

**DESIGN AND DEVELOPMENT OF INTEGRATED CLASS ROUTINE INFORMATION
AND MANAGEMENT SYSTEM**

BY

THANJIDA AKHTER

ID: 102-15-1030

AND

SAIFUL ISLAM

ID: 102-15-1045

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

Supervised By

Dr. Syed Akhter Hossain

Professor and Head

Department of Computer Science and Engineering (CSE)

Daffodil International University (DIU)



**DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
MAY 2014**

APPROVAL

This Project titled "**Design and Development of Integrated Class Routine Information and Management System**", submitted by Thanjida Akhter and Saiful Islam 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 11 May.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain

Professor and Head

Department of CSE

Faculty of Science & Information Technology

Daffodil International University

Chairman

Dr. Sheak Rashed Haider Noori

Internal Examiner

Assistant Professor

Department of CSE

Faculty of Science & Information Technology

Daffodil International University

Anisur Rahman

Internal Examiner

Assistant Professor

Department of CSE

Faculty of Science & Information Technology

Daffodil International University

Dr. Muhammad Shorif Uddin

External Examiner

Professor

Department of CSE

Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Dr. Syed Akhter Hossain, Professor and head, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

**Dr. Syed Akhter Hossain
Professor and head
Department of CSE
Daffodil International University**

Submitted by:

**Thanjida Akhter
ID: 102-15-1030
Department of CSE
Daffodil International University**

**Saiful Islam
ID: 102-15-1045
Department of CSE
Daffodil International University**

ACKNOWLEDGEMENT

First and foremost, we would like to express our gratitude to almighty Allah for successful completion of our project work. We take the opportunity to thanks Daffodil International University for providing us with such a graceful opportunity to become a part of this institution. It has been a privilege for us to pursue the Degree of Bachelor of Science in Computer Science and Engineering here.

We would like to thanks our supervisor Dr. Syed Akhter Hossain, Professor and head, Department of Computer Science and Engineering, Daffodil International University and also thanks all the faculty members of the department who have helped us in various ways during the period.

Lastly we would like to thanks all persons those are related to the project directly or indirectly.

ABSTRACT

The life cycle of any academic institution may it be school, college or university depends on successful integration of information with the management of the institution. One of the most important facts in routines especially class routines management and its synchronization with timely updates. This project was initiated 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 project intended to facilitate necessary sharing, exchange and update time, to time as per the needs of the department.

The requirement specification was gathered from the practices of the department. Based on the requirement the propose system was designed keeping the simplicity in the mind. The system was implemented using MVC3 architecture in .NET platform with MSSQL as a backend database. The implemented system was tested thoroughly by the teachers in the department and found satisfactory. In addition a mobile apps based on android platform will be developed in future to facilitated general query about class routine for the end user. In the future the system will be integrated with learning management system.

TABLE OF CONTENTS

CONTENTS	PAGE NO
BOARD OF EXAMINERS	I
DECLARATION	II
ACKNOWLEDGMENTS	III
ABSTRACT	IV
 CHAPTER	
CHAPTER 1: INTRODUCTION	1-3
1.1 Objective	1
1.2 Motivation	2
1.3 Layout of the Report	3
CHAPTER 2 ROUTINE INFORMATION SYSTEM	4-6
2.1 Introduction	4
2.2 Typical Routine Management System	4
2.3 Existing Solution and Challenges	5
CHAPTER 3: REQUIREMENT SPECIFICATION	7-19
3.1 Introduction	7
3.2 General System Requirement	7
3.3 Use Case Model	8
3.4 Use Case Description	9
3.5 Conceptual Data Model	19

CHAPTER 4: DESIGN AND IMPEMENTION	20-44
4.1 Implementation of Data Model	20
4.2 UI Design and Model	29
4.3 System Implementation	41
4.4 Integration and Testing	41
CHAPTER 5: CONCLUSION AND FUTURE SCOPE	45-46
5.1 Discussion and Conclusion	45
5.2 Future Scope	46
REFERENCE	47

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.3.1: Use Case Model (Web)	8
Figure 3.3.2: Use Case Model (Mobile)	9
Figure 3.5.1: Conceptual Data Model (ERD Diagram)	19
Figure 4.1.1: Database Design	20
Figure 4.1.2: Database Design	21
Figure 4.2.1: Home page	29
Figure 4.2.2: Home Page (Show Routine)	30
Figure 4.2.3: About Page	31
Figure 4.2.4: Login Page	32
Figure 4.2.5: Teacher Page	32
Figure 4.2.6: Teacher (Create a single Teacher)	33
Figure 4.2.7: Teacher (Create multiple Teacher at a time)	34
Figure 4.2.8: Course (Initial View)	34
Figure 4.2.9: Course (create a single course)	35
Figure 4.2.10: Course (create multiple course)	36
Figure 4.2.11: Classroom Page	36
Figure 4.2.12: Department Page	37
Figure 4.2.13: Upload Routine Page	37
Figure 4.2.14: Upload Successful	38
Figure 4.2.15: Upload Failed	38
Figure 4.2.16: Upload Failed	39

Figure 4.2.17: Routine Page (without filter)	39
Figure 4.2.18: User Management	40
Figure 4.2.19: Empty Slots	40
Figure 4.4.1: Login Testing	41
Figure 4.4.2: Upload Testing	42
Figure 4.4.3: Upload Testing-1	42
Figure 4.4.4: Upload Testing-2	43
Figure 4.4.5: Teacher Create Test	43
Figure 4.4.6: Course Create Test	44

LIST OF TABLE

TABLES	PAGE NO
Table 3.4.1: Use case description of Login	10
Table 3.4.2: Use case description of Create User	11
Table 3.4.3: Use case description of Manage User	11
Table 3.4.4: Use case description of Upload Routine	12
Table 3.4.5: Use case description of Manage Routine	13
Table 3.4.6: upload Course Offer	13
Table 3.4.7: Use case description of Manage Course Offer	14
Table 3.4.8: Use case description of Create/Upload Course List	14
Table 3.4.9: Use case description of Manage Course List	15
Table 3.4.10: Use case description of Create/Upload Class Room	15
Table 3.4.11: Use case description of Manage Class Room	16
Table 3.4.12: Use case description of Create/Upload Teacher	16
Table 3.4.13: Use case description of Manage Teacher	17
Table 3.4.14: Use case description of Search Routine	14
Table 4.1.1: Application Table	22
Table 4.1.2: Classroom Table	22
Table 4.1.3: Course Table	22
Table 4.1.4: Day Table	23
Table 4.1.5: Department Table	23
Table 4.1.6: Membership Table	24
Table 4.1.7: Profile Table	25

Table 4.1.8: Role Table	25
Table 4.1.9: Routine Table	26
Table 4.1.10: Routine Type Table	26
Table 4.1.11: Semester Table	27
Table 4.1.12: Teacher Table	27
Table 4.1.13: User Table	27
Table 4.1.14: User Id Role Table	28

CHAPTER 1

INTRODUCTION

1.1 Objectives

Routine Information system is one of the most important tasks of an educational system. Our main goal is to develop a web based Routine Information system which will mainly be able to

- Upload, update Routine on the web
- Search Routine on course code, semester, classroom, teacher initial, timing, level, term and day.

Upload Routine is possible only for a CSV file format. Update routine gives us the option to remove a single class on a defined day and time. It also gives us the option to add a class on a defined day and time.

Search Routine gives us the option to find a routine for a defined semester of a single department and then customize the routine as one's necessity. For example a student of second semester can get his or her routine from the main routine by filtering it with level and term as 2, 2. Similarly one can find out the minimum routine list as he or she desires.

Besides these there are many minor but important goals for this project. Like one can easily find empty slots from a routine and book that room for a defined period of time.

One can print the routine from the system after an update is made and many more like these.

1.2 Motivation

One of the most essential facts in Routine management system is synchronization of timely updates in a semester's routine. In this current time our university is using a manual routine management system which is implemented by the departmental teachers and submitted in the departmental notice board as a printed copy. This routine is also submitted in the virtual notice board once and once only per semester. When it gets updated it is again shown in the departmental notice board as a printed copy. But this updated information is not available to the virtual notice board. It is also not available to the teachers and departments authority until they are in need of a printed copy. The students collect their routine from the departmental notice board again and again. So it became necessary to find a way by which we could solve the problem for all kind of user. This scenario motivated us to come up front with a web based solution like ours which we are describing here. This project was initiated from the thoughts of the teachers of CSE department of Daffodil International University. It was necessary for sharing, exchanging and updating routine information of a semester with all the faculty members of the department. The necessity of the teachers for a media through which they could share or update the routine and the requirement of the students to find his or her required part in the routine have motivated us to build the system.

For the last four years of our B.Sc. program we have collected the routine from the notice board manually. Then find out the classes as per our need. These activities motivated us to find a way so that no one has to go through the searching manually. Instead they will tell the software to do that for them.

Our Department Head Prof. Dr. Syed Akhter Hossain also motivated us to come up with a solution that will help us to find the Routine System more easy and flexible.

1.3 Layout of the Report

Layout of the report means the overall view of the project report where one describes about all chapters in a report and the information which the report contains. Our project “Integrated Class Routine Information and management System” has been described within this Project report in a total of five chapters.

Chapter one gives us the overall view of this report containing our objectives and motivation to complete this project as a successful one. It also contains Introduction and the layout report part of this project.

Chapter two starts with some discussion about the functionality of the routine information system and following up with introduction part, typical routine management system, existing solution and challenges for the project.

Before getting started one have to collect enough information about the work he is going to do in order to get successful in that particular work. Chapter three describes what we have done in order to collect enough information to complete “Integrated Class Routine Information and management System” project. In short one can say that this chapter is about the requirement specification of our project. Here one can find the general system requirement, Use Case Model of the project and the conceptual data model.

After the completion of requirement collection the next step is to go to develop the project for which one has worked so hard. Chapter four is the one where we have discussed about how we have implemented the project to develop the software. We have shown the Database design, its table definition, UI design and its work in this chapter. From here one can find out how we have worked through the developing phase of this particular project. One will also find the system integration and testing part of our project.

The last but not the least we have another chapter. This is the last one of this project report. Chapter five of our project contains the discussion and the conclusion part where we have shared our point of view about this project. We have also discussed about the future plan of our project here. The above discussions are all about the content of the chapters of our project report.

CHAPTER 2

ROUTINE INFORMATION SYSTEM

2.1 Introduction

Every School, College, University or any Educational System has a common goal. To develop a good Routine Information System based on which they can control their all activities. Not only for educational system, is it essential for every work if we want to make it a successful one. Here we are going study about the Routine Information System of Daffodil International University and its possibility to make it an e-based routine system.

2.2 Typical Routine Management System

In our country normally a routine is made by the teachers manually which consists of some common things. Such as

- Classroom
- Time
- Day
- Subject or Course
- Section

Daffodil International University's Routine System is quite similar to the typical routine system of our country. The routine is made by a teacher for a department and for every department there are some teachers who create the routine manually. They calculate everything by themselves. Handling all the exceptions, errors and ensuring every one's flexibility is a tough job but this job is done by the teachers manually. But as human is not out of error they makes error too. And this error is solved by creating the routine again and again. This process kills a lot of time of both teachers and students to ensure a good curriculum by which they both can benefit.

2.3 Existing Solution and Challenges

On the Golden age of Modern Technology and e-commerce system it is sad to think that there is no existing e-based solution for Routine Information System. Currently the pdf file of the manual routine is shared in both online notice board and Department's notice board to share the routine with everyone. On updating the routine a second pdf file is available to share again. This process continues until the routine is in a satisfactory position for everyone. Alas! This way the goal is very hard to achieve.

To convert this Routine Information system to an e-based Information system and integrate it with the current system is a big challenge itself. Besides this we have to learn a many things to do as we desire. So there are two kinds of challenges for us, one is from learning perspective and the other is from current system and its necessity perspective.

Challenges from learning perspective are

- ASP.Net
- MVC3 architecture
- jquery
- Java script
- HTML (Razor syntax)
- CSS
- Ajax
- MSSQL client/server setup

Challenges from current system's necessity perspective are

- To be able to upload the routine file, then parse it automatically
- To be able to print a defined routine
- To be able to book a room temporarily
- To be able to upload multiple course information at a time
- To be able to show a teacher's total course which he has taken for a semester
- To be able to upload multiple teacher information at a time
- To be able to modify routine as system requires
- To be able to search the routine based on different things like teacher initial, course code, day, time, section etc.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Introduction

Every Website needs some Hardware components and certain Software resources to be present on the system to run efficiently. These prerequisites are known as Requirement Specification and are often used as a guideline as opposed to an absolute rule.

3.2 General System Requirement

General system requirement deals with both minimum amounts of software and hardware that is needed to run the optimal functioning of an application. To run “Integrated Routine Management System” application on a website one should need the following:

- Windows XP or higher
- Microsoft .Net9
- MSSQL
- Browser (Firefox, Chrome etc.)
- Microsoft Office (Excel to convert CSV)
- Pentium – 4 processor
- 512MB RAM
- Hard Disk Space (50MB)
- Keyboard
- Serial Mouse
- Color Monitor
- Printer

With above software and hardware one should be able to work with “Integrated Routine Management System”.

3.3 Use Case Model

The following figure is our use case model for the Integrated Class Routine Information and management system. It shows the abstract view of our web site and its corresponding activity. According to the use case one can easily say that our web site mainly works with three kinds of use. They are Admin, Teacher and student.

Admin can create and manage user, Upload and manage routine, create and manage course list, create and manage teacher list, create and manage department list, create and manage temporary routine, create and manage classroom and last but not the least he can add roles to the users.

Teachers can only see all the routines and he can book a room temporarily in the routine. One teacher can book only three rooms in a week for extra classes.

Students can see any routine as he defines. He can search a routine with lots of option until he gets his desire results.

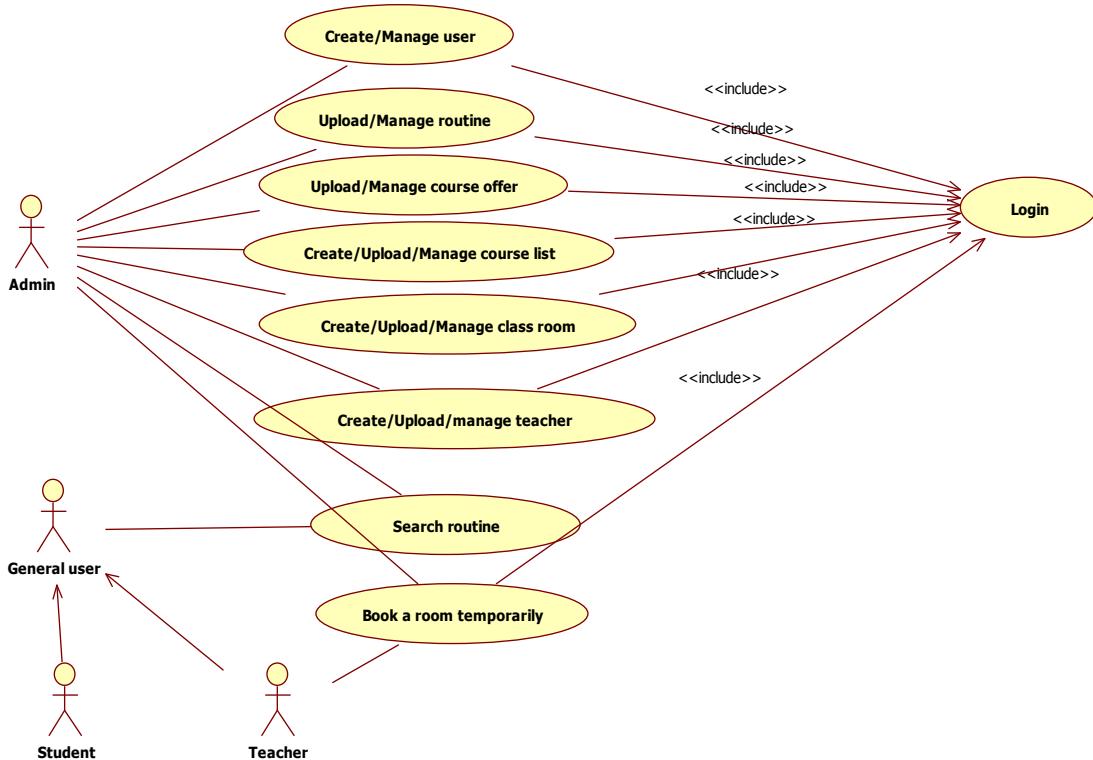


Figure 3.3.1: Use case model for integrated routine information and management system (Web)

Following figure is the Mobile Application use case model's view which will be our future plan to support this application in the MOBILE platform. The application detail is pretty much simple. This will be able to search the routine only. The search could be for a full routine or for a partial routine of a define semester.

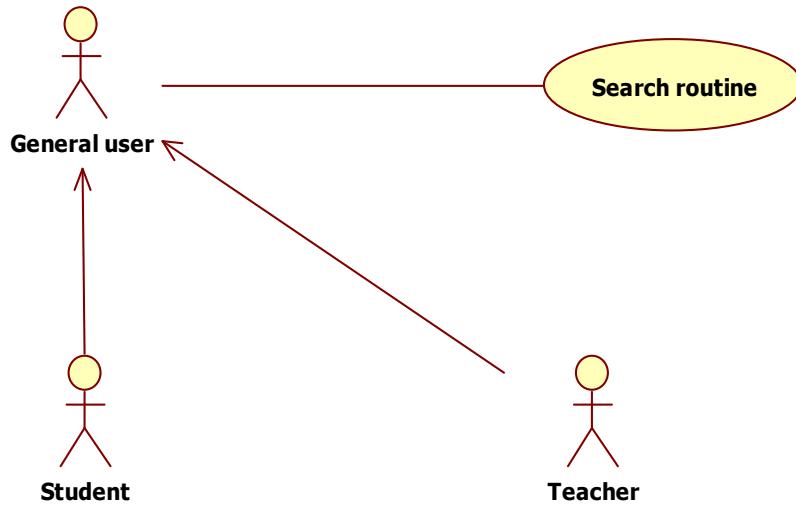


Figure 3.3.2: Use case model for integrated routine information and management system
(Mobile)

3.4 Use Case Description

It contains information about all the use case of the Use case Model shown in the above picture. We have described every use case through a table in the below section. Each table shows five kinds of detail information about a use case. They are Use case name, Actor, Pre-condition, Primary Path and Exceptional Path. Use case name means the name of the process. Actor defines who will be the user. Pre-condition tells whether or not another process is needed to enter or use this process. Primary Path defines how the process works step by step. And last but not the least Exceptional Path shows us what will the process do if any exception occurs in a primary path.

We have a total of thirteen tables which is given below for the better understanding of processes of our project.

Table 3.4.1 describes about “login” process of our project and it gives us an understanding how the process works in project. Login process has three primary paths they are enter user ID, enter password and click login button. In this three primary path have two exceptional paths that are user ID/Password is invalid.

Table 3.4.1: Use case description of Login

Use case name:	Login
Actor:	Admin
Pre-condition:	None
Primary Path:	<ol style="list-style-type: none">1. Enter user ID2. Enter password3. Click “Login” Button
Exceptional Path:	<ol style="list-style-type: none">1.1 User ID/Password is invalid2.1 User ID/Password is invalid

In table 3.4.2 describe about “create user” process of our project and given us an understanding how this process works in this project. Create new user admin must be login. To create user at first admin have to give some information about user. There are also some exceptional path which defines, user name must be characters only, user email address must be valid and unique, also user ID have to be unique and confirm password should same as the password.

Table 3.4.2: Use case description of Create User

Use case name:	Create User
Actor:	Admin
Pre-condition:	Login
Primary Path:	<ol style="list-style-type: none"> 1. Enter Name 2. Enter Email 3. Enter user ID 4. Enter password 5. Enter confirm password 6. Click “Create” button
Exceptional Path:	<ol style="list-style-type: none"> 1.1 Name should be characters only 2.1 Email address is not correct 2.1 This email is already in use 3.1 This user ID is already in use 5.1 Password does not match

In table 3.4.3 there is description about “Manage user” process of our project and which gives us an understanding how this process works in the project. To manage user admin have to login first. Manage user has seven primary path. In this process admin can delete, block, edit or make new admin. To delete, block, edit or make new admin have to fill all fields.

Table 3.4.3: Use case description of Manage User

Use case name:	Manage User
Actor:	Admin
Pre-condition:	Login

Primary Path:	<ol style="list-style-type: none"> 1. Select a user 2.2 Click “Delete” or “Block” button 2.3 Click “Edit” button 2.3 Click “Make admin” button 3.1 Click “Confirm” button 3.2 Enter update data 4.2 Click “Save” button
Exceptional Path:	3.2.1 There cannot be an empty field

Table 3.4.4 describes about “Upload routine” process of our project. To manage user admin should be login. It has two primary paths at first select a file in your computer and then press “Upload” button. File must be CSV format.

Table 3.4.4: Use case description of Upload Routine

Use case name:	Upload routine
Actor:	Admin
Pre-condition:	Login
Primary Path:	<ol style="list-style-type: none"> 1. Select a file 2. Press upload button
Exceptional Path:	<ol style="list-style-type: none"> 1.1 Error occurred, please upload again 2.1 File should be CSV type

Table 3.4.5 describes about “Manage Routine” process of our project. Here at first admin have to select a routine row and then he/she can edit or delete routine data. After edit or delete he/she can save current occurrence. There is no exceptional path for primary path.

Table 3.4.5: Use case description of Manage Routine

Use case name:	Manage Routine
Actor:	Admin
Pre-condition:	Login
Primary Path:	<ol style="list-style-type: none"> 1. Select a routine 2.1 Click “Edit” button 2.2 Click “Delete” button 3.1 Change the data 3.2 Click “Confirm” button 4.1 Click “Save” button
Exceptional Path:	None

Table 3.4.6 describes about “Upload course offer” process of our project and gives us an understanding how this process works in this project.

Table 3.4.6: upload Course Offer

Use case name:	Upload course offer
Actor:	Admin
Pre-condition:	Login
Primary Path:	<ol style="list-style-type: none"> 1. Select a semester 2. Select a course 3. Click upload button
Exceptional Path:	3.1 This course is already in use for this semester

In table 3.4.7 we described about “Manage course offer” process of our project and which gives us an understanding how this process works in this project.

Table 3.4.7: Use case description of Manage Course Offer

Use case name:	Manage course offer
Actor:	Admin
Pre-condition:	Login
Primary Path:	1.Select a semester 2.Click to choose a course 3.Clock remove button
Exceptional Path:	None

“Create/Upload course list” is described in table 3.4.8 and this process defines how to upload a course list to save in the database.

Table 3.4.8: Use case description of Create/Upload Course List

Use case name:	Create/Upload course list
Actor:	Admin
Pre-condition:	Login
Primary Path:	1. Enter course name 2. Enter course ID 3. Click create button
Exceptional Path:	1.1 This name already exists 2.1 This ID already exists

“Manage course list” is a process to modify the uploaded course list which has been described in the table 3.4.9. The process is very simple and easy to work with.

Table 3.4.9: Use case description of Manage Course List

Use case name:	Manage course list
Actor:	Admin
Pre-condition:	Login
Primary Path:	1. Select a course 2.1 Click “Edit” button 2.2 Click “Delete” button 3.1 Enter new course name 4.1 Enter new course code 5.1 Click “Save” button
Exceptional Path:	3.1.1 This name is already used 4.1.1 This code is already used

In table 3.4.10 “Create/Upload class room” process of our project is described. This process shows how to create or upload classrooms data in the website.

Table 3.4.10: Use case description of Create/Upload Class Room

Use case name:	Create/Upload class room
Actor:	Admin
Pre-condition:	Login
Primary Path:	1. Enter a class room 2. Enter description(if any) 3. Click “Upload” button
Exceptional Path:	3.1 This class room already exist

Table 3.4.11 describes about “Manage class room” process of our project and given us an understanding how this process works in this project

Table 3.4.11: Use case description of Manage Class Room

Use case name:	Manage class room
Actor:	Admin
Pre-condition:	Login
Primary Path:	<ol style="list-style-type: none">1. Select a class room2. Click “Remove” button
Exceptional Path:	None

Table 3.4.12 describes about “Create/Upload Teacher” process of our project and which gives us an understanding how this process works in this project

Table 3.4.12: Use case description of Create/Upload Teacher

Use case name:	Create/Upload Teacher
Actor:	Admin
Pre-condition:	Login
Primary Path:	<ol style="list-style-type: none">1. Enter name2. Enter email3. Enter contact number4. Select a designation5. Enter address6. Click “Create” button
Exceptional Path:	<ol style="list-style-type: none">1.1 Name should be characters only1.1 Email is not valid1.2 Email is already in use3.1 Contact should be number only

In table 3.4.13 “Manage Teacher” process of our project has been described.

Table 3.4.13: Use case description of Manage Teacher

Use case name:	Manage Teacher
Actor:	Admin
Pre-condition:	Login
Primary Path:	<ol style="list-style-type: none"> 1. Select a teacher 2.1 Click “Edit” button 2.2 Click “Remove” button 3.1 Enter upload data 3.2 Click “Confirm” button
Exceptional Path:	3.1.1 Update data is not correct

“Search routine” is one of the most important processes in the project. This allows a user to search a routine as he or she likes to search. One can customize a defined routine with time, course-code, teacher initial, day, classroom, level, term and with many other fields.

Table 3.4.14: Use case description of Search Routine

Use case name:	Search routine
Actor:	Admin, Teacher, Student
Pre-condition:	None
Primary Path:	<ol style="list-style-type: none"> 1.1 Select search by time button 1.2 Select search by course code button 1.3 Select search by level/semester button 1.4 Select search by class room button 1.5 Select search by teacher initial button 2.1 Select start time 2.2 Select a course code 2.3 Select a semester/level

	<p>2.4 Select a class room</p> <p>2.5 Select a teacher initial</p> <p>1.1 Select finish/end time</p> <p>1.2 Click “Search” button</p> <p>1.3 Click “Search” button</p> <p>1.4 Click “Search” button</p> <p>1.5 Click “Search” button</p> <p>4.1 Click “Search” button</p>
Exceptional Path:	None

3.5 Conceptual Data Model

Conceptual Data Model represents the first ER diagram which we had drawn by gathering all the information about the routine management system of our university from our CSE department teachers. Here we have in total of six entities and three relations. The entities are Classroom, Semester, Course, Teacher, Department and Db user. A department has many teachers and many courses. A routine is a combination of all the entities except Db user and it has a time span, section, day and a year. Figure 3.4.1 shows the conceptual data model of “Integrated Routine Management System”.

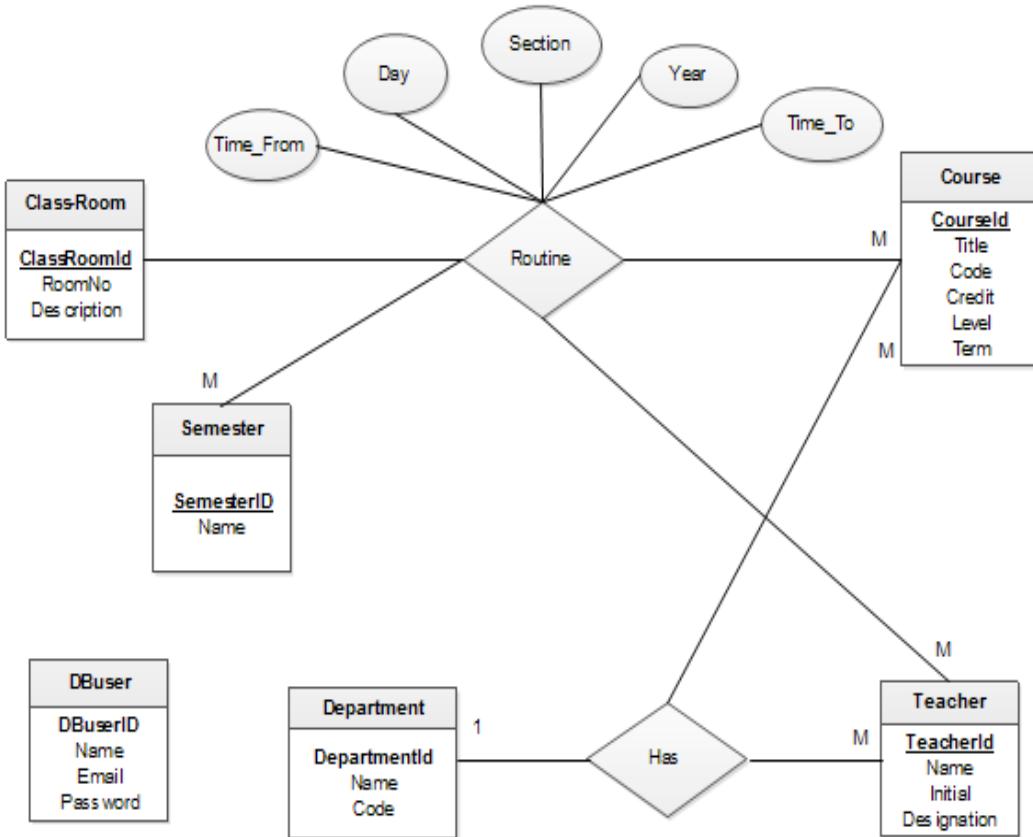


Figure 3.5.1: Conceptual Data Model (ERD Diagram)

CHAPTER 4

DESIGN AND IMPEMENTATION

4.1 Implementation of Data Model

In the following there are two figures for the database design. One is created on the basis of the conceptual data model and the other is to control the users, its roles, membership and profiles, in short to manage users.

On the basis of our conceptual data model we implemented a database in the MSSQL and the following view is shared to understand how the database has been designed. Here we can see that a department has many courses and many teachers. Routine has courses, departments, classrooms, semesters and the other attributes as our conceptual data model. Some extra attributes are added as per the necessity. Some extra entities are also added as per requirement of the department.

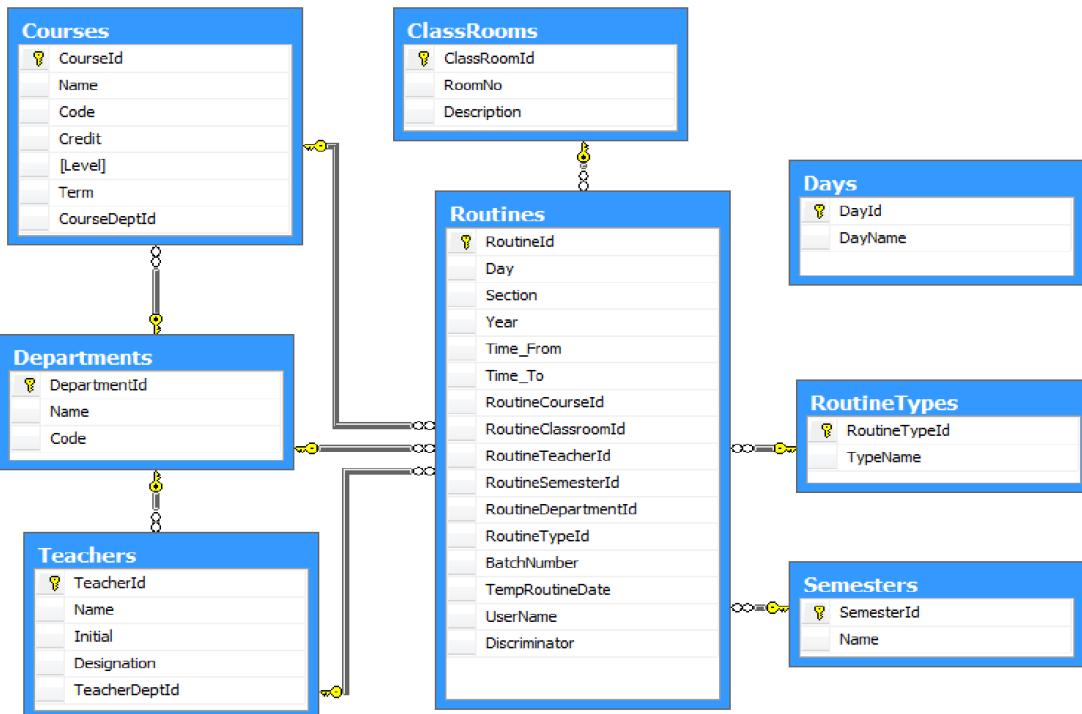


Figure 4.1.1: Database Design

The following figure 4.1.2 shows how the user and its roles are controlled. To ensure total security the user management of the database design has been made as complex as possible but simple from the front end of the website.

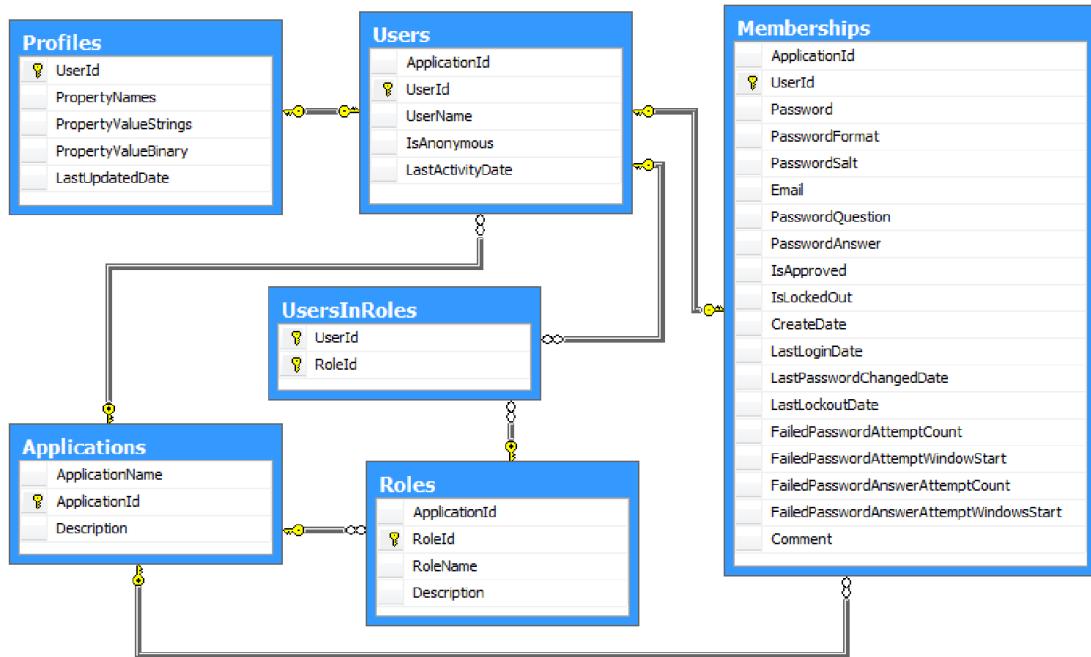


Figure 4.1.2: Database Design

The following are the description of database table and its attributes.

Application Table shows in table 4.1.1 it has three attributes where ApplicationsId is the primary key, no foreign key value and the other two are character type data which is required to store information about the entire Application.

Table 4.1.1: Application Table

	Column Name	Data Type	Allow Nulls
	ApplicationName	nvarchar(235)	<input type="checkbox"/>
!	ApplicationId	uniqueidentifier	<input type="checkbox"/>
	Description	nvarchar(256)	<input checked="" type="checkbox"/>

Classroom Table shows in table 4.1.2 it has three attributes where ClassRoomId is the primary key, no foreign key value and the other two are character type data which is required to store information about the entire classroom.

Table 4.1.2: Classroom Table

	Column Name	Data Type	Allow Nulls
!	ClassRoomId	int	<input type="checkbox"/>
	RoomNo	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Description	nvarchar(MAX)	<input checked="" type="checkbox"/>

Course Table shows in Table 4.1.3 it has seven attributes where CourseId is the primary key, CourseDeptId is a foreign key value and the other five are character type data which is required to store information about all the courses.

Table 4.1.3: Course Table

	Column Name	Data Type	Allow Nulls
!	CourseId	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Code	nvarchar(MAX)	<input type="checkbox"/>
	Credit	nvarchar(MAX)	<input checked="" type="checkbox"/>
	[Level]	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Term	nvarchar(MAX)	<input checked="" type="checkbox"/>
	CourseDeptId	int	<input type="checkbox"/>

Days Table shows in Table 4.1.4 it has seven attributes where DayId is the primary key, no foreign key value and the other one is character type data which is required to store information about all the days.

Table 4.1.4: Day Table

	Column Name	Data Type	Allow Nulls
PK	DayId	int	<input type="checkbox"/>
	DayName	nvarchar(MAX)	<input checked="" type="checkbox"/>
▶			<input type="checkbox"/>

Department Table shows in Table 4.1.5 it has three attributes where DepartmentId is the primary key and the other two are character type data which is required to store information about all the departments.

Table 4.1.5: Department Table

	Column Name	Data Type	Allow Nulls
PK	DepartmentId	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Code	nvarchar(MAX)	<input checked="" type="checkbox"/>
▶			<input type="checkbox"/>

Memberships Table shows in Table 4.1.6 it has three attributes where UserId is the primary key and ApplicationsId is a foreign key value and the other six are date time type, six are character type data, three is integer's type data which is required to store information about all the memberships.

Table 4.1.6: Membership Table

	Column Name	Data Type	Allow Nulls
	ApplicationId	uniqueidentifier	<input type="checkbox"/>
?	UserId	uniqueidentifier	<input type="checkbox"/>
	Password	nvarchar(128)	<input type="checkbox"/>
	PasswordFormat	int	<input type="checkbox"/>
	PasswordSalt	nvarchar(128)	<input type="checkbox"/>
	Email	nvarchar(256)	<input checked="" type="checkbox"/>
	PasswordQuestion	nvarchar(256)	<input checked="" type="checkbox"/>
	PasswordAnswer	nvarchar(128)	<input checked="" type="checkbox"/>
	IsApproved	bit	<input type="checkbox"/>
	IsLockedOut	bit	<input type="checkbox"/>
	CreateDate	datetime	<input type="checkbox"/>
	LastLoginDate	datetime	<input type="checkbox"/>
	LastPasswordChangedDate	datetime	<input type="checkbox"/>
	LastLockoutDate	datetime	<input type="checkbox"/>
	FailedPasswordAttemptCount	int	<input type="checkbox"/>
	FailedPasswordAttemptWindowStart	datetime	<input type="checkbox"/>
	FailedPasswordAnswerAttemptCount	int	<input type="checkbox"/>
	FailedPasswordAnswerAttemptWindowsStart	datetime	<input type="checkbox"/>
	Comment	nvarchar(256)	<input checked="" type="checkbox"/>
▶			<input type="checkbox"/>

Profile Table shows in table 4.1.7 it has five attributes where UserId is the primary key, no foreign key value and the other two are character type data, one is date time type data and one is image type data which is required to store information about the entire profile.

Table 4.1.7: Profile Table

	Column Name	Data Type	Allow Nulls
	UserId	uniqueidentifier	<input type="checkbox"/>
	PropertyNames	nvarchar(4000)	<input type="checkbox"/>
	PropertyValueStrings	nvarchar(4000)	<input type="checkbox"/>
	PropertyValueBinary	image	<input type="checkbox"/>
	LastUpdatedDate	datetime	<input type="checkbox"/>
▶			<input type="checkbox"/>

Role Table shows in table 4.1.8 it has five attributes where RoleId is the primary key, ApplicationId is a foreign key value and the other two are character type data which is required to store information about the entire role.

Table 4.1.8: Role Table

	Column Name	Data Type	Allow Nulls
	ApplicationId	uniqueidentifier	<input type="checkbox"/>
	RoleId	uniqueidentifier	<input type="checkbox"/>
	RoleName	nvarchar(256)	<input type="checkbox"/>
	Description	nvarchar(256)	<input checked="" type="checkbox"/>
▶			<input type="checkbox"/>

Routine Table shows in table 4.1.9 it has sixteen attributes where RoutineId is the primary key, RoutineCourseId, RoutineClassRoomId, RoutineTeacherId, RoutineSemesterId, RoutineDepartmentId, and RoutineTypeId are a foreign key value and the other five are character type data, one is integer's type data and three is date time type data which is required to store information about the entire routine.

Table 4.1.9: Routine Table

	Column Name	Data Type	Allow Nulls
PK	RoutineId	int	<input type="checkbox"/>
	Day	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Section	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Year	nvarchar(MAX)	<input type="checkbox"/>
	Time_From	datetime	<input type="checkbox"/>
	Time_To	datetime	<input type="checkbox"/>
	RoutineCourseId	int	<input type="checkbox"/>
	RoutineClassroomId	int	<input type="checkbox"/>
	RoutineTeacherId	int	<input type="checkbox"/>
	RoutineSemesterId	int	<input type="checkbox"/>
	RoutineDepartmentId	int	<input type="checkbox"/>
	RoutineTypeId	int	<input type="checkbox"/>
	BatchNumber	int	<input checked="" type="checkbox"/>
	TempRoutineDate	datetime	<input checked="" type="checkbox"/>
	UserName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Discriminator	nvarchar(128)	<input type="checkbox"/>

Routine type Table shows in table 4.1.10 it has two attributes where RoutineTypeId is the primary key, no foreign key value and the other one is character type data which is required to store information about the entire routine type.

Table 4.1.10: Routine Type Table

	Column Name	Data Type	Allow Nulls
PK	RoutineTypeId	int	<input type="checkbox"/>
	TypeName	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Semester table shows in table 4.1.11 it has two attributes where Semester is the primary key, no foreign key value and the other one integer type data which is required to store information about semester.

Table 4.1.11: Semester Table

Column Name	Data Type	Allow Nulls
SemesterId	int	<input type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Teacher Table shows in table 4.1.11 it has five attributes where TeacherId is the primary key, TeacherDeptId is a foreign key value and the other three are character type data and two is integer's type data which is required to store information about the entire teacher.

Table 4.1.12: Teacher Table

Column Name	Data Type	Allow Nulls
TeacherId	int	<input type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
Initial	nvarchar(MAX)	<input checked="" type="checkbox"/>
Designation	nvarchar(MAX)	<input checked="" type="checkbox"/>
TeacherDeptId	int	<input type="checkbox"/>
		<input type="checkbox"/>

User Table shows in table 4.1.13 it has five attributes where UserId is the primary key, ApplicationId is a foreign key value and the other one are character type data, one is integer's type data and one is date time type data which is required to store information about the entire user.

Table 4.1.13: User Table

Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
UserName	nvarchar(50)	<input type="checkbox"/>
IsAnonymous	bit	<input type="checkbox"/>
LastActivityDate	datetime	<input type="checkbox"/>
		<input type="checkbox"/>

User Id Role Table shows in table 4.1.14 it has two attributes where UserId andRoleId is the primary key, no foreign key value which is required to store information about the entire user id role.

Table 4.1.14: User Id Role Table

	Column Name	Data Type	Allow Nulls
↑	UserId	uniqueidentifier	<input type="checkbox"/>
↑	RoleId	uniqueidentifier	<input type="checkbox"/>
▶			<input type="checkbox"/>

4.2 UI Design and Interaction

The website was designed simultaneously by communicating with the Department. This section shows the pages of the website and its corresponding activities.

Figure 4.2.1 shows the home page where one can search a routine with some define fields. Here none of the fields can be empty. To find a desired routine successfully one needs to fill all the four fields.

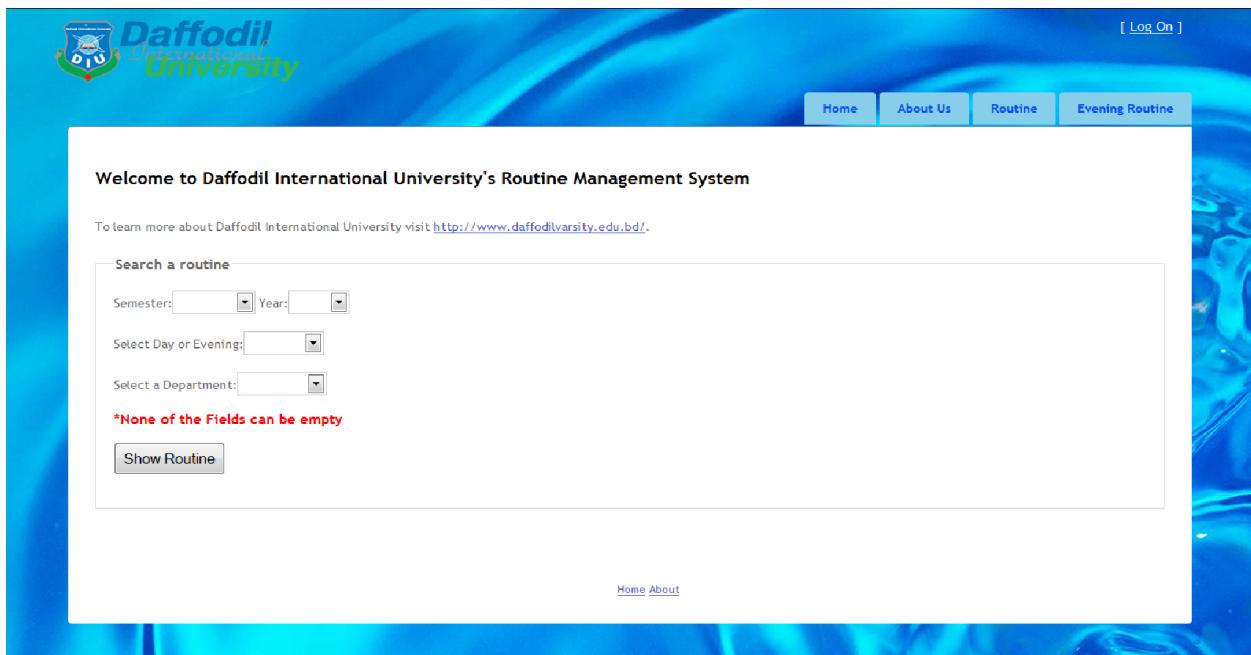


Figure 4.2.1: Home page

The following figure 4.2.2 shows the master routine that will be shown after clicking the “Show Routine” button. Here one will find the master page view of a routine like they are used to see in the departmental notice board. We have given four data in the Home page to show us the master routine of CSE department’s day routine for the semester Spring 2014.

Saturday																	
08:30:00			10:00:00			11:30:00			01:00:00			02:30:00			04:00:00		
Room	Course	Initial	Room	Course	Initial	Room	Course	Initial	Room	Course	Initial	Room	Course	Initial	Room	Course	Initial
L-305AB	ACM	---	L-305AB	ACM	---	L-305AB	ACM	---	L-305AB	ACM	---	L-305AB	ACM	---	L-305AB	ACM	---
L-301AB	CSE 222 (B)	NHM	L-301AB	CSE 222 (B)	NHM	L-301AB	Booked	---	L-301AB	CSE 123 (C)	SRH	L-301AB	CSE 123 (C)	SRH	L-301AB	CSE 311 (C)	MTR
L-304AB	CSE 222 (C)	KA	L-304AB	CSE 222 (C)	KA	L-304AB	CSE 432	APC	L-304AB	CSE 412L (C)	ABMM	L-304AB	CSE 412L (C)	ABMM	L-304AB	CSE 112 (E)	SR
406 AB	---	---	406 AB	CSE 231 (A)	SKM	406 AB	CSE 112 (F)	SKM	406 AB	MGT 414 (A)	FM	406 AB	ENG 113 (F)	TF	406 AB	PHY 113 (H)	AT
202	CSE 132 (B)	AH	202	MAT 121 (D)	SP	202	MAT 121 (C)	SP	202	CSE 112 (H)	MOI	202	GED 201 (G)	AJ	202	GED 201 (G)	AJ
203	---	203	CSE 234 (B)	KWN	203	MGT 414 (B)	FM	203	CSE 224 (C)	HR	203	GED 201 (E)	JUS	203	GED 201 (E)	JUS	
501 AB	---	---	501 AB	CSE 132 (C)	HK	501 AB	Booked	AH	501 AB	CSE 234 (A)	KWN	501 AB	MGT 414 (A)	FM	501 AB	---	---
504 AB	CSE 214 (B)	KWN	504 AB	---	---	504 AB	CSE 413 (C)	JM	504 AB	GED 201 (C)	JUS	504 AB	CSE 112 (C)	MOI	504 AB	CSE 413 (A)	JM
601 AB	---	---	601 AB	CSE 432	APC	601 AB	CSE 214 (C)	ABMM	601 AB	STA 133 (A)	RA	601 AB	CSE 224 (A)	HR	601 AB	ENG 113 (C)	TF
604 AB	---	---	604 AB	CSE 112 (B)	MOI	604 AB	GED 201 (B)	JUS	604 AB	ENG 113 (B)	TF	604 AB	Booked	NRC	604 AB	Booked	NRC
306 AB	---	---	306 AB	MGT 414 (B)	FM	306 AB	STA 133 (B)	RA	306 AB	CSE 134 (D)	MTR	306 AB	MAT 121 (A)	SP	306 AB	PHY 113 (F)	MKA
Sunday																	
08:30:00			10:00:00			11:30:00			01:00:00			02:30:00			04:00:00		
Room	Course	Initial	Room	Course	Initial	Room	Course	Initial	Room	Course	Initial	Room	Course	Initial	Room	Course	Initial
L-305AB	CSE 122 (D)	SN	L-305AB	CSE 212 (F)	SN	L-305AB	CSE 112 (A)	MOI	L-305AB	CSE 122 (B)	RS	L-305AB	CSE 314 (B)	NRC	L-305AB	CSE 314 (B)	NRC
L-301AB	CSE 215 (H)	SMH	L-301AB	CSE 215 (H)	SMH	L-301AB	CSE 232 (B)	NNM	L-301AB	CSE 312 (C)	MTR	L-301AB	CSE 312 (C)	MTR	L-301AB	CSE 421L (A)	NNM
L-304AB	---	---	L-304AB	CSE 135 (D)	MTR	L-304AB	CSE 135 (D)	MTR	L-304AB	PHY 113 (A)	SSK	L-304AB	CSE 312 (A)	AR	L-304AB	CSE 312 (A)	AR
406 AB	GED 201 (A)	JUS	406 AB	MAT 111 (F)	SF	406 AB	CSE 431	MFH	406 AB	MAT 111 (G)	SP	406 AB	PHY 113 (E)	SKN	406 AB	CSE 112 (D)	MOI
202	CSE 122 (A)	RS	202	PHY 123 (B)	KI	202	PHY 123 (A)	KI	202	CSE 323 (A)	AHMSI	202	CSE 322 (B)	RR	202	CSE 233 (B)	RR
203	MAT 131 (D)	MJU	203	PHY 123 (D)	SSK	203	MAT 211 (F)	MPL	203	CSE 321 (B)	NN	203	CSE 212 (A)	FNN	203	CSE 412 (C)	ABMM
501 AB	---	---	501 AB	CSE 231 (B)	NNM	501 AB	CSE 311 (B)	MW	501 AB	ENG 123 (A)	NJ	501 AB	CSE 411 (C)	SR	501 AB	CSE 411 (A)	FNN
504 AB	MAT 111 (C)	SF	504 AB	CSE 313 (A)	NRC	504 AB	CSE 214 (E)	MMH	504 AB	GED 201 (H)	SKC	504 AB	CSE 214 (F)	MMH	504 AB	CSE 112 (E)	SR
601 AB	CSE 134 (B)	BH	601 AB	MAT 131 (B)	MJU	601 AB	MAT 211 (A)	BCD	601 AB	CSE 132 (A)	JI	601 AB	CSE 331	RH	601 AB	---	---
604 AB	MAT 211 (E)	MPL	604 AB	CSE 112 (A)	MOI	604 AB	CSE 134 (A)	SAH	604 AB	CSE 214 (A)	SAH	604 AB	CSE 421 (A)	NNM	604 AB	CSE 332	SAH
306 AB	MAT 111 (H)	SP	306 AB	CSE 322 (A)	MFH	306 AB	ENG 123 (B)	NJ	306 AB	CSE 412 (A)	SR	306 AB	CSE 214 (C)	ABMM	306 AB	MAT 211 (C)	BCD

Figure 4.2.2: Home Page (Show Routine)

The given figure 4.2.3 in the following shows about page where some information about us is stored along with the information of our project supervisor Syed Akter Hossain. Here one will find information along with a picture which shows the person about whom we are talking.

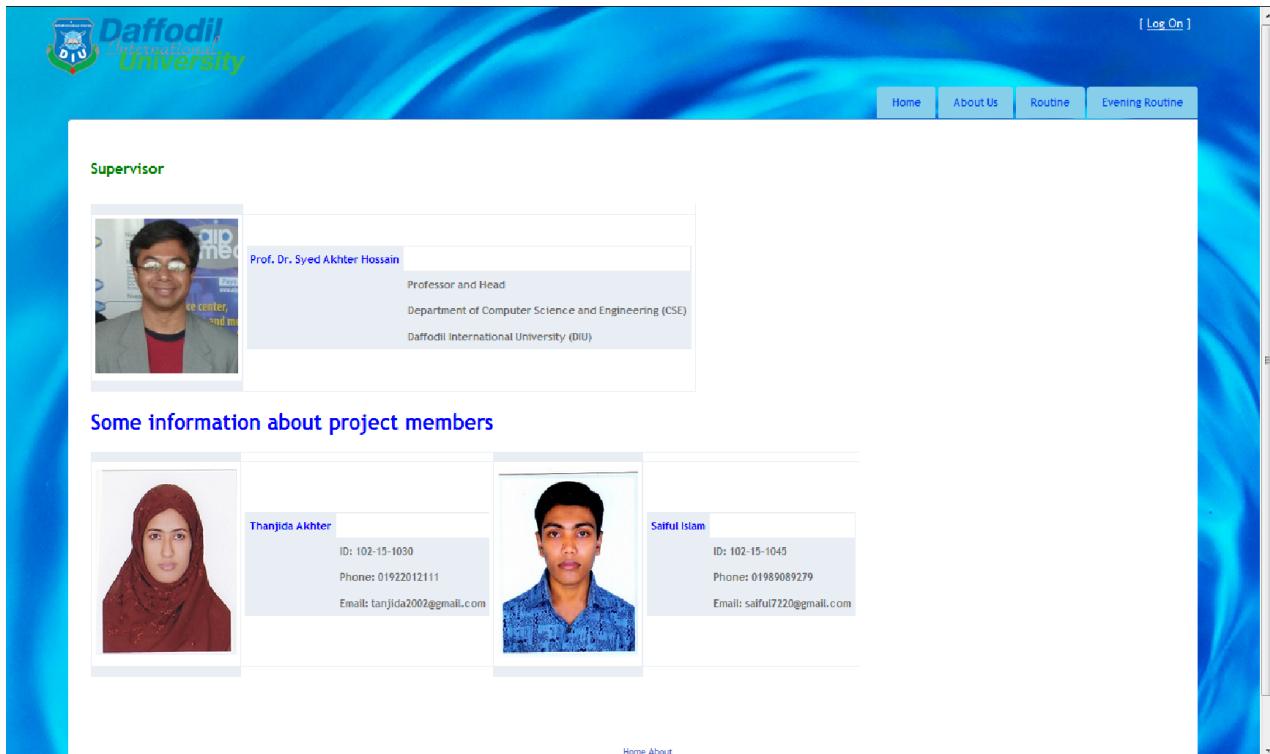


Figure 4.2.3: About Page

The given figure 4.2.4 was user should log in to the system using his or her password. We have a total of two kinds of user. They are Admin and teacher. Admin can add as many user roles as he wants. So one can update the system for different kinds of user if he thinks it is necessary. We have also added different view for different kinds of user and their accessibility is also restricted as the system required.

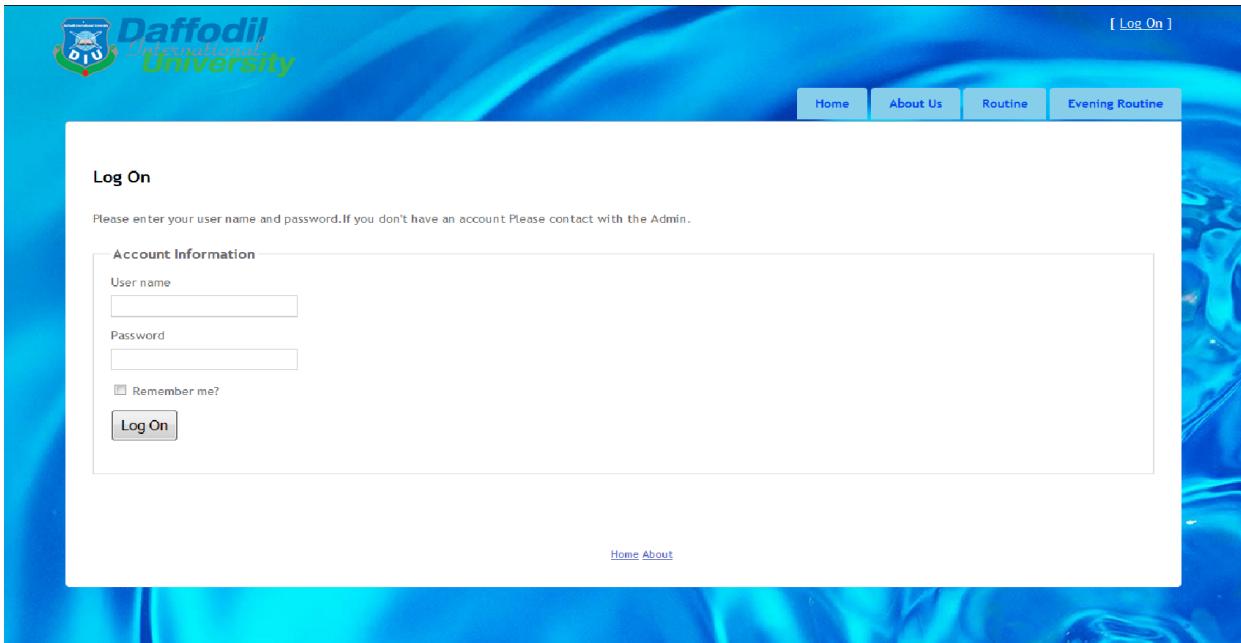


Figure 4.2.4: Login Page

Figure 4.2.5 shows the initial view of the teacher's page. This view is only for the admin user. Here one admin can filter the teacher list with teacher initial. He can see the details of a teacher, edit one or delete one if he thinks it is essential. He will find two more options here. One is to create a single teacher and the other one is to create multiple teachers at a single time. These two processes are described in the later sections.

The screenshot shows the 'All teacher list' page for the admin user. At the top right is a 'Welcome at [Log Off]' message. Below the header is a navigation bar with ten tabs: Home, About Us, Routine, Evening Routine, Empty Slots, Upload a Routine, Other Managements, Reports, and Change Password. The main content area is titled 'All teacher list' and includes two 'Create' buttons: 'Create single New' and 'Create multiple New'. It features a dropdown menu for filtering by teacher initial ('Initial: AH') and a 'Filter' button. A table displays teacher data with columns: Name, Initial, Designation, and Department. The first row shows: ---, AH, Unknown, Unknown, with 'Edit | Details' links. At the bottom of the page are links for 'Home' and 'About'.

Figure 4.2.5: Teacher Page

Figure 4.2.6 shows how to add a single teacher. This view is also available for an admin user. We have here four fields to create a teacher; they are name, initial, designation and department. Initial field is a must here.

The screenshot shows a web application interface for Daffodil International University. At the top, there is a navigation bar with links for Home, About Us, Routine, Evening Routine, Empty Slots, Upload a Routine, Other Managements, Reports, and Change Password. On the right side of the header, it says "Welcome a! [Log Off]". Below the header, there is a sub-navigation menu titled "Create" under the heading "Teacher". This menu includes fields for Name (text input), Initial (text input), Designation (text input), Department (dropdown menu), and a "Create" button. At the bottom of the form, there is a link "Back to List". At the very bottom of the page, there are links for "Home" and "About".

Figure 4.2.6: Teacher (Create a single Teacher)

Figure 4.2.7 shows how to create multiple teachers at a time. Only admin can access this page. He can give information about the teacher as the page has shown or copy data from a excel sheet and paste in the text box. If the given format is correct then the system will generate the teachers as the admin has given.



Welcome a! [Log Off]

[Home](#) [About Us](#) [Routine](#) [Evening Routine](#) [Empty Slots](#) [Upload a Routine](#) [Other Managements](#) [Reports](#) [Change Password](#)

Give Teacher's information as following format

Name [tab] Initial [tab] Designation [tab] Dept. Code

Or

You can copy data from excel sheet and paste in the box.

[Save](#)

Figure 4.2.7: Teacher (Create multiple Teacher at a time)

From this page the admin can control everything about courses. He can create, update or delete a course or a multiple at a time. Figure 4.2.8 shows the initial page of the course.



Welcome a! [Log Off]

[Home](#) [About Us](#) [Routine](#) [Evening Routine](#) [Empty Slots](#) [Upload a Routine](#) [Other Managements](#) [Reports](#) [Change Password](#)

All information about course

[Create Single New](#)

[Create Multiple New](#)

CourseCode:

Name	Code	Credit	Level	Term	Department
...	CSE112	0	0	CSE	Edit Details

[Home](#) [About](#)

Figure 4.2.8: Course (Initial View)

Figure 4.2.9 shows how to add a single course. To create a single course the admin have to provide six data fields. Course name, credit, code, level, term and the department based on which the course will be created.

The screenshot shows a web application interface for Daffodil International University. At the top, there is a navigation bar with links: Home, About Us, Routine, Evening Routine, Empty Slots, Upload a Routine, Other Managements, Reports, and Change Password. On the far right of the header, it says "Welcome at [Log Off]". Below the header, there is a main content area titled "Create". This area is specifically for adding a new course. It contains six input fields labeled "Name", "Code", "Credit", "Level", "Term", and "Department". Each field has a corresponding text input box. After the "Department" field, there is a dropdown menu icon. Below these fields is a "Create" button. At the bottom left of the "Create" form, there is a link "Back to List".

Figure 4.2.9: Course (create a single course)

Figure 4.2.10 shows how to add multiple courses at a time. Like the multiple teachers entry this view is only for the admin and he can create courses almost like teachers. Only difference is the data fields are different.



Daffodil
International
University

Welcome a! [Log Off]

Home About Us Routine Evening Routine Empty Slots Upload a Routine Other Managements Reports Change Password

[Back To Course Index](#)

Give Course's information as following format

Name [tab] Code [tab] Credit [tab] Level [tab] Trem[tab] Department

Or

You can copy data from excel sheet and paste in the box.

Figure 4.2.10: Course (create multiple course).

Figure 4.2.11 shows all the classrooms in a list with edit, delete and update options. One can also create a classroom from here. This is also a restricted page. It is only for the admin user.



Daffodil
International
University

Welcome a! [Log Off]

Home About Us Routine Evening Routine Empty Slots Upload a Routine Other Managements Reports Change Password

Index

[Create New](#)

ClassRoomNumber: 306AB

RoomNo	Description
306AB	Admin Building

[Edit](#) | [Details](#)

[Home](#) [About](#)

Figure 4.2.11: Classroom Page

Figure 4.2.12 shows the all the department's list in the University. Admin can control this page through login. He can edit, update or delete a department from here.

Name	Code
Computer Science And Engineering	CSE
Electrical And Electronics Engineering	EEE
Bachelor of Business Administration	BBA
Textile Engineering	TE
Unknown	Unknown

Figure 4.2.12: Department Page

Figure 4.2.13 shows admin can upload a routine. He has to upload a CSV file. Because the system can only work with the CSV file format. On successful data upload it will show a success message and on failure it will show an error message.

Please upload an CSV file and well try to parse it.

No file selected.

Figure 4.2.13: Upload Routine Page

In figure 4.2.14 shows on successful upload it will show something like below. We have shown a message for successful upload and the message shown will always be in green color.

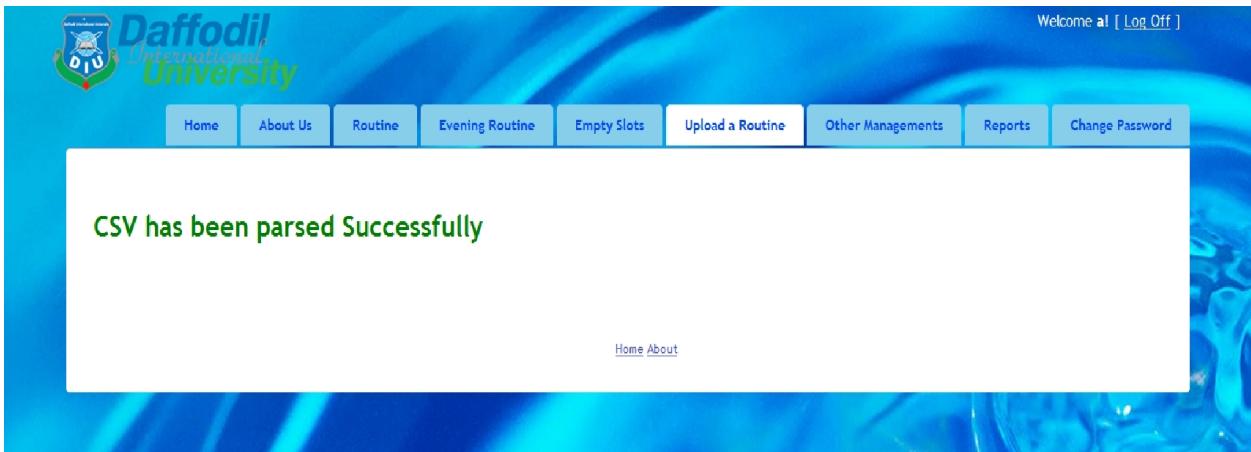


Figure 4.2.14: Upload Successful

On failure the system will generate different kinds of message formats depending on the failure reasons. One of them is like figure 4.2.15. The failure message will always be shown in red color.

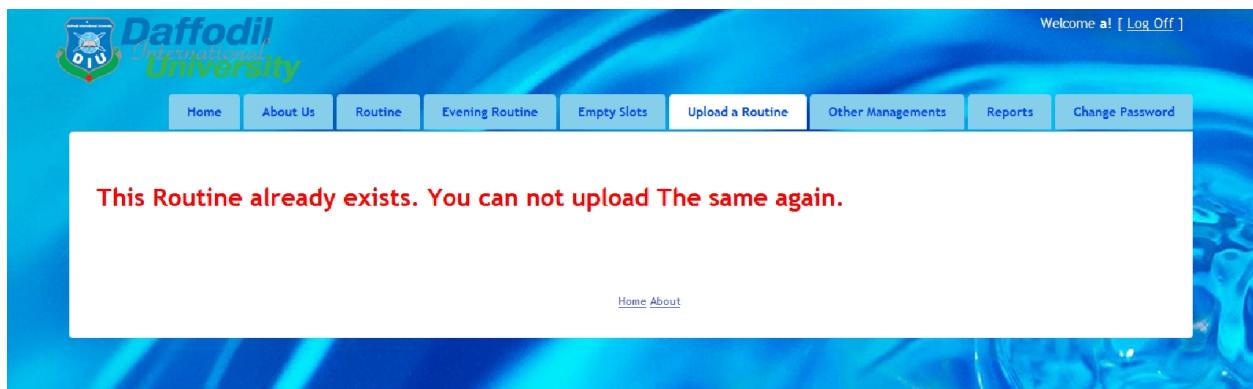


Figure 4.2.15: Upload Failed

This page is designed to view all the routine of the university at a time. For more flexibility we add many filter option with it so that one can find exactly what he wants. Figure 4.2.16 shows the Routine page view for the admin without filter option.

Welcome at! [Log Off]

[Home](#) [About Us](#) [Routine](#) [Evening Routine](#) [Empty Slots](#) [Upload a Routine](#) [Other Managements](#) [Reports](#) [Change Password](#)

List of all departments Routine

[Create New](#)

Department: All RoutineType: All Semester: All Year: All

Day: All Section: All CourseCode: All ClassRoom: All Initial: All [Filter](#)

Day	ClassRoom	Time_From	Time_To	Course	Section	Teacher	Department	Semester	Year	Action
Saturday	501AB	06:00 PM	07:30 PM	CSE323	B	SN	CSE	Spring	2014	Edit Details Delete
Saturday	501AB	07:30 PM	09:00 PM	CSE222	B	MTR	CSE	Spring	2014	Edit Details Delete
Saturday	504AB	06:00 PM	07:30 PM	CSE431	A	JM	CSE	Spring	2014	Edit Details Delete
Saturday	504AB	07:30 PM	09:00 PM	CSE231	A	RS	CSE	Spring	2014	Edit Details Delete
Saturday	601AB	06:00 PM	07:30 PM	CSE331	A	MW	CSE	Spring	2014	Edit Details Delete
Saturday	601AB	07:30 PM	09:00 PM	CSE221	A	SRH	CSE	Spring	2014	Edit Details Delete
Saturday	306AB	06:00 PM	07:30 PM	CSE412		SR	CSE	Spring	2014	Edit Details Delete
Saturday	306AB	07:30 PM	09:00 PM	MGT414	A	TBA	CSE	Spring	2014	Edit Details Delete
Saturday	401AB	06:00 PM	07:30 PM	CSE224	A	MMR	CSE	Spring	2014	Edit Details Delete
Saturday	401AB	07:30 PM	09:00 PM	CSE321	B	MHR	CSE	Spring	2014	Edit Details Delete
Saturday	404AB	06:00 PM	07:30 PM	PHY123	A	AT	CSE	Spring	2014	Edit Details Delete

Figure 4.2.16: Routine Page (without filter)

Figure 4.2.17 shows the Routine page view for all users with filter options. As in the pictures we can say that one can filter a routine with many options. Some of them are departments, semester, teacher initial, course, classroom, day, course code etc.

Welcome at! [Log Off]

[Home](#) [About Us](#) [Routine](#) [Evening Routine](#) [Empty Slots](#) [Upload a Routine](#) [Other Managements](#) [Reports](#) [Change Password](#)

List of all departments Routine

[Create New](#)

Department: CSE RoutineType: Day Semester: Spring Year: All

Day: Saturday Section: All CourseCode: All ClassRoom: L305AB Initial: All [Filter](#)

Day	ClassRoom	Time_From	Time_To	Course	Section	Teacher	Department	Semester	Year	Action
Saturday	L305AB	08:30 AM	10:00 AM	ACM		---	CSE	Spring	2014	Edit Details Delete
Saturday	L305AB	10:00 AM	11:30 AM	ACM		---	CSE	Spring	2014	Edit Details Delete
Saturday	L305AB	11:30 AM	01:00 PM	ACM		---	CSE	Spring	2014	Edit Details Delete
Saturday	L305AB	01:00 PM	02:30 PM	ACM		---	CSE	Spring	2014	Edit Details Delete
Saturday	L305AB	02:30 PM	04:00 PM	ACM		---	CSE	Spring	2014	Edit Details Delete
Saturday	L305AB	04:00 PM	05:30 PM	ACM		---	CSE	Spring	2014	Edit Details Delete

[Home](#) [About](#)

Figure 4.2.17: Routine Page (with filter)

Figure 4.2.18 is a strictly restricted view which can only be accessed by the admin user. From here he can create new user for the system. He can also add or remove a role to a user from this page.

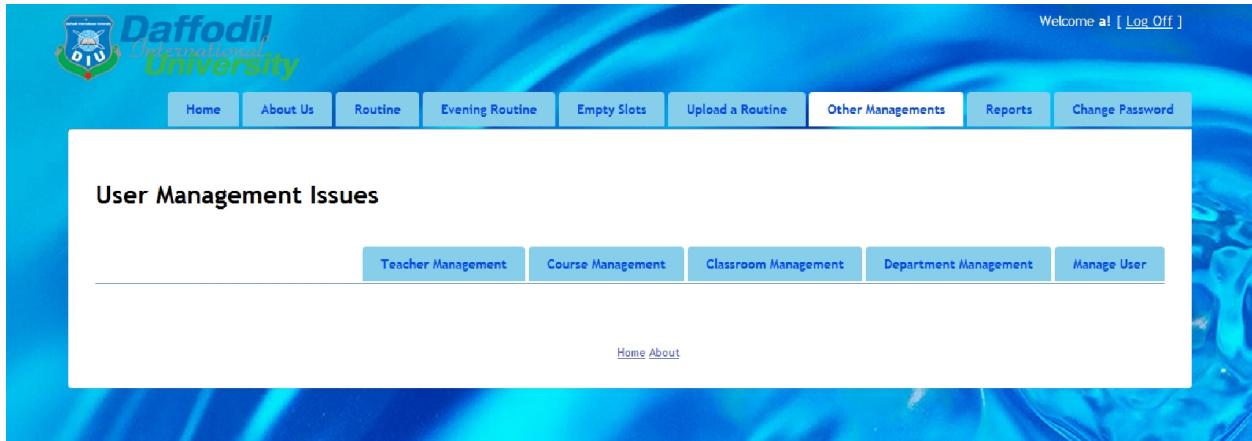


Figure 4.2.18: User Management

4.2.19 is a web page figure of the system which has been design especially for the teachers. A teacher can book a single room in a defined day with a single time slot. This process can only be done if the room is empty at the time which will be given by a teacher.

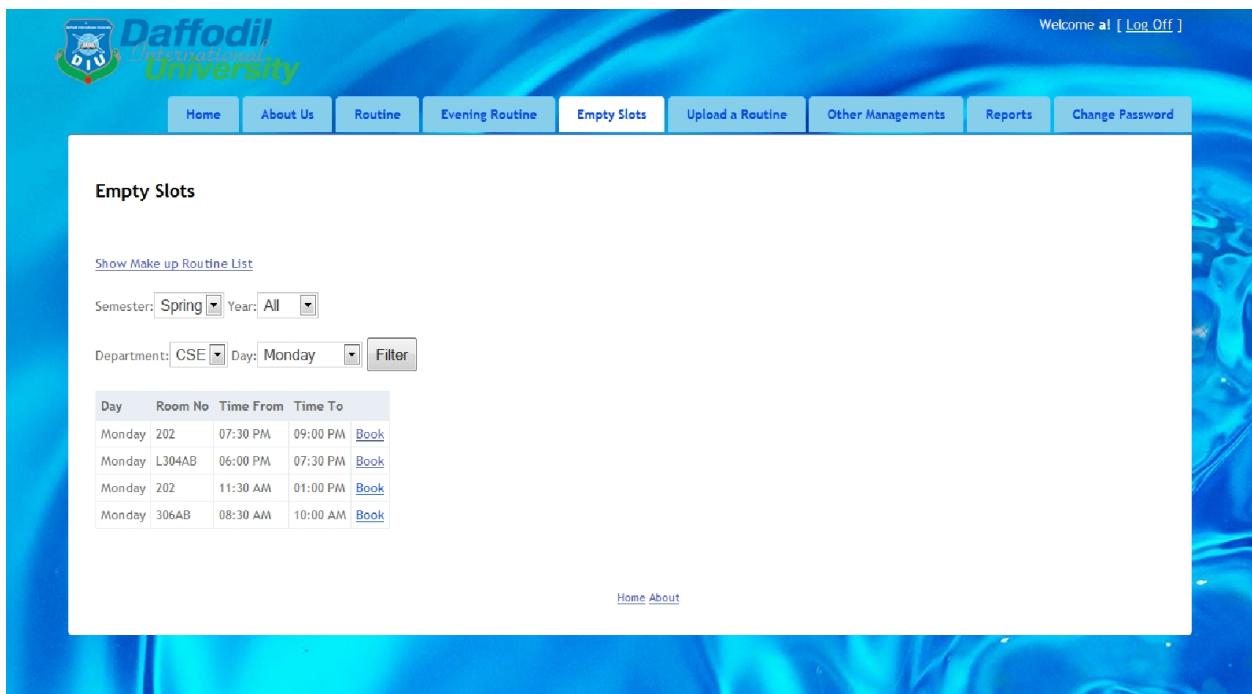


Figure 4.2.19: Empty Slots

4.3 System Implementation

“Integrated Routine Management System” project is implemented on .Net platform using MVC3 architecture where C# is used as font end language along with CSS, java script, HTML and MSSQL as the back end database. The system is developed as per the requirement of the Daffodil International University. Different department routine has been integrated here in a common side for the future benefit of the University. The system has also been kept as simple as possible for the end user’s flexibility. The implementation of this project has taken a long period of four months after the completion of the requirement collection.

4.4 Integration and Testing

After the implementation the project went through several testing process where the system was integrated again if any test failed. Some of the testing process is given below.

Figure 4.4.1 shows the login process testing where Username or Password is not correct.

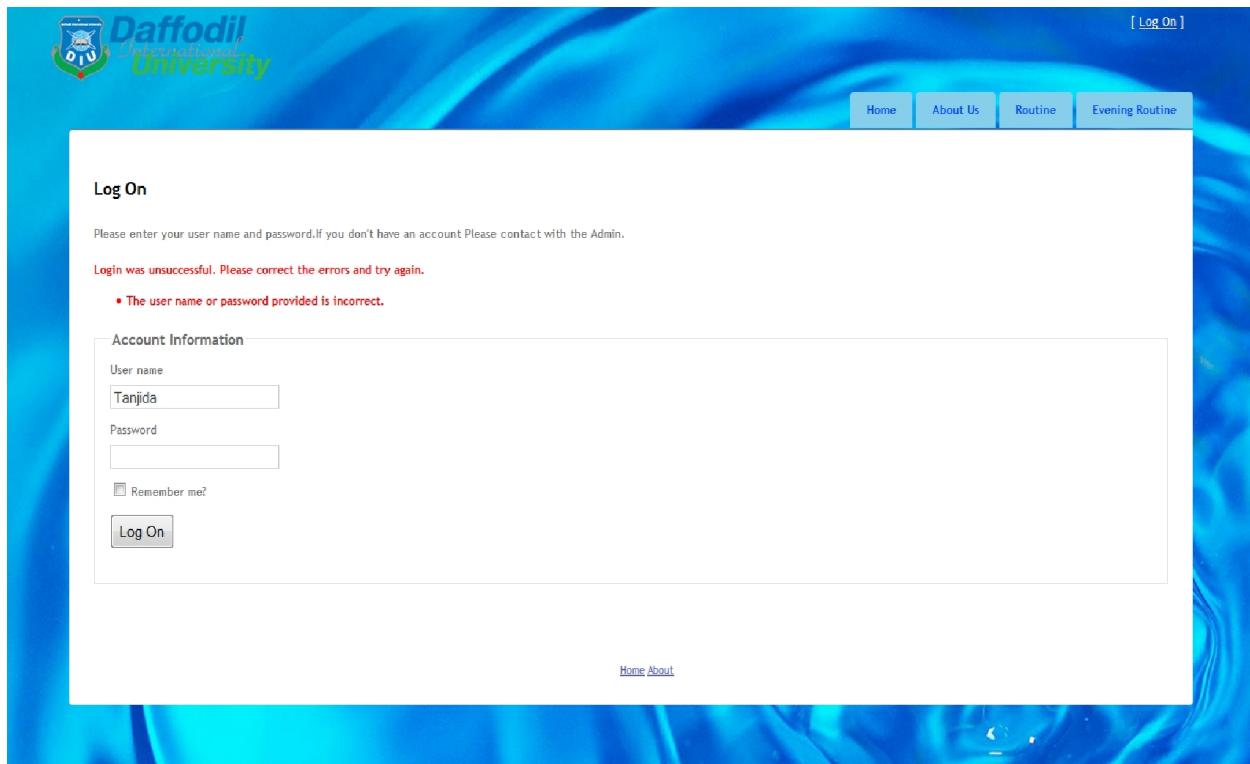


Figure 4.4.1: Login Testing

Figure 4.4.2 shows the upload testing process where an existing routine was uploaded again.

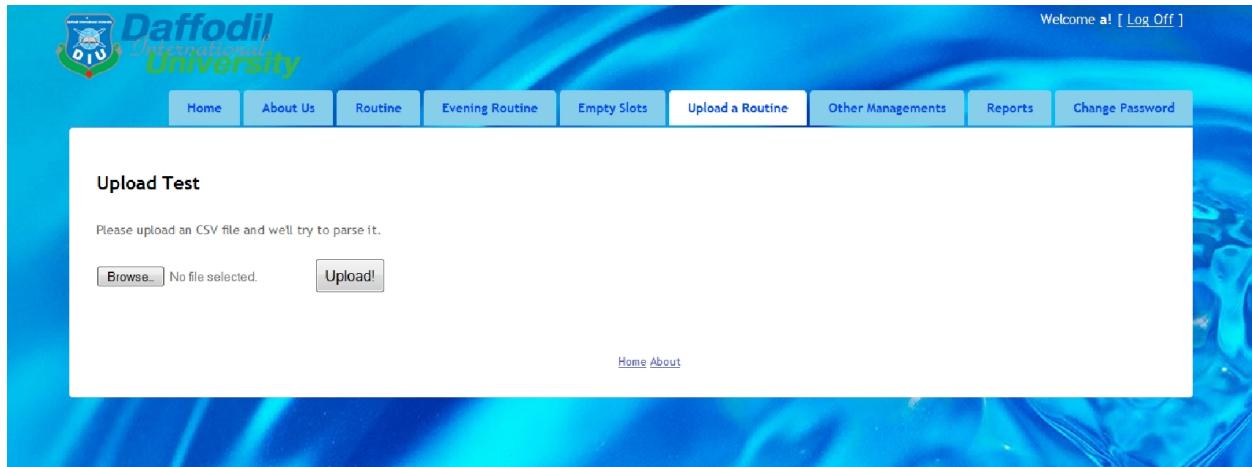


Figure 4.4.2: Upload Testing

Figure 4.4.3 shows the upload testing process where an existing routine was uploaded again and its shows the following image.

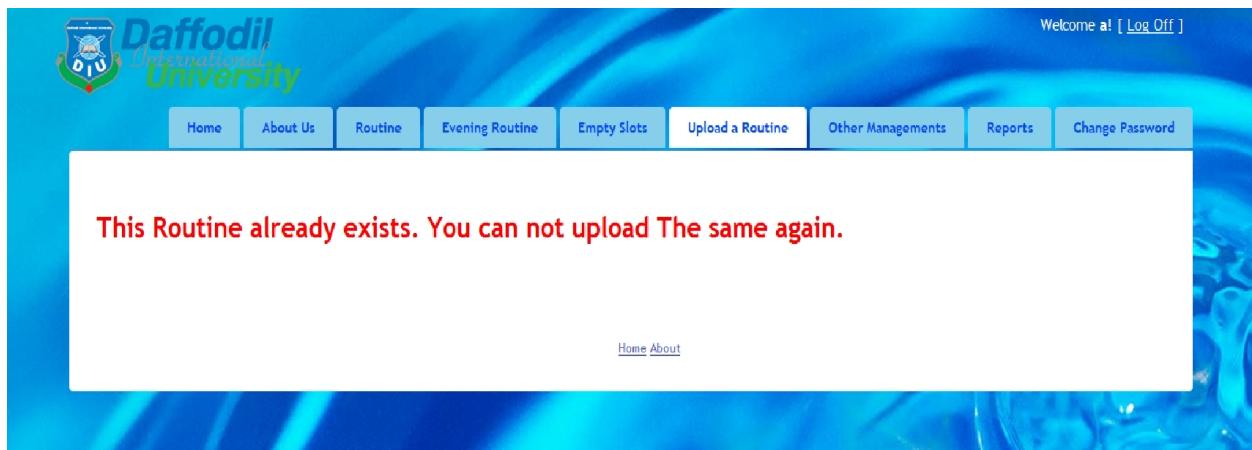


Figure 4.4.3: Upload Testing-1

Figure 4.4.4 shows the upload testing process where a file is uploaded which is not a CSV file and its shows the following image.

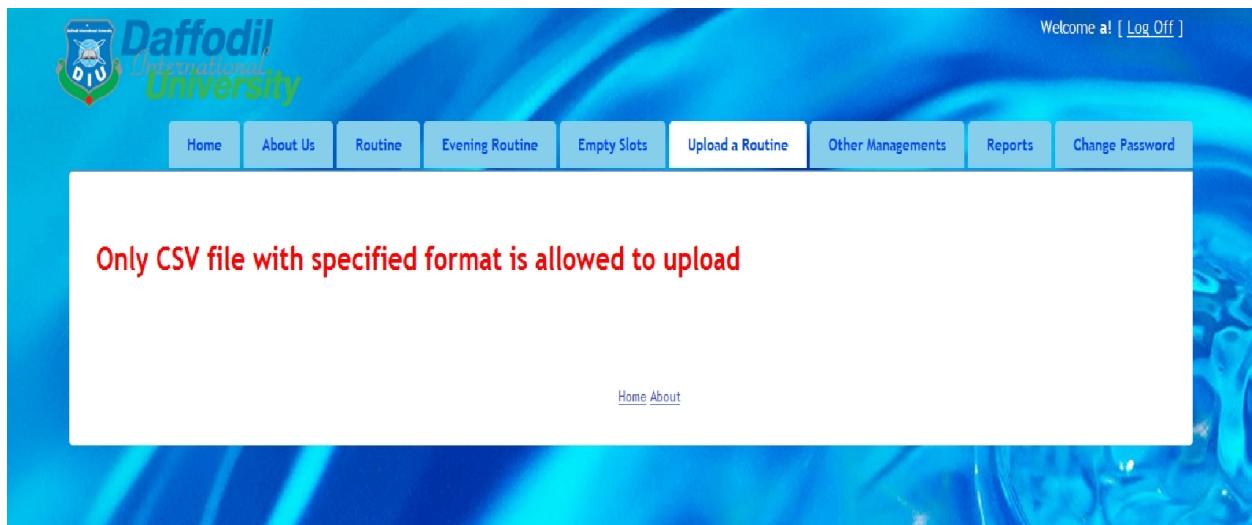


Figure 4.4.4: Upload Testing-2

Figure 4.4.3 shows the testing of the teacher creation where the Department of the teacher was not given.

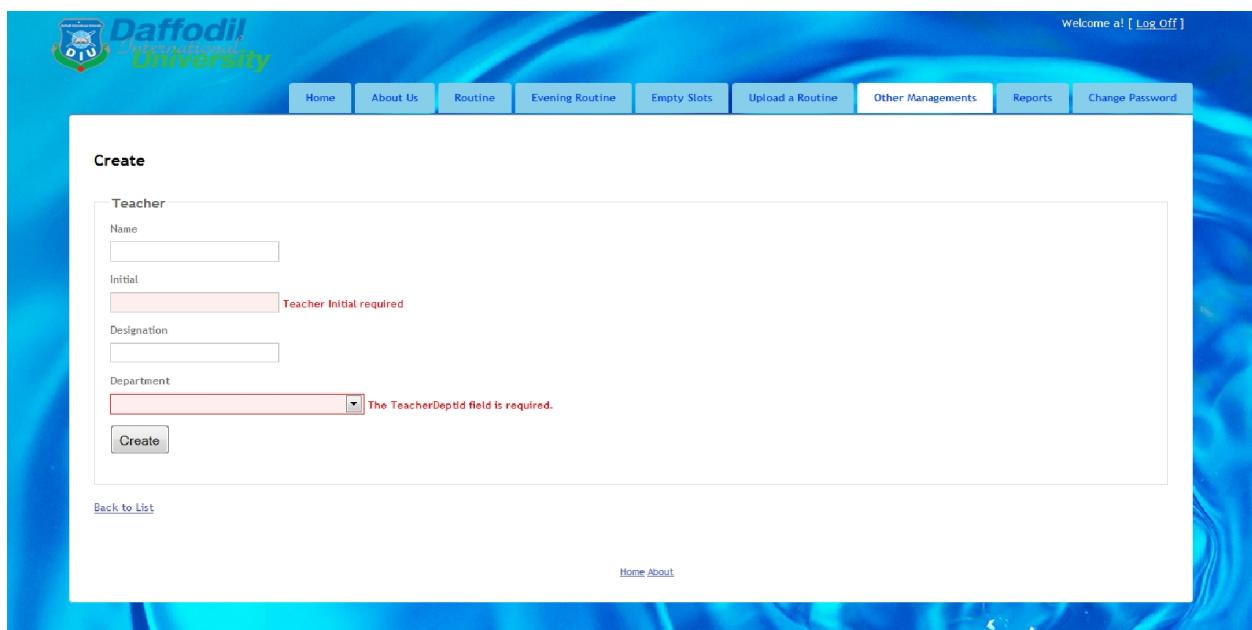


Figure 4.4.5: Teacher Create Test

Figure 4.4.4 shows the testing of the course creation where the Department of the course was not given.

The screenshot shows a web application interface for 'Daffodil International University'. At the top, there is a navigation bar with links: Home, About Us, Routine, Evening Routine, Empty Slots, Upload a Routine, Other Managements, Reports, and Change Password. On the right side of the header, it says 'Welcome al [Log Off]'. Below the header, there is a 'Create' button followed by a 'Course' section. The 'Course' section contains fields for Name, Code, Credit, Level, Term, and Department. The 'Code' field is empty and has a red border, with the error message 'Course Code required' displayed below it. The 'Department' field is also empty and has a red border, with the error message 'The CourseDeptId field is required.' displayed below it. There is a 'Create' button at the bottom of the form. At the very bottom of the page, there are links for 'Home' and 'About'.

Figure 4.4.6: Course Create Test

Similarly every process of the project went through multiple testing to ensure the correctness of the project. Each time the project has been integrated with the current System to ensure better quality.

CHAPTER 5

CONCLUSION AND FUTURE

5.1 Discussion and Conclusion

“Integrated Routine Management System” is developed for increasing flexibility of the current Routine system. It has been developed to help students and teacher to cooperate with the routine system more easily, to update the routine as per necessity and to find the updates in the routine at the quickest. This system has many strong points like

- Login process to verify the admin.
- Find necessary information about routine without login.
- Manage the routine more easily.
- Find the updates of routine when it is updated.
- Minimum input is required to work the system.
- Consumes less time to manage the Routine System.

At the beginning of this project we thought that it is going to be very helpful for all the students and teachers. But as time progress we found something more interesting. It is much more helpful for the administration. It is far more powerful information and management system then one can ever imagine hearing the name of routine management system. This system can be used to control an educational system in many ways that we have never thought about. It can also be used as a powerful tool to make the system a fully digitalized one. After working with this project none of us can wait any longer to build a better and more interactive Routine Information Management System as a developed version of this one which will have many more uses then the current project.

5.2 Future Scope

There are many scopes for this project in the future. Some of them are

- In the future this project will be integrated with the University's Social web System.
- MySql database will be used as the back end database.
- In addition of this system a Mobile application will be developed for only search option.
- Enhancing the capability and usability
- Upgrade functionality according to user feedback
- Applying genetic algorithm to generate routine automatically

REFERENCES

- [1] Microsoft Developer network. library. [Online]. Available: <http://msdn.microsoft.com/library>, Last Modified: 18 April 2014, 02:47pm
- [2] stackoverflow.asp.net.mvc[Online]. Available:<http://stackoverflow.com/questions/tagged/asp.net-mvc>, Last Modified: 14 April 2014, 01:47am
- [3] codeproject.C#. [Online]. Available <http://www.codeproject.com/?cat=3>, Last Modified: 17 April 2014, 12:47pm
- [4] Codeproject..NET. [Online]. Available <http://www.codeproject.com/?cat=24>, Last Modified: 26April 2014, 10:10am
- [5]Microsoft.ASP.NET. Membership and Authorization. [Online]. Available:<http://www.asp.net/mvc/tutorials/mvc-music-store/mvc-music-store-part-7>, Last Modified: 20April 2014, 9:10pm
- [6] Cineplex. ASP.NET MVC3 Site Template.[Online]. Available: <http://mvc3sitetemplate.codeplex.com/>, Last Modified: 28April 2014, 12:10am