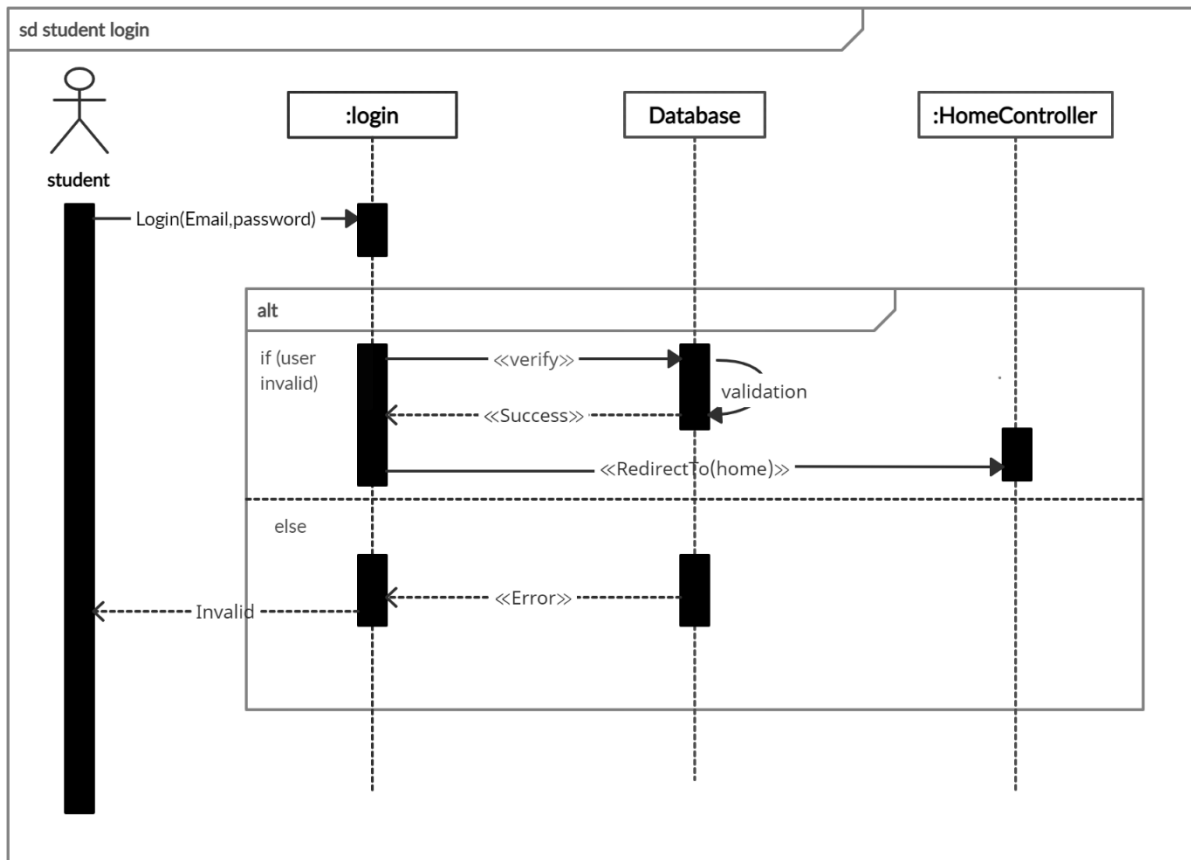


Figure 3.4.9: Student Login Sequence Diagram

3.4.10 View Timetable

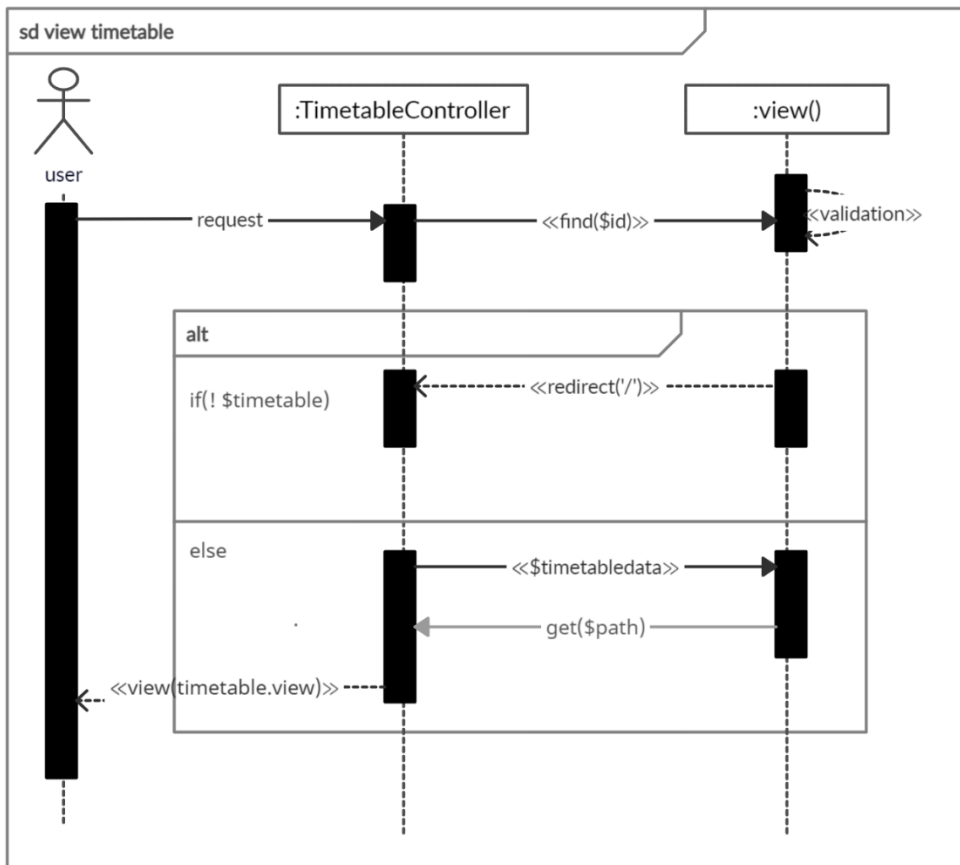


Figure 3.4.10: View Timetable Sequence Diagram

3.4.11 Print PDF

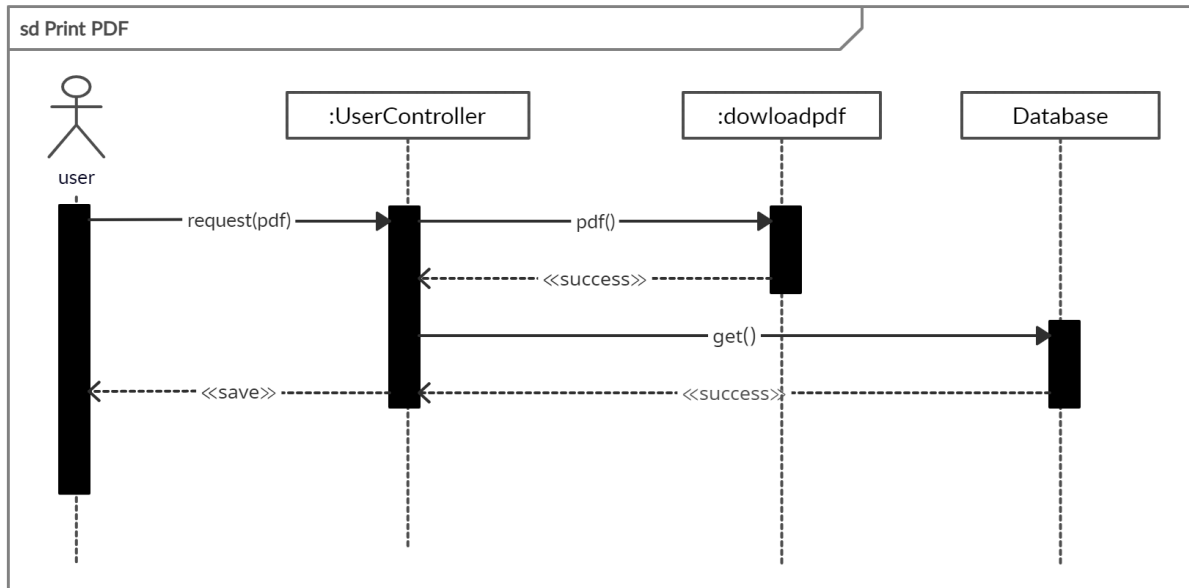


Figure 3.4.11: Print PDF Sequence Diagram

3.5 System Entity Relation Diagram

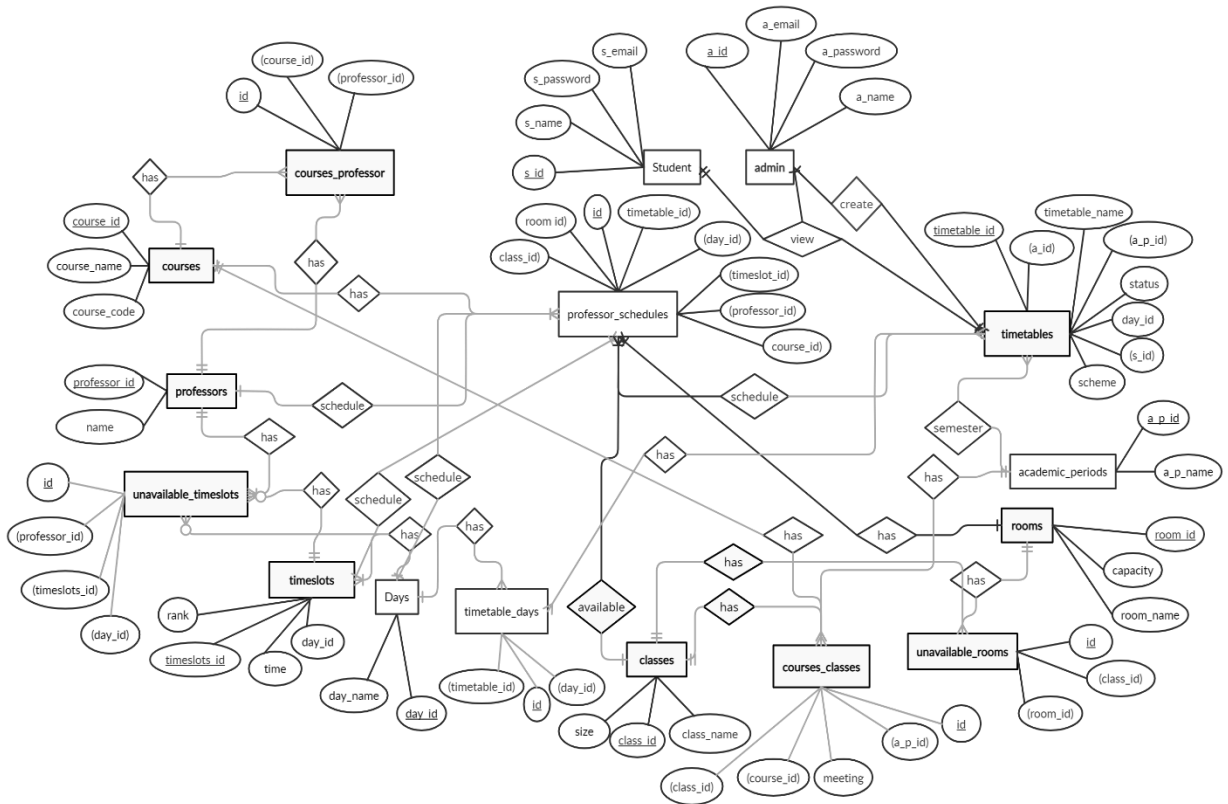


Figure 3.5: Entity Relation Diagram

3.7 Design and Implementation Constraints

3.7.1 Framework or Software Language:

‘Timetable Displayer’ is written by PHP using Laravel framework with a bootstrap template for user interface and data stored in MySQL database.

3.8 Development Tools and Technology:

- IDE: Atom
- Database: MySQL
- UI: HTML, CSS, Bootstrap, JavaScript
- Framework: Laravel 5.5.10
- Language: PHP
- Web-Server: XAMPP Server

3.9 Project Organization

3.9.1 Software Process Model

The first Process Model to be implemented was the Waterfall Model. Every process must be completed in a Waterfall model before the next phase can begin and there is no overlap between the phases. The model of the waterfall is the earliest SDLC technique used for software development. The whole process of software development is broken into different stages in "The Waterfall" strategy. The product of one stage serves sequentially as the reference for the next step. This suggests that every step in the phase of creation continues only if the previous stage is complete. The waterfall model is a sequential modeling process in which development through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance is seen as continuously flowing downward (like a waterfall).

As the Waterfall Model depicts the process of software creation in a linear sequential flow, it is also often referred to as a model of the linear sequential life cycle.

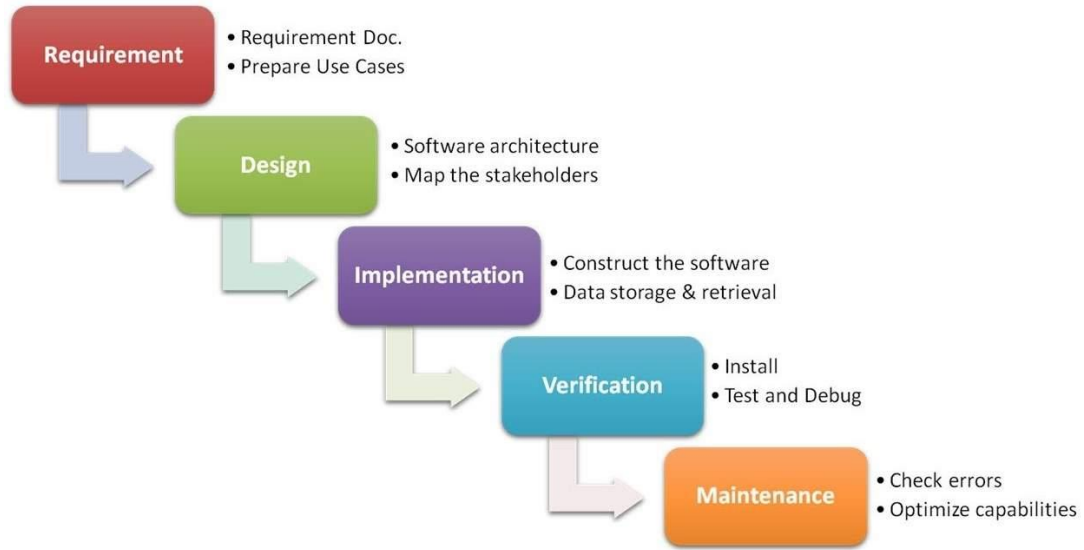


Figure3.9.1: Waterfall Model

CHAPTER 4

SYSTEM TESTING

4.1 Testing Features

Testing is classified as an operation to verify that the real results conform to the predicted results and to guarantee that the device is free of defects. It is possible to view checking features as an adjustment to add additional functionality or change an existing project.

4.1.1 Features to be tested

There are some features below:

- Login
- Add Teacher

4.2 Testing Strategies

Test methods decide how to handle each aspect of the application. It is a series of full instructions for the nature of the test and for each stage of the case. It specifies what sort of methodology should be implemented and which module should be tested.

4.2.1 Test Approach

In every software project, testing is one of the essential things. It proves the efficiency of software. It also helps find glitches and errors in applications. So the program seems to the consumer to be very basic and error-free. I have tested and validated every function one by one without any automation tool.

4.2.2 Test Category

- Integration Testing
- Module Testing

4.2.3 Success/Failed Criteria

The criterion for pass / fail test will be determined by the test engineers. It all depends on how well all of the specifications perform. When the test results are fine, a pass or a loss can be considered. In any case, I have the 100 per cent pass requirements. If, during the evaluation, a function does not operate correctly, it will be deemed a mistake.

4.3 Testing Environment

There are some key area to set up for testing

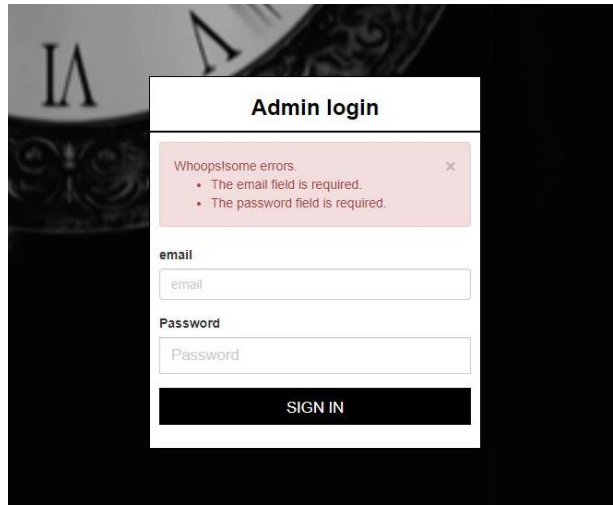
- Operating System
- Browser
- System
- Application
- Database server
- Test data
- Network

4.4 Test Cases

Testing Case No-1 (Integration Testing)

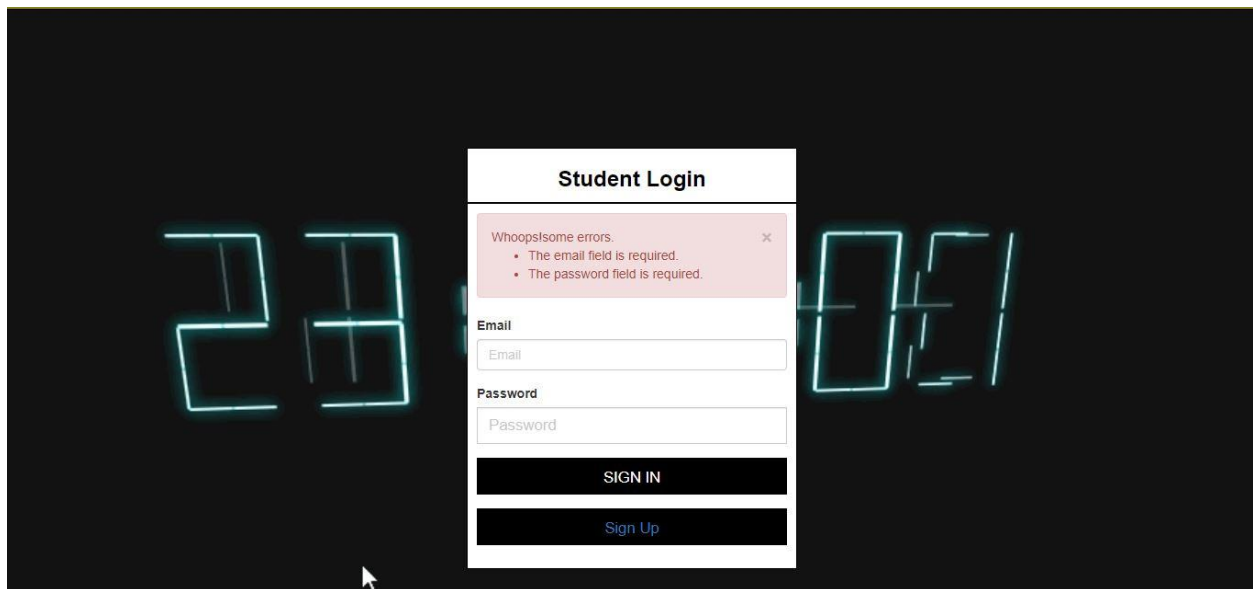
Test Case ID.1	Module name: Admin & Student Login
Test Priority: High	Test Date:22.09.2020
Test Title: Admin & Student Login Verification with valid email & password	Test executed by: Kanak Kumar Sarker
Description: Test Admin & Student Login Page	Test executed date: 22.09.2020
Pre-condition:	Users must have valid email and password.
Test steps:	1. Go to login page 2. Provide valid email & password 3. Click Sign in button
Test Data:	Admin: Email: admin@gmail.com Password: admin Student: Email: kanak@gmail.com Password: 123456
Expected Results:	User should able to login
Actual Result:	User logged in successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully Logged in.

Table 4.4.1: Testing Case (Admin & Student Login)



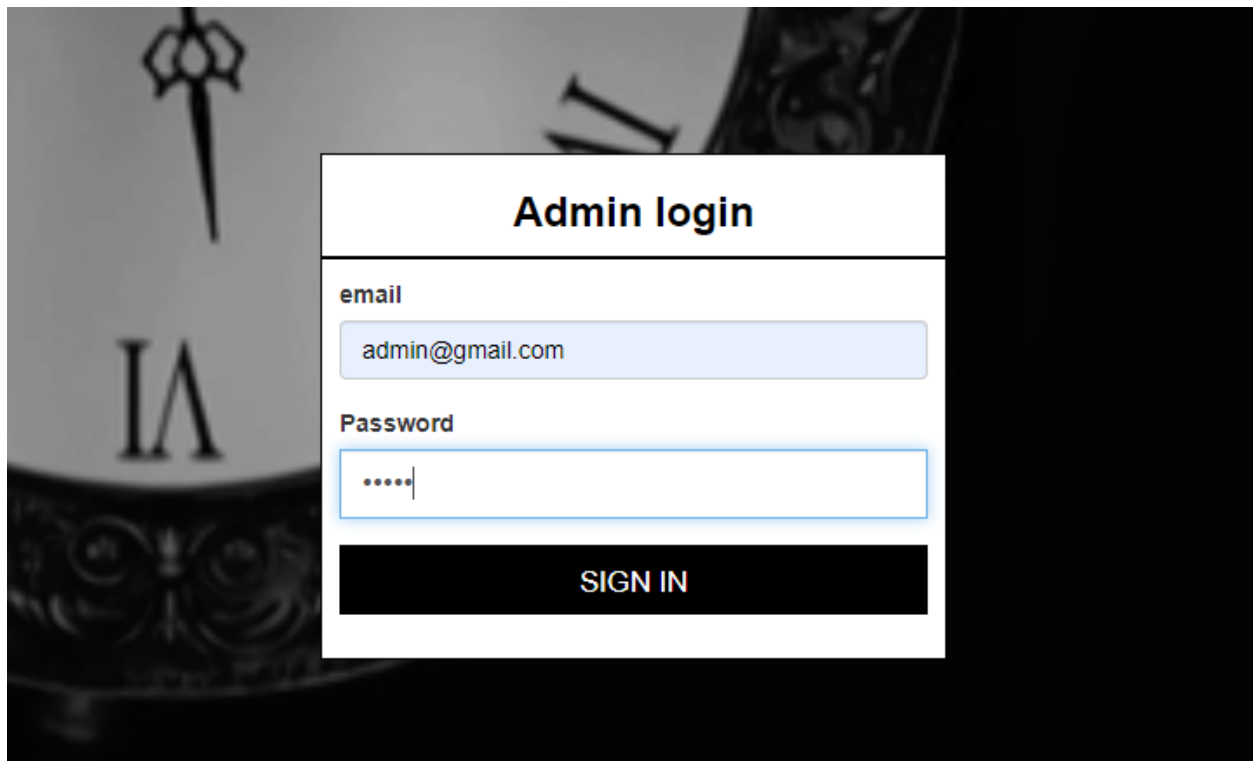
The image shows a web form titled "Admin login" centered on a dark background with a faint clock face. The form has a white border and contains the following elements: a red error message box at the top stating "Whoops! some errors." with two bullet points: "The email field is required." and "The password field is required."; an "email" label above a text input field containing the placeholder "email"; a "Password" label above a text input field containing the placeholder "Password"; and a black "SIGN IN" button at the bottom.

Figure 4.4.1.1: Blank Admin Login Attempt



The image shows a web form titled "Student Login" centered on a dark background with a faint digital clock showing "23:05". The form has a white border and contains the following elements: a red error message box at the top stating "Whoops! some errors." with two bullet points: "The email field is required." and "The password field is required."; an "Email" label above a text input field containing the placeholder "Email"; a "Password" label above a text input field containing the placeholder "Password"; a black "SIGN IN" button; and a blue "Sign Up" button at the bottom.

Figure 4.4.1.2: Blank Student Login Attempt



The image shows a login form titled "Admin login" centered on a dark background with a faint, ornate pattern. The form is white with a thin black border. It contains two input fields: "email" with the text "admin@gmail.com" and "Password" with masked characters ".....". Below the fields is a black button with the text "SIGN IN" in white capital letters.

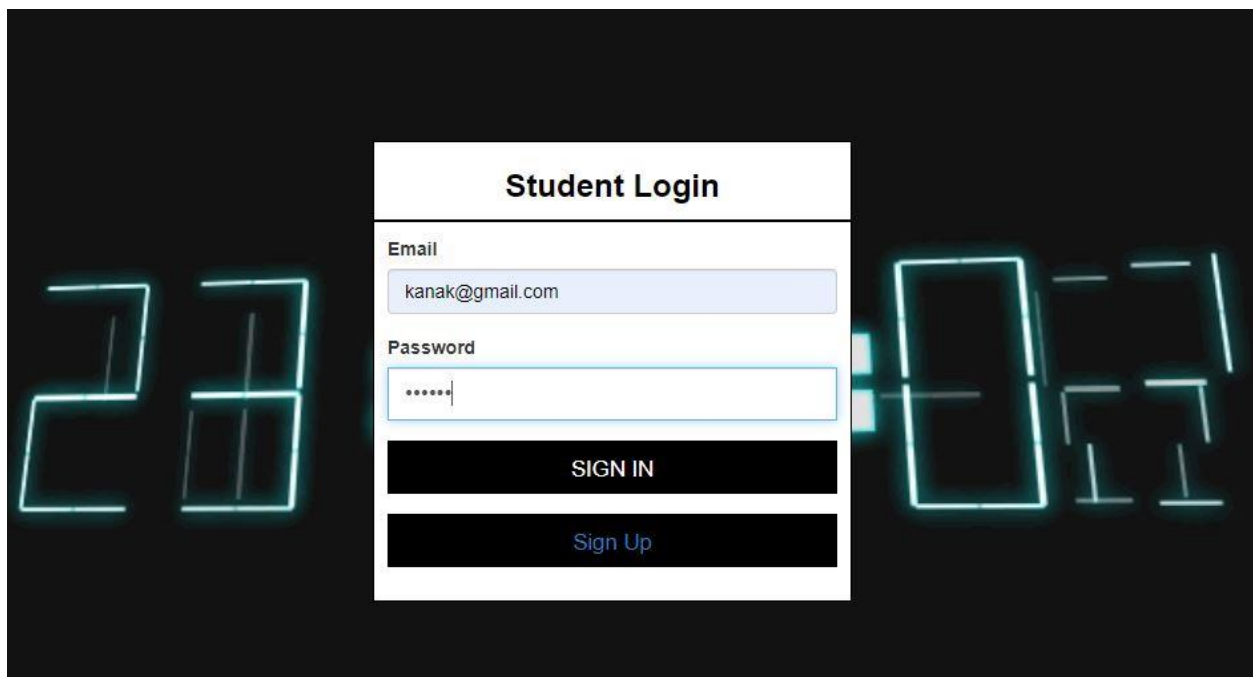
Admin login

email
admin@gmail.com

Password
.....

SIGN IN

Figure 4.4.1.3: Admin Login Attempt with Correct Data



The image shows a login form titled "Student Login" centered on a dark background with a faint, glowing neon pattern. The form is white with a thin black border. It contains two input fields: "Email" with the text "kanak@gmail.com" and "Password" with masked characters ".....". Below the fields are two buttons: a black button with the text "SIGN IN" in white capital letters, and a black button with the text "Sign Up" in blue capital letters.

Student Login

Email
kanak@gmail.com

Password
.....

SIGN IN

Sign Up

Figure 4.4.1.4: Student Login Attempt with Correct Data

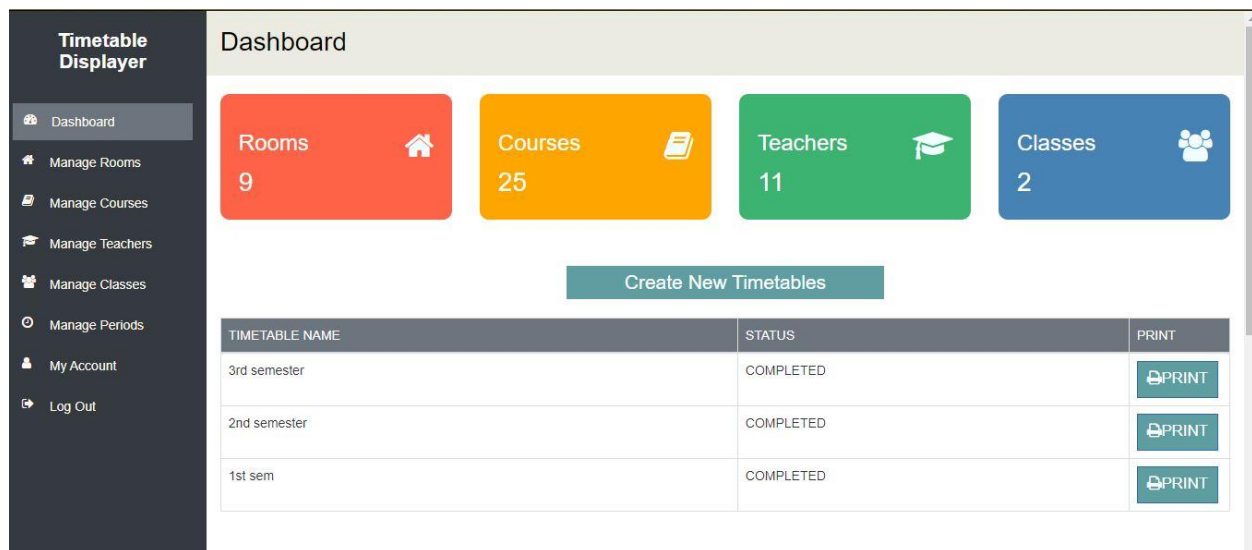


Figure 4.4.1.5: Admin Login Successful with Authentication

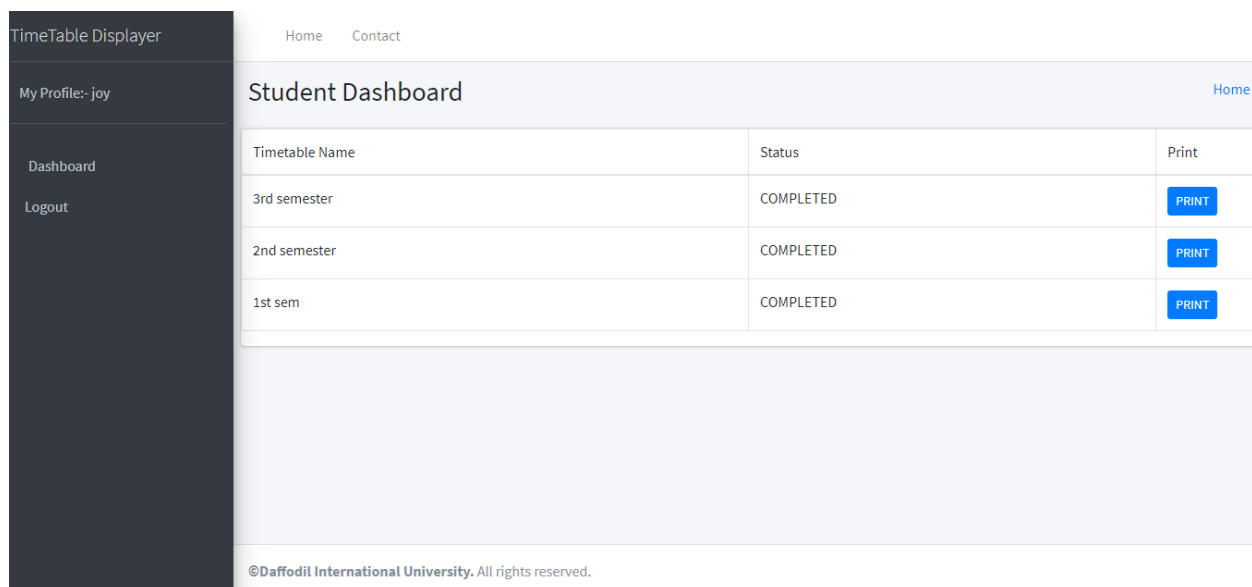


Figure 4.4.1.6: Student Login Successful with Authentication

Testing Case No-2 (Module Testing)

Test Case ID.2	Module name: Add Teacher
Test Priority: High	Test Date:11.10.2020
Test Title: Add teacher with validation	Test executed by: Kanak Kumar Sarker
Description: Admin can add teacher	Test executed date: 11.10.2020
Pre-condition:	Admin must login and insert valid data
Test steps:	1. After login go to manage teachers 2. Click add teacher 3. Insert data all the field 4. Click Submit button
Test Data:	Name: Ms. Fatama Binta Rafiq Email: fatama.swe@diu.edu.bd Course : SWE 304 Programming Languages SWE 305 System Analysis SWE 314 Theory of Computing Unavailable periods: Saturday 8:00-10:30 Saturday 12:00-13:30
Expected Results:	User add Successfully
Actual Result:	User add Successfully
Status (Pass/Fail):	Pass

Table 4.4.2: Testing Case (Add Teacher)

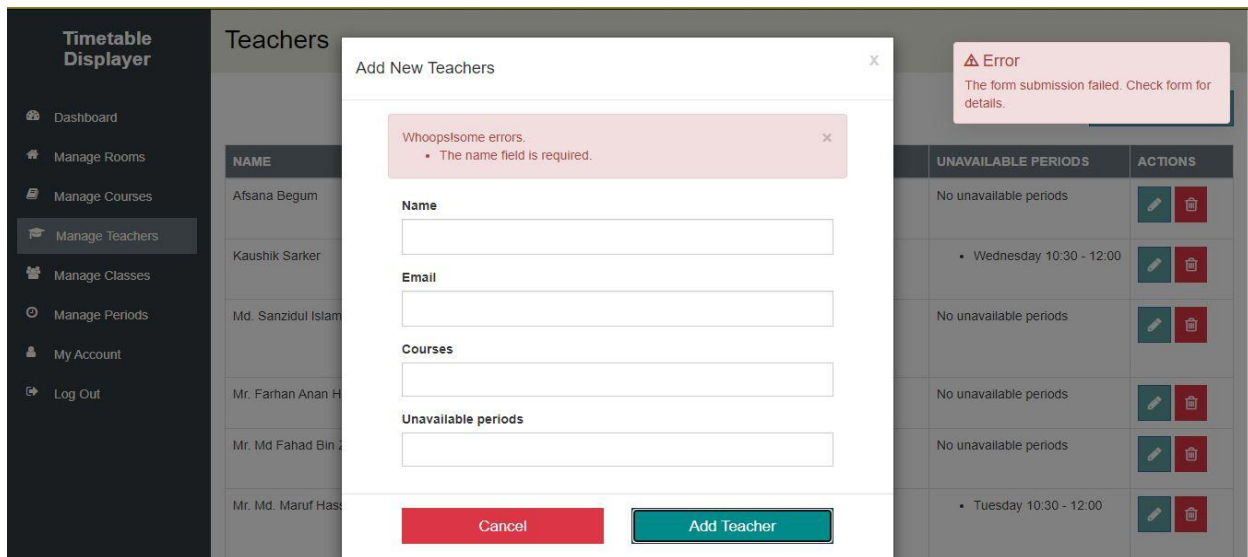


Figure 4.4.2.1: Blank filled submit

Manage Teachers			SWE 312 System Analysis and Design	
Manage Classes	Kaushik Sarker		- SWE 307 Operations Research I - SWE 405 computer fundamental	Wednesday 10:30 - 12:00
Manage Periods	Md. Sanzidul Islam		- SWE 306 Artificial Intelligence - SWE 404 Mathematics-II - SWE 411 Robotics	No unavailable periods
My Account	Mr. Farhan Anan Himu		SWE 415 Software Engineering II	No unavailable periods
Log Out	Mr. Md Fahad Bin Zamal		- SWE 401 Computer Networks - SWE 413 Data Communication	No unavailable periods
	Mr. Md. Maruf Hassan		- SWE 308 Web Development - SWE 402 Computer Security - SWE 412 Software Security	Tuesday 10:30 - 12:00
	Ms. Fatama Binta Rafiq		- SWE 304 Programming Languages - SWE 305 System Analysis - SWE 406 Software Engineering I - SWE 314 Theory of Computing	- Saturday 8:00 - 10:30 - Saturday 12:00 - 13:30
	Ms. Lamisha Rawshan		- SWE 313 C Programming - SWE 403 Information Systems	No unavailable periods
	Ms. Nusrat Jahan		SWE 301 Computer Architecture	No unavailable periods

Figure 4.4.2.2: Teacher Data Insert Successfully

CHAPTER 5

USER MANUAL

5.1 Admin Login Page

The admin will login with their email and password on this tab.

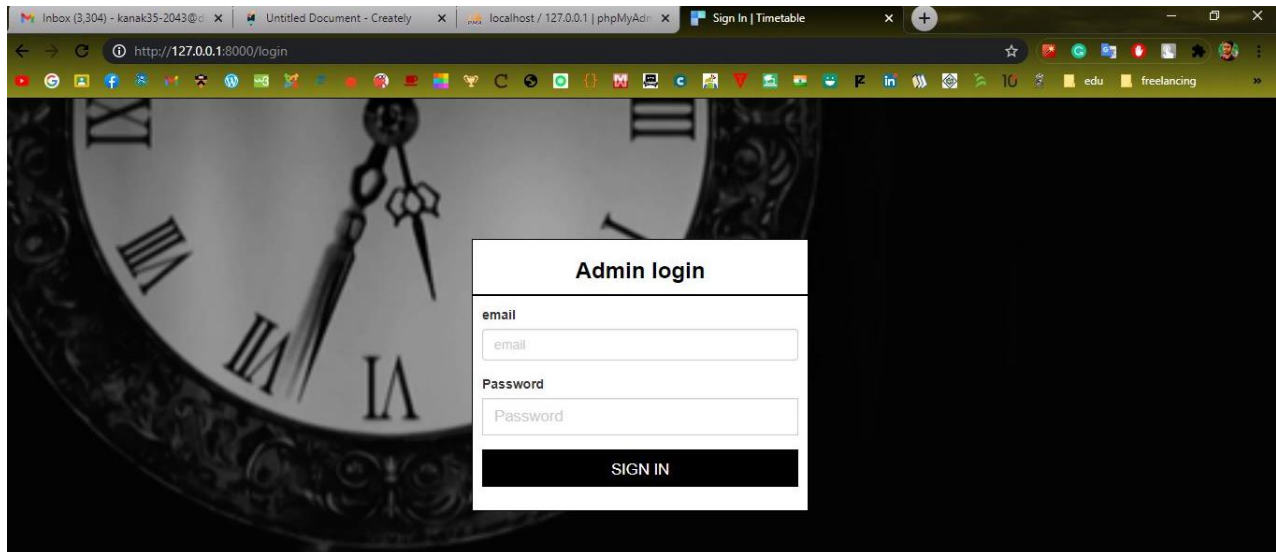


Figure 5.1 UI (Admin Login Page)

5.2 Admin Dashboard

Users can view all aspects of the menu & dashboard with data.

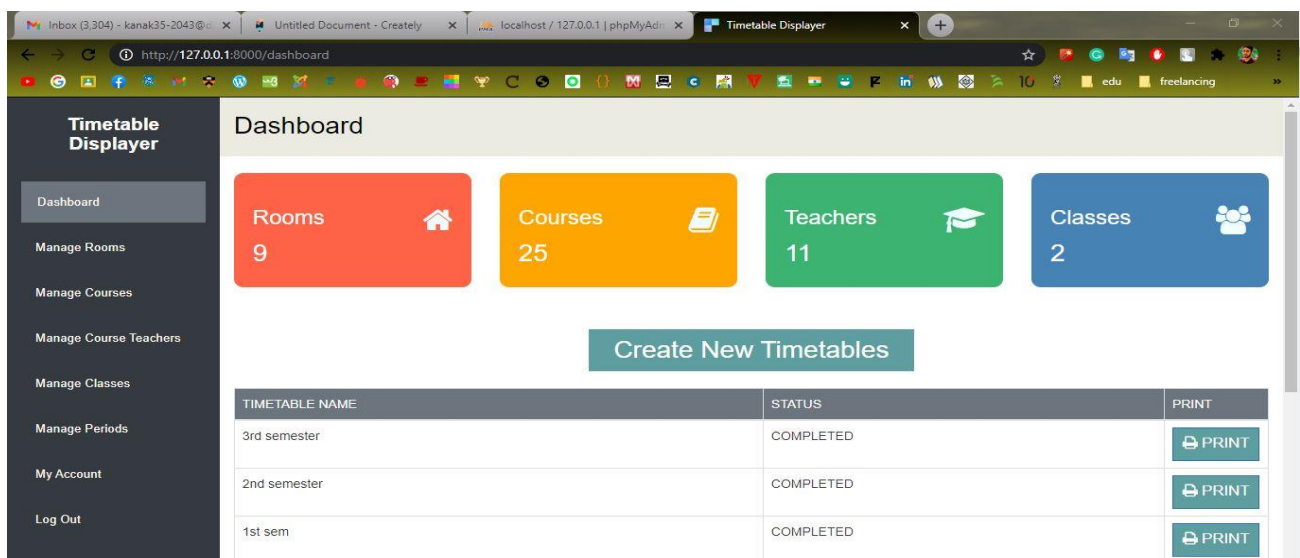


Figure 5.2 UI (Admin Dashboard Page)

5.3 Admin Manage Periods

5.3.1 View Periods

Only the admin can access a list of periods.

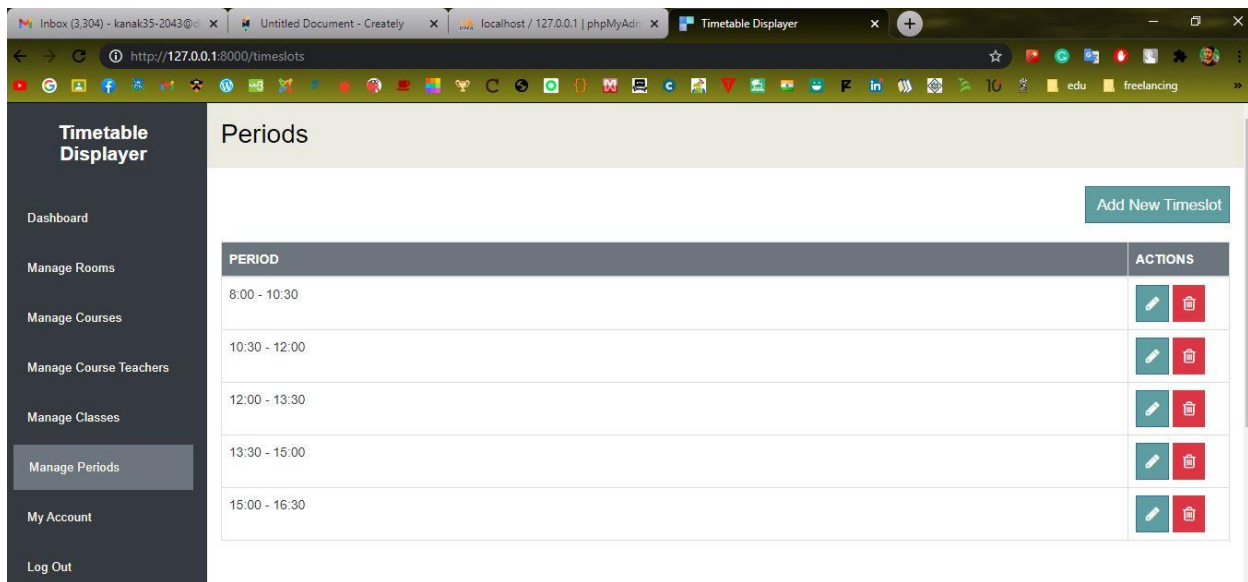


Figure 5.3.1 UI (View Periods Page)

5.3.2 Add New Timeslots

Only admin can add new timeslots.

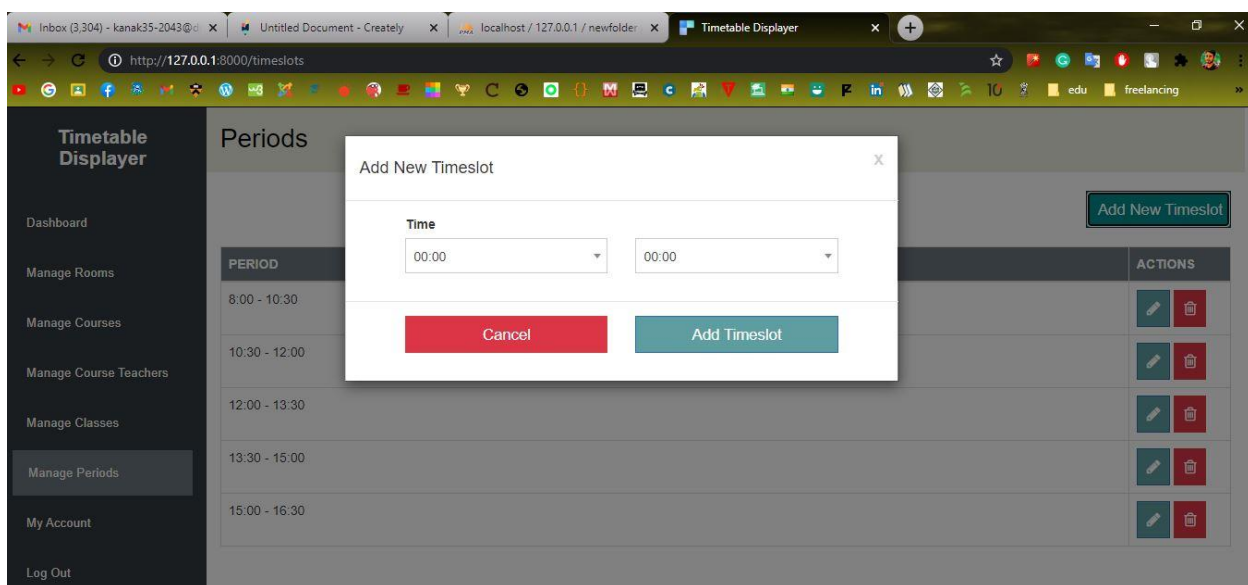


Figure 5.3.2 UI (Add New Timeslots Page)

5.3.3 Update Timeslot

Only Admin can update timeslot.

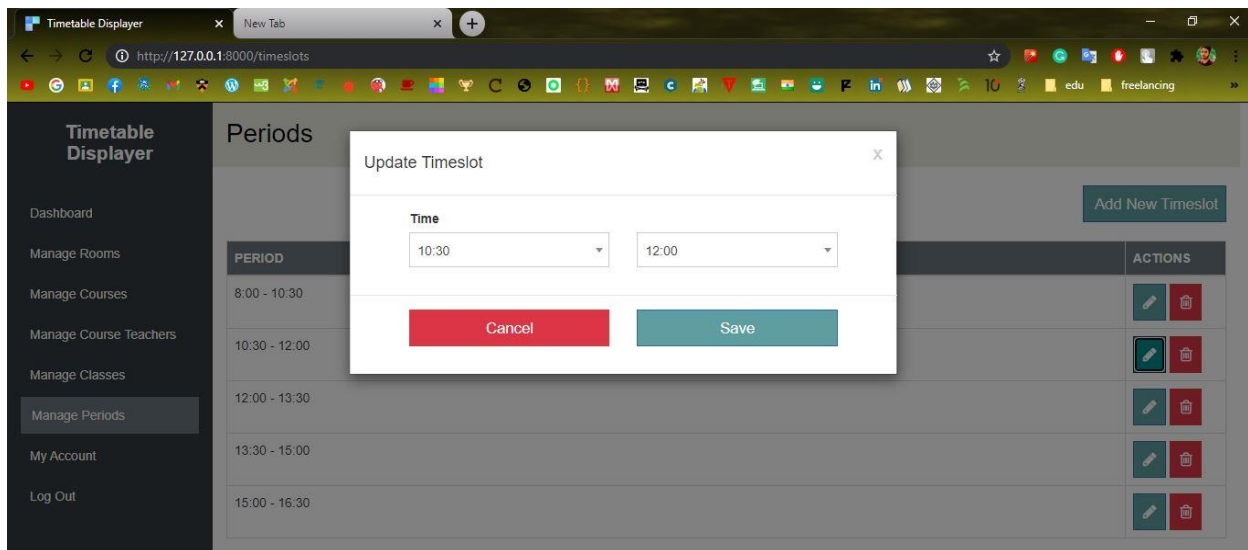


Figure 5.3.3 UI (Update Timeslot Page)

5.3.4 Delete Timeslots

Only admin can delete timeslots.

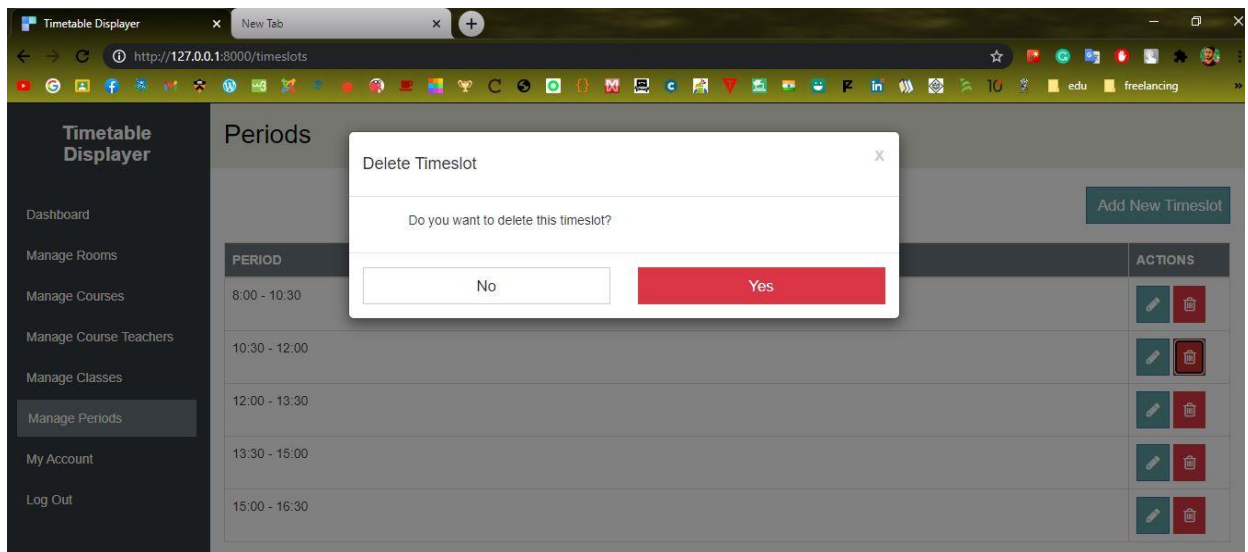
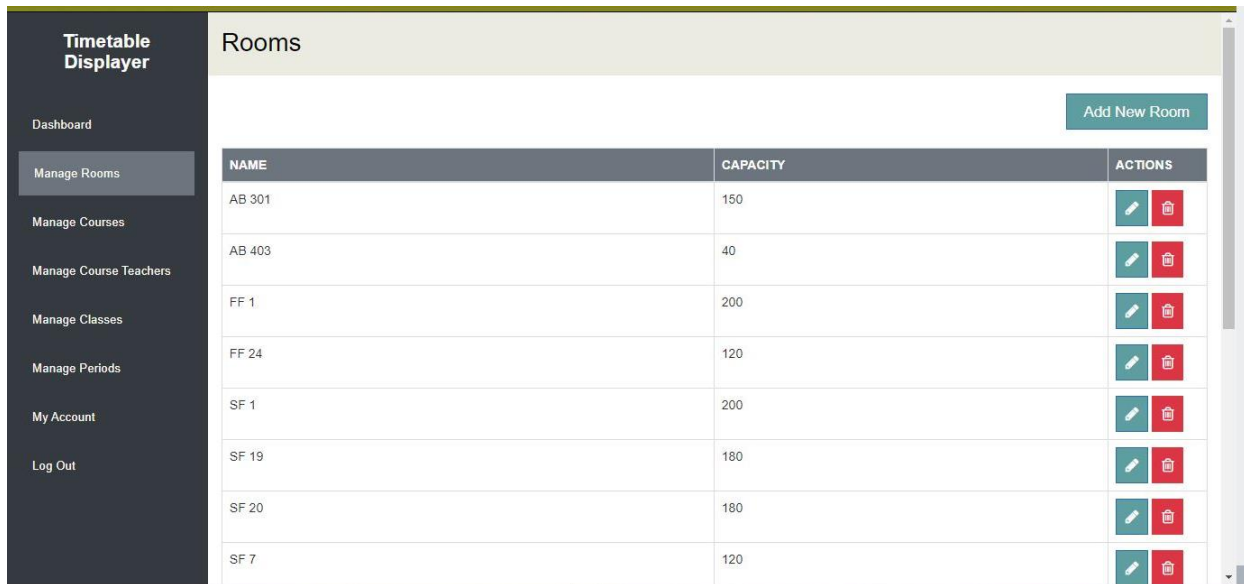


Figure 5.3.4 UI (Delete Timeslots Page)

5.4 Admin Manage Rooms

5.4.1 View Rooms

Only the admin can access a list of rooms with info.

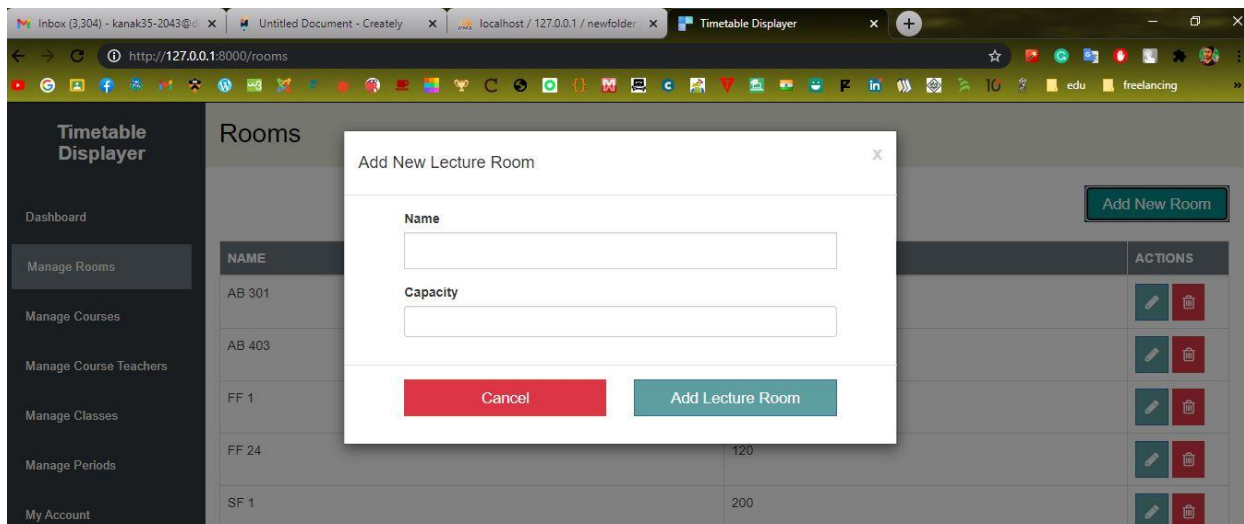


Rooms		
NAME	CAPACITY	ACTIONS
AB 301	150	 
AB 403	40	 
FF 1	200	 
FF 24	120	 
SF 1	200	 
SF 19	180	 
SF 20	180	 
SF 7	120	 

Figure 5.4.1 UI (View Rooms Page)

5.4.2 Add Room

Only admin can add new room.



Add New Lecture Room

Name

Capacity

Figure 5.4.2 UI (Add Room Page)

5.4.3 Update Rooms

Only admin can update rooms.

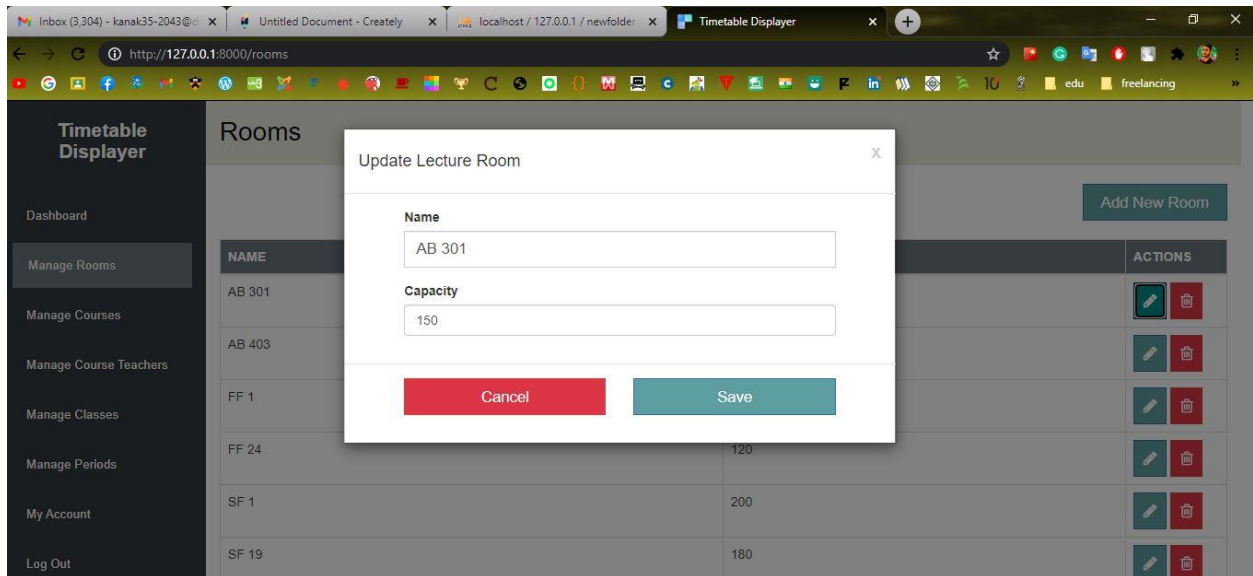


Figure 5.4.3 UI (Update Room Page)

5.4.4 Delete Room

Only admin can delete rooms.

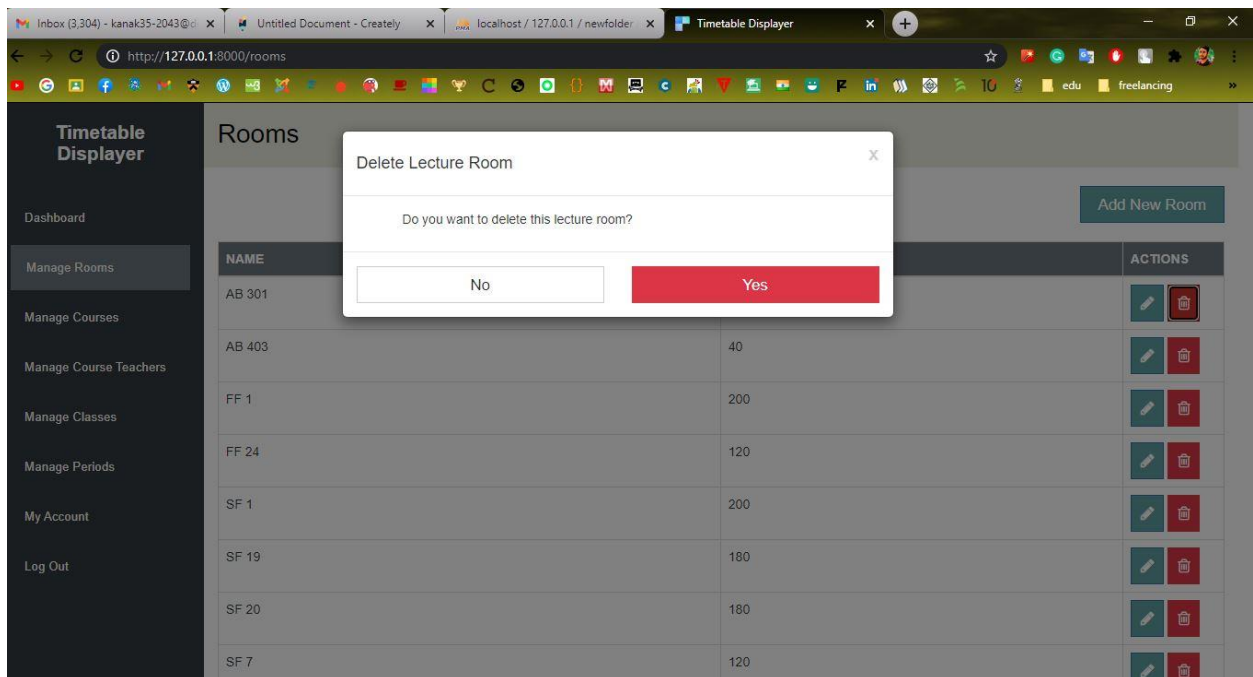
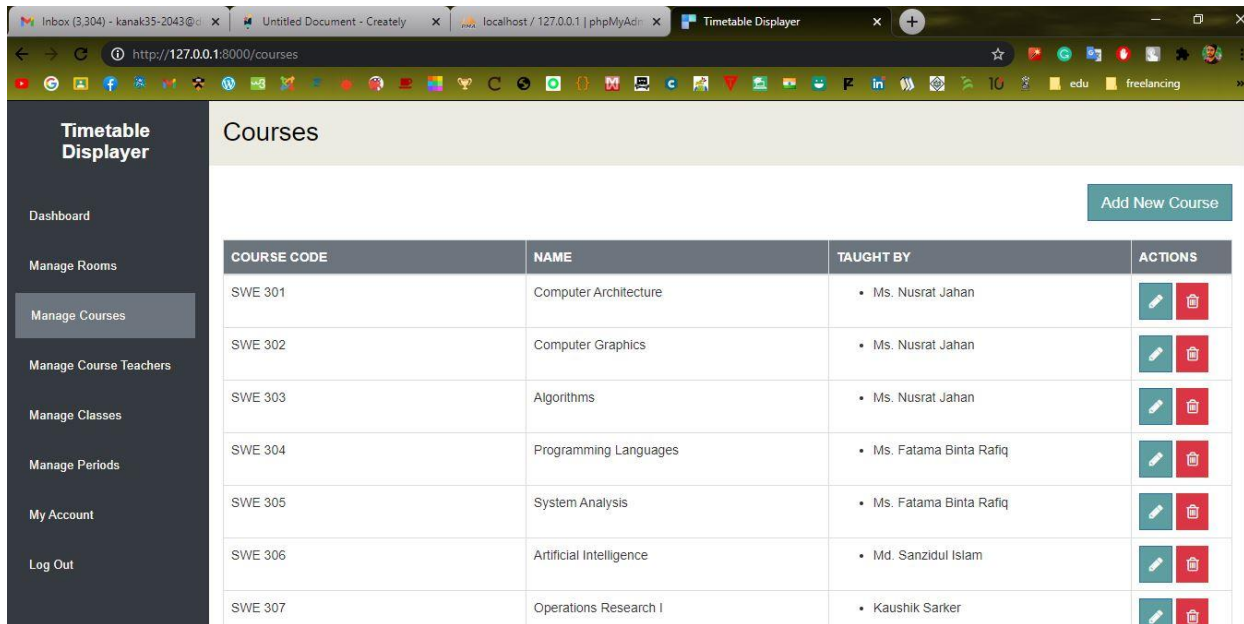


Figure 5.4.4 UI (Delete Room Page)

5.5 Admin Manage Courses

5.5.1 View Courses

Only the admin can access a list of Courses with info.

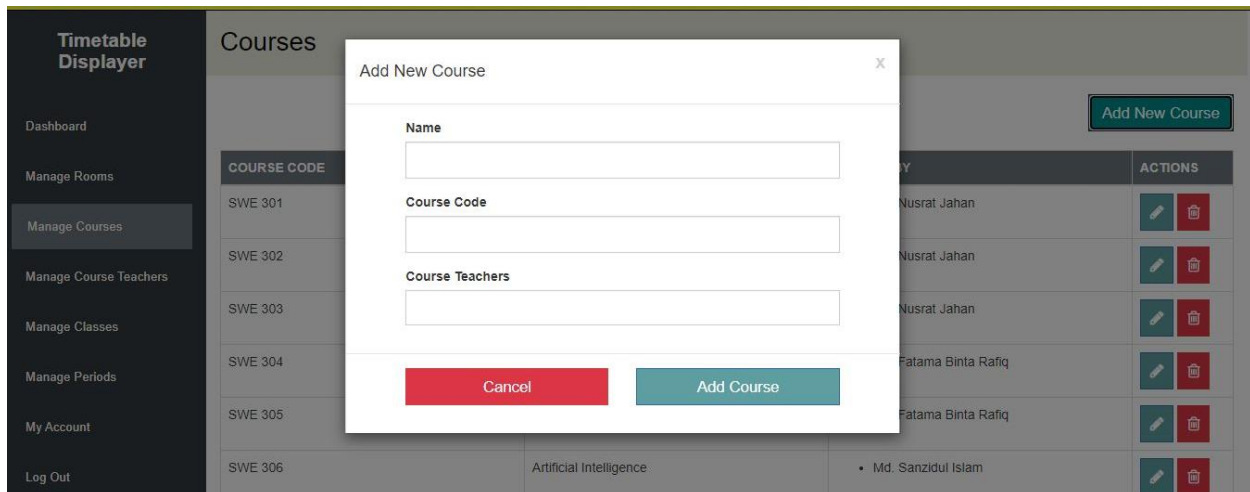


COURSE CODE	NAME	TAUGHT BY	ACTIONS
SWE 301	Computer Architecture	• Ms. Nusrat Jahan	 
SWE 302	Computer Graphics	• Ms. Nusrat Jahan	 
SWE 303	Algorithms	• Ms. Nusrat Jahan	 
SWE 304	Programming Languages	• Ms. Fatama Bintu Rafiq	 
SWE 305	System Analysis	• Ms. Fatama Bintu Rafiq	 
SWE 306	Artificial Intelligence	• Md. Sanzidul Islam	 
SWE 307	Operations Research I	• Kaushik Sarker	 

Figure 5.5.1 UI (View Courses Page)

5.5.2 Add Course

Only admin can add courses.



Add New Course

Name

Course Code

Course Teachers

Figure 5.5.2 UI (Add Course Page)

5.5.3 Update Course

Only admin can update course.

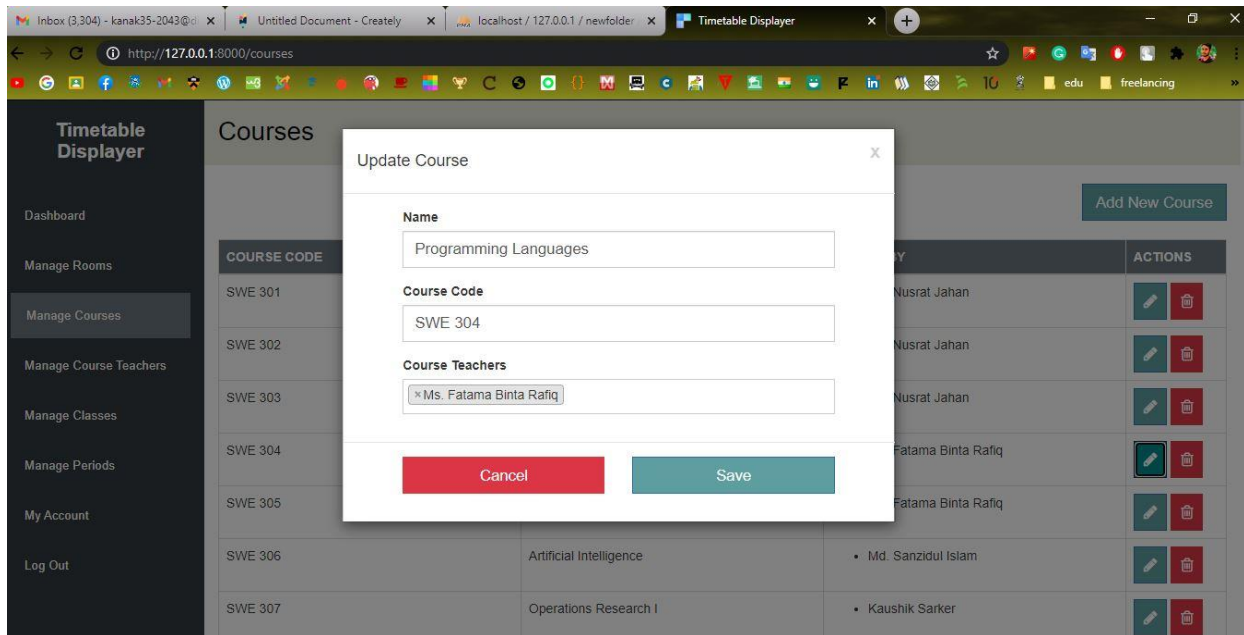


Figure 5.5.3 UI (Update Course Page)

5.5.4 Delete Course

Only admin can delete courses.

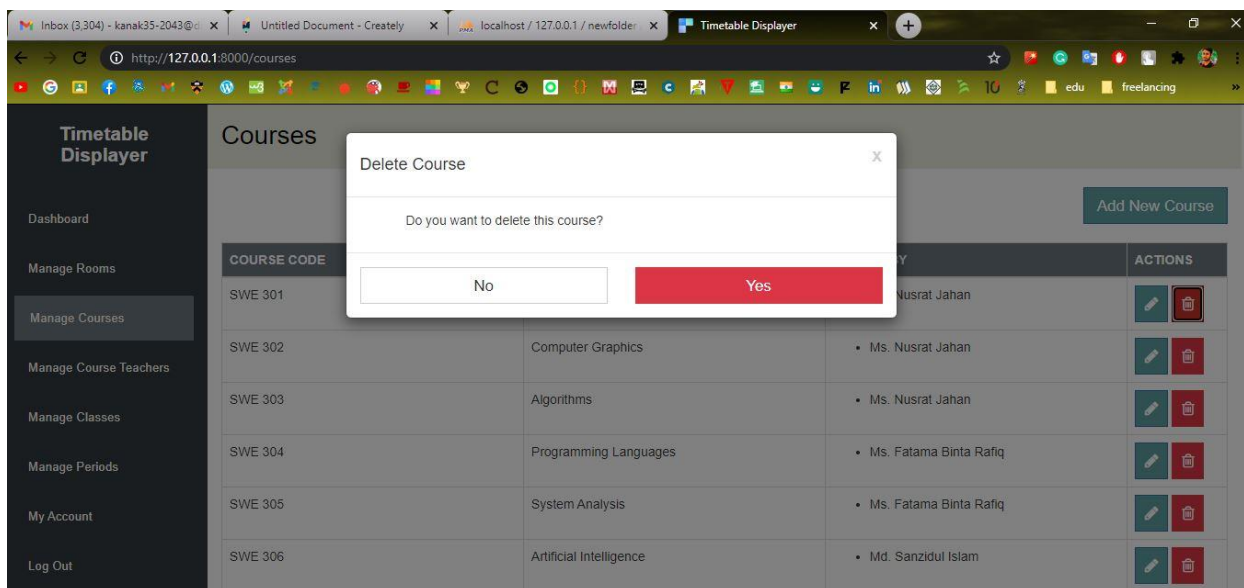


Figure 5.5.4 UI (Delete Course Page)

5.6 Admin Manage Teachers

5.6.1 View Teachers

Only the admin can access a list of teachers with info

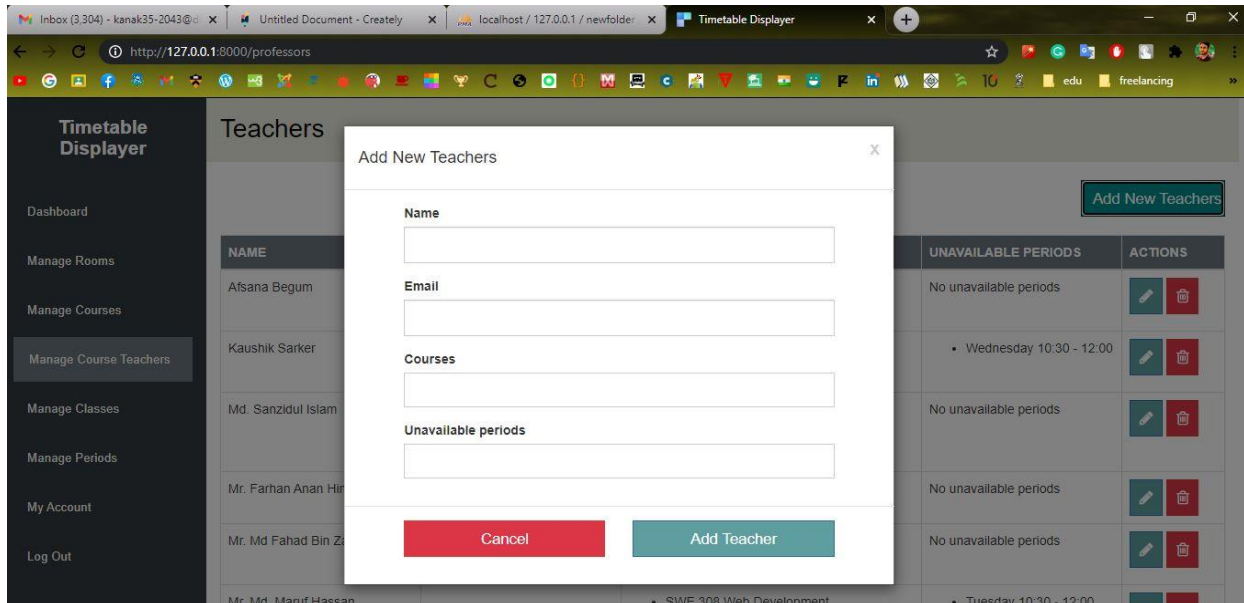


NAME	EMAIL	COURSES TAUGHT	UNAVAILABLE PERIODS	ACTIONS
Afsana Begum		- SWE 311 Data Structures - SWE 312 System Analysis and Design	No unavailable periods	 
Kaushik Sarker		- SWE 307 Operations Research I - SWE 405 computer fundamental	Wednesday 10:30 - 12:00	 
Md. Sanzidul Islam		- SWE 306 Artificial Intelligence - SWE 404 Mathematics-II - SWE 411 Robotics	No unavailable periods	 
Mr. Farhan Anan Himu		SWE 415 Software Engineering II	No unavailable periods	 
Mr. Md Fahad Bin Zamal		- SWE 401 Computer Networks - SWE 413 Data Communication	No unavailable periods	 
Mr. Md. Maruf Hassan		SWE 308 Web Development	Tuesday 10:30 - 12:00	 

Figure 5.6.1 UI (View Teachers Page)

5.6.2 Add Teacher

Only admin can add teacher.



Add New Teachers

Name

Email

Courses

Unavailable periods

Figure 5.6.2 UI (Add Teacher Page)

5.6.3 Update Teacher

Only admin can update teacher.

The screenshot shows the 'Update Teachers' modal form. The form has the following fields:

- Name:** A text input field containing 'Afsana Begum'.
- Email:** An empty text input field.
- Courses:** A multi-select dropdown menu showing 'SWE 311 Data Structures' and 'SWE 312 System Analysis and Design'.
- Unavailable periods:** An empty text input field.

At the bottom of the modal are two buttons: 'Cancel' (red) and 'Add Teacher' (teal). The background shows a table of teachers with columns for NAME, COURSES, UNAVAILABLE PERIODS, and ACTIONS.

Figure 5.6.3 UI (Update Teacher Page)

5.6.4 Delete Teacher

Only admin can delete teacher.

The screenshot shows the 'Delete Teacher' modal form. The form has the following fields:

- Message:** A text input field containing 'Do you want to delete this Teacher?'.

At the bottom of the modal are two buttons: 'No' (white) and 'Yes' (red). The background shows a table of teachers with columns for NAME, COURSES, UNAVAILABLE PERIODS, and ACTIONS.

Figure 5.6.4 UI (Delete Teacher Page)

5.7.1 View Classes

Timetable
Displayr

Dashboard
Manage Rooms
Manage Courses
Manage Course Teachers
Manage Classes
Manage Periods
My Account
Log Out

Classes

Add New Class

NAME	SIZE	COURSES	UNAVAILABLE ROOMS	ACTIONS
SEC A	180	Semester I - SWE 304 Programming Languages - SWE 305 System Analysis - SWE 306 Artificial Intelligence - SWE 307 Operations Research I - SWE 308 Web Development Semester II - SWE 311 Data Structures - SWE 312 System Analysis and Design - SWE 313 C Programming - SWE 314 Theory of Computing - SWE 315 Mathematics-I Semester III - SWE 301 Computer Architecture - SWE 302 Computer Graphics - SWE 303 Algorithms - SWE 404 Mathematics-II - SWE 414 Physics	None specified	

5.7.2 Add Class

The screenshot shows a web browser window with the URL `http://127.0.0.1:8000/classes`. The application is titled "Timetable Displayer". On the left, a dark sidebar contains navigation links: "Dashboard", "Manage Rooms", "Manage Courses", "Manage Course Teachers", "Manage Classes" (highlighted), "Manage Periods", "My Account", and "Log Out". The main content area displays the "Add New class" modal form. The form has a title bar with a close button (X). It contains the following sections:

- Name:** A single text input field.
- Courses +:** A section with three columns:
 - Course:** Three dropdown menus, each with the text "Select a course".
 - Semester:** Three dropdown menus, each with the text "Select an Semester".
 - Class Per Week:** Three text input fields, each followed by a red "X" icon.
- People:** A single text input field.
- Unavailable Rooms:** A single text input field.

At the bottom of the modal, there are two buttons: a red "Cancel" button and a teal "Add class" button. In the background, a sidebar on the right shows "ROOMS" and "ACTIONS" sections, with an "Add New Class" button at the top.

© Daffodil International University

5.7.3 Update Class

Only admin can update class.

The screenshot shows the 'Update class' form in the Timetable Displayer application. The form has a sidebar on the left with navigation links: Dashboard, Manage Rooms, Manage Courses, Manage Course Teachers, Manage Classes (highlighted), Manage Periods, My Account, and Log Out. The main content area is titled 'Update class' and contains a form with the following fields:

- Name:** A text input field containing 'SEC A'.
- Courses +:** A table with three columns: Course, Semester, and Class Per Week. Each row has a dropdown menu for the course, a dropdown menu for the semester, and a text input field for the class per week. There is an 'x' icon next to each row.

Course	Semester	Class Per Week
Physics	Semester III	2
Mathematics-II	Semester III	2
Mathematics-I	Semester II	2
Theory of Computing	Semester II	1
C Programming	Semester II	2
System Analysis and Design	Semester II	2
Data Structures	Semester II	2
Web Development	Semester I	1
Operations Research I	Semester I	2
Artificial Intelligence	Semester I	2

On the right side of the form, there is a sidebar with a table titled 'ROOMS' and 'ACTIONS'. The 'ACTIONS' column has a green edit icon and a red delete icon.

Figure 5.7.3 UI (Update Class Page)

5.7.4 Delete Class

Only admin can delete class.

The screenshot shows the 'Delete class' dialog box in the Timetable Displayer application. The dialog box is titled 'Delete class' and contains the text 'Do you want to delete this class?'. There are two buttons: 'No' and 'Yes'.

The background shows the 'Classes' page in the Timetable Displayer application. The page has a sidebar on the left with navigation links: Dashboard, Manage Rooms, Manage Courses, Manage Course Teachers, Manage Classes (highlighted), Manage Periods, My Account, and Log Out. The main content area is titled 'Classes' and contains a table with the following columns: NAME, UNAVAILABLE ROOMS, and ACTIONS. The table has three rows: SEC A, SEC B, and SEC C.

NAME	UNAVAILABLE ROOMS	ACTIONS
SEC A	None specified	 
SEC B	200	 
SEC C	Semester I	 

The 'SEC A' row is expanded, showing a list of courses:

- Semester I
 - SWE 305 System Analysis
 - SWE 306 Artificial Intelligence
 - SWE 307 Operations Research I
 - SWE 308 Web Development
- Semester II
 - SWE 311 Data Structures
 - SWE 312 System Analysis and Design
 - SWE 313 C Programming
 - SWE 314 Theory of Computing
 - SWE 315 Mathematics-I
- Semester III
 - SWE 301 Computer Architecture
 - SWE 302 Computer Graphics
 - SWE 303 Algorithms
 - SWE 404 Mathematics-II
 - SWE 414 Physics

Figure 5.7.4 UI (Delete Class Page)

5.8 Admin Create Timetable

Only admin can create timetable per semester.

The screenshot shows the 'Timetable Displayer' admin interface. A modal titled 'Create New Timetable Set' is open. The modal contains the following fields and options:

- Timetable Name:** A text input field.
- Semester:** A dropdown menu with the text 'Select an Semester'.
- Select Days:** A group of radio buttons for each day of the week: Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, and Friday.
- Buttons:** 'Cancel' (red) and 'Submit' (teal) buttons at the bottom.

The background interface shows a sidebar with 'Dashboard', 'Manage Rooms', 'Manage Courses', 'Manage Course Teachers', 'Manage Classes', 'Manage Periods', 'My Account', and 'Log Out'. The main content area has 'Rooms 9' and 'Classes 2' cards, and a table with 'PRINT' buttons.

Figure 5.8 UI (Create Timetable Page)

5.9 Student Sign Up

Students can create an account using a student name, email, and password.

The screenshot shows a web browser window with the URL 'http://127.0.0.1:8000/student/register'. The page has a dark background with a large, ornate clock face. A white 'Student SignUp' form is centered on the page. The form contains the following fields and a button:

- Name:** A text input field with placeholder text 'Full name'.
- Email:** A text input field with placeholder text 'Email'.
- Password:** A text input field with placeholder text 'Password'.
- Confirm Password:** A text input field with placeholder text 'Confirm Password'.
- Submit:** A black button with white text.

Figure 5.9 UI (Student Sign up Page)

5.10 Student Login

Students can log in using student email and password.

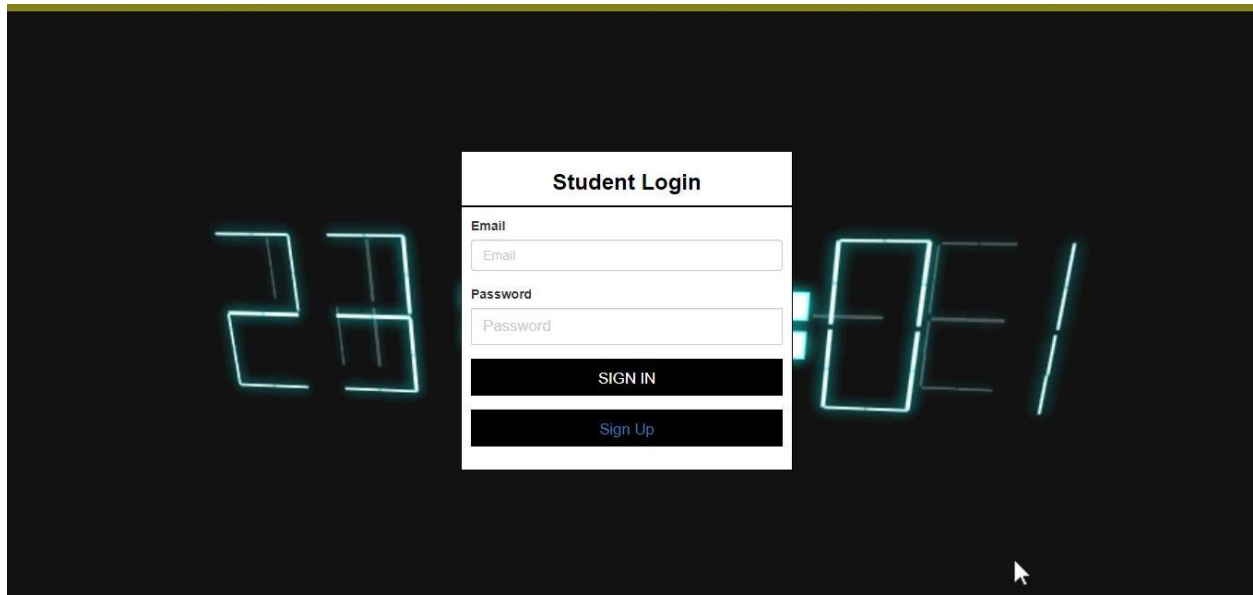


Figure 5.10 UI (Student Login Page)

5.11 Student Dashboard

Students can view a dashboard with all timetable lists.

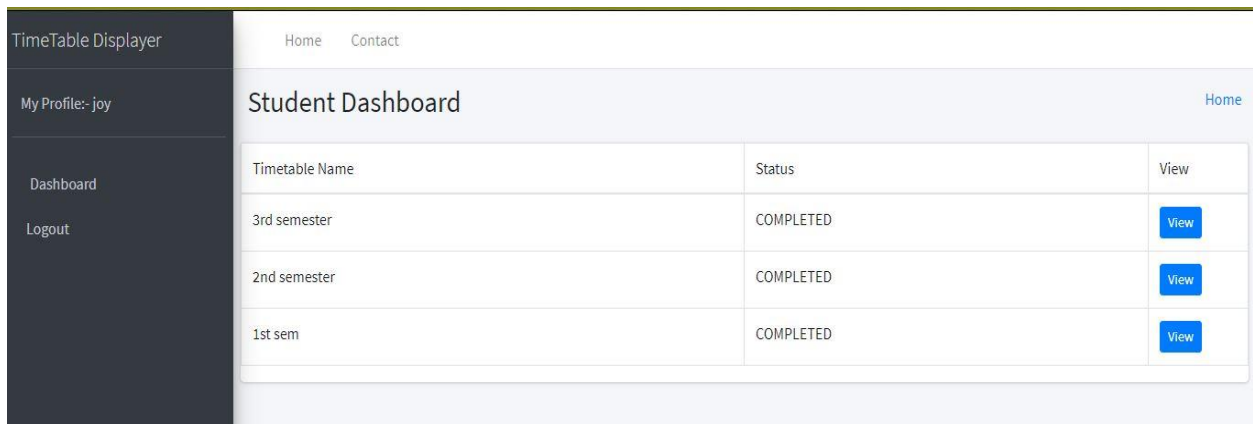


Figure 5.11 UI (Student Dashboard Page)

5.12 Student Profile

Student can view profile.

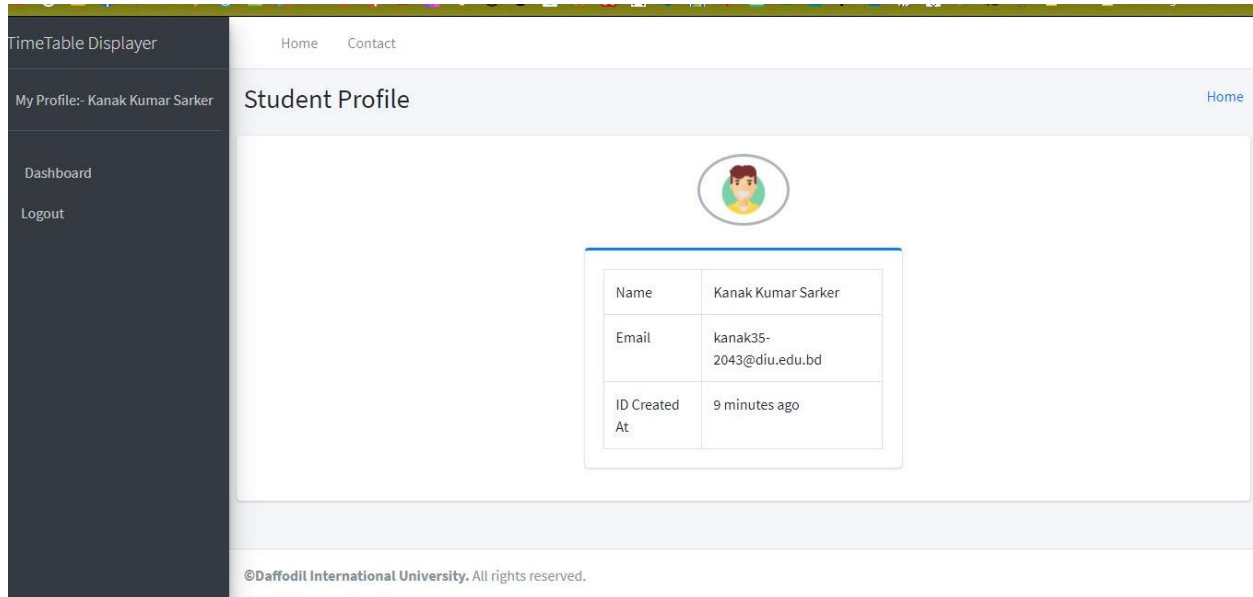


Figure 5.12 UI (Student Profile Page) [2]

5.13 View Timetable and Download

Users can view & download class routines per semester.

SEC A						Download PDF
DAYS	8:00 - 10:30	10:30 - 12:00	12:00 - 13:30	13:30 - 15:00	15:00 - 16:30	
MON						
TUE	SWE 404 SP 01 PHE Sarobita Islam					
WED		SWE 301 PP 1 PHE Nazimul Islam	SWE 414 SP 1 PHE Shadya Roy		SWE 302 SP 20 PHE Nazimul Islam	
THUR		SWE 302 SP 20 PHE Nazimul Islam				
FRI		SWE 303 SP 20 PHE Nazimul Islam	SWE 414 PP 1 PHE Shadya Roy	SWE 302 PP 1 PHE Nazimul Islam	SWE 404 PP 1 PHE Sarobita Islam	
SAT		SWE 301 PP 1 PHE Nazimul Islam	SWE 303 SP 01 PHE Nazimul Islam			
SEC B						
DAYS	8:00 - 10:30	10:30 - 12:00	12:00 - 13:30	13:30 - 15:00	15:00 - 16:30	
MON		SWE 314 SP 1 PHE Shadya Roy				

Figure 5.13 UI (View Timetable and Download Page)

5.14 PDF Format

Users can view pdf format.

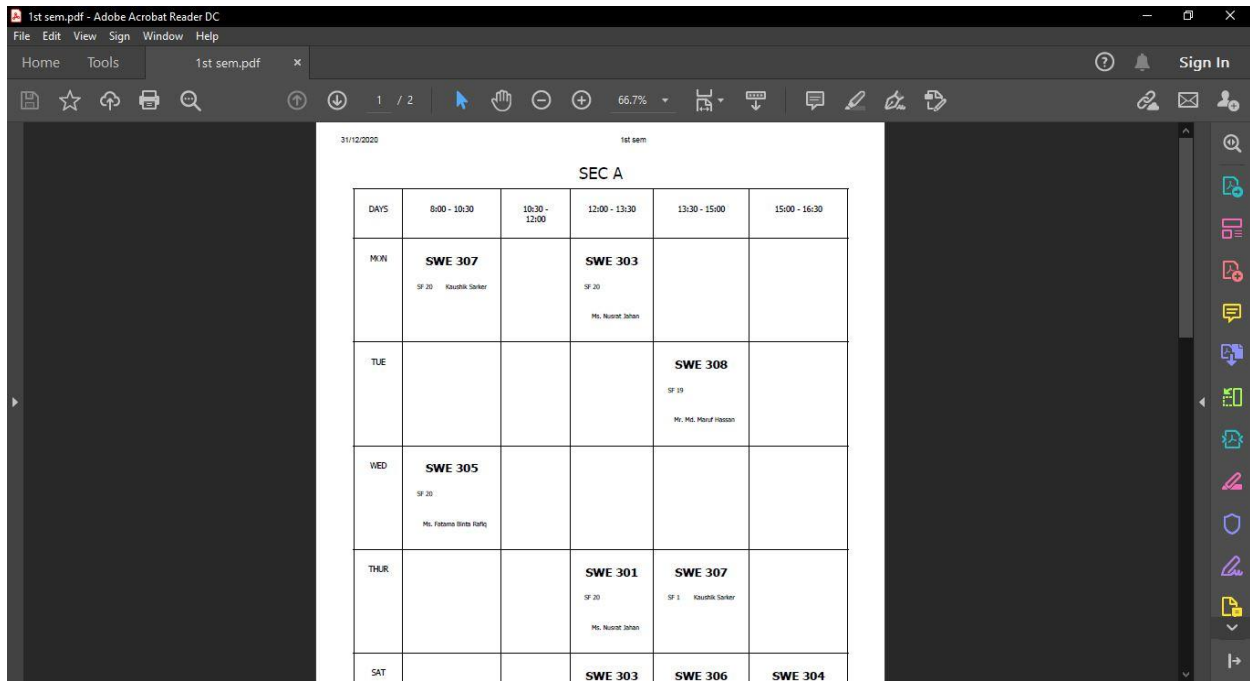


Figure 5.14 UI (PDF format Page)

CHAPTER 6

PROJECT SUMMARY

6.1 GitHub Link

https://github.com/Kanak43/Timetable_Displayer

6.2 Limitations

On my way, I faced some limitations. I am also learning the relevant technology, but I have not yet been able to conquer these challenges. But I'm hopeful that I'll be able to learn the advanced topics and develop the code so much with enough time.

Email Server: I was unable to set up an email server to retrieve passwords. But it still requires the module.

Mobile version: Because it is a travel-related service, it will be chosen by the consumer on their mobile device. I couldn't build a mobile version of the application as I'm not yet adept at mobile related technology and also the time was tight.

6.3 Obstacles and Achievements

I have learned too many things since the beginning of the method that is important for the developer.

Firstly, I don't know how to properly develop a design, such as how to write algorithms and diagrams connected with project work, such as database design. I don't know the meaning of the algorithm until this and how much it is required for a programmer to build a project. Many important aspects of the vocabulary I use to build this system at the time of constructing this system are important. Before beginning the logical part, I also learn if the database design and project UI are ready, then executing the code would be very simple. It was, in a word, a major achievement for me to build this program.

6.4 Future Scope

The software being developed is the software of good quality. I'm doing my hardest to fulfill the true need for the stage of the procedure. But still, there is space to expand more. I have introduced the program and do my hardest to produce an exceptional system.

6.5 References

- [1] "creately," creately, 2008-2021. [Online]. Available: <https://creately.com/>. [Accessed 2021].
- [2] AdminLTE, "AdminLTE," AdminLTE, 2014-2019. [Online]. Available: <https://adminlte.io/>. [Accessed 2021].
- [3] J. resig., "Jquery.," 2006. [Online]. Available: <http://jquery.com/>. [Accessed 12 09 2020].
- [4] K. Sun, "Lucidchart," 2008. [Online]. Available: www.lucidchart.com/pages/. [Accessed 10 10 2020].
- [5] s. c. j. k. Chad hurley, "YouTube," 2005. [Online]. Available: <https://www.youtube.com/>. [Accessed 06 08 2020].

6.6 Plagiarism Report

[Document Viewer](#)

Turnitin Originality Report

Processed on: 30-Jan-2021 12:54 +06
ID: 1497608272
Word Count: 7174
Submitted: 1

171-35-2043 By Kanak Kumar Sarker

Similarity Index 20%	Similarity by Source <table><tr><td>Internet Sources:</td><td>18%</td></tr><tr><td>Publications:</td><td>1%</td></tr><tr><td>Student Papers:</td><td>8%</td></tr></table>	Internet Sources:	18%	Publications:	1%	Student Papers:	8%
Internet Sources:	18%						
Publications:	1%						
Student Papers:	8%						

[include quoted](#) [include bibliography](#) [excluding matches < 10 words](#) mode:

quickview (classic) report ▼ Change mode [print](#) [refresh](#) [download](#)