



# **Daffodil International University**

Department of Software Engineering, FSIT

## **SWE-431 Project / Thesis**

**Project Documentation**

**Exam hall Seating allotment system**

**Supervised by: Md. Habibur Rahman**

Senior Lecturer Department of Software Engineering

Daffodil International University **Submitted by: Khalequzzaman Miraj 161-**

**35-1548**

Department of Software Engineering  
Daffodil International University

## **Approval**

This **Project** titled “**Exam hall seating allotment system**”, submitted by **Khalequzzaman miraj**, ID-161-35-1548 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc in Software Engineering and approved as to its style and contents.

### **BOARD OF EXAMINERS**

-----  
**Dr. Touhid Bhuiyan**  
**Professor and Head**

Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

**Chairman**

-----  
**Md Fahad Bin Zamal**  
**Associate Professor**

Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

**Internal Examiner 1**

-----  
**Md. Shohel Arman**  
**Lecturer**

Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

**Internal Examiner 2**

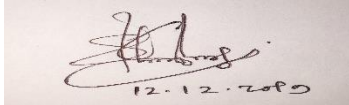
-----  
**Prof Dr. Mohammad Abul Kashem**  
**Professor**

Department of Computer Science and Engineering  
Faculty of Electrical and Electronic Engineering  
Dhaka University of Engineering & Technology, Gazipur

**External Examiner**

## DECLARATION

We hereby declare that we have taken this thesis under the supervision of **Md. Habibur Rahman, Lecturer, Department of Software Engineering, Daffodil International University**. We also declare that neither this project nor any part of this has been submitted elsewhere for award of any degree.

A rectangular box containing a handwritten signature in black ink. The signature is cursive and appears to read 'Khalequzzaman miraj'. Below the signature, the date '12.12.2020' is written in a smaller, simpler font.

**Khalequzzaman miraj**

ID: 161-35-1548

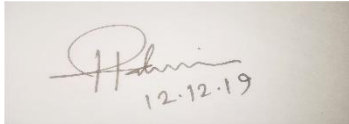
Batch: 19<sup>th</sup>

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Certified by:

A rectangular box containing a handwritten signature in black ink. The signature is cursive and appears to read 'Md. Habibur Rahman'. Below the signature, the date '12.12.19' is written in a smaller, simpler font.

**Md. Habibur Rahman**

Senior Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

## Acknowledgement

First of all, thanks to **ALLAH, ALHAMDULILLAH** because I am very much blessed as I have successfully reached towards the final semester. From the very beginning of my university life, I have learned a lot about software engineering as well as core computer science related knowledge from my course teachers. Moreover, they teach us ethics, morality and politeness.

It is a fortunate opportunity for me as a student of the Department of Software Engineering, one of the exalted academic centers of the Science and Information Technology Faculty of the Daffodil International University, to express my deep feelings of gratitude to the department and to my honorable teachers and also to the department staff.

Besides, I am so much thankful to my parents that, I was always supported by them. They always prioritize my opinion and inspired me as well.

I am also so many grateful to my supervisor **Md. Habibur Rahman**, Senior Lecturer, Department of Software Engineering, FSIT. Daffodil International University, Dhaka, for his excellent guidance, inspiration, encouragement and also for allowing me to work with this project. It was not possible for me to complete my thesis paper successfully without his help.

Specially, I am very thankful to my friend, elder and younger brothers of our university who always appreciate me to my studies and my daily works for learning new ones.

## **Abstract**

I have started to develop this project from April. From the very starting of development of this project, it demands hard working, patients, persistency to meet the requirements of stakeholders. After that I have proposed the design. And then started to work.

Database plays a vital role for any application software. And so this is why, I have designed the database diagram having tables with proper relationship. After that, I have prepared the user interface and take their approval to continue to the next part. It is to be said that, the interface of my application is very simple and easy to understand. After completing that, I have started to write the core functionality of the project.

Developing project is not end of all tasks actually. There are some other important tasks to perform. And that is testing. It is also known as quality assurance also. Almost at every software company there are a quality assurance team. Their main responsibility is to find the loop holes or vulnerability of software. If there any bug remains before handover to the stakeholders, there is a change to ruin the whole project. So testing plan is very important. And after developing the project, I have tried to assure the quality of this project.

<b>Chapter1.....</b>	<b>1</b>
<b>1.1Project Overview .....</b>	<b>2</b>
<b>1.2Project Purpose .....</b>	<b>2</b>
<b>1.2.1Background.....</b>	<b>2</b>
<b>1.2.2Benefits &amp; Beneficiaries .....</b>	<b>3</b>
<b>1.2.3Goals.....</b>	<b>3</b>
<b>1.3Stakeholders.....</b>	<b>3</b>
<b>1.4Project Schedule .....</b>	<b>4</b>
<b>1.6Gantt Chart.....</b>	<b>4</b>
<b>1.7Release Plan or Milestone.....</b>	<b>5</b>
<b>Chapter2.....</b>	<b>6</b>
<b>2.1Functional Requirements.....</b>	<b>7</b>
<b>2.1.1Create account.....</b>	<b>7</b>
<b>2.1.2Manager account .....</b>	<b>7</b>
<b>2.1.3Manager student info .....</b>	<b>8</b>
<b>2.1.4Add departments .....</b>	<b>8</b>
<b>2.1.5Add exam hall.....</b>	<b>8</b>
<b>2.1.6Arrange seat info.....</b>	<b>9</b>
<b>2.1.7Manage profile.....</b>	<b>9</b>
<b>2.1.8view seat arrangements .....</b>	<b>9</b>
<b>2.1 Data requirements .....</b>	<b>10</b>
<b>2.3Performance Requirements.....</b>	<b>10</b>
<b>2.3.1Speed &amp; Latency Requirements.....</b>	<b>10</b>
<b>2.3.2Precision &amp; Accuracy Requirements.....</b>	<b>11</b>
<b>2.3.3Capacity Requirements .....</b>	<b>12</b>
<b>2.4Understandability and Politeness Requirements.....</b>	<b>13</b>
pg. v	
©Daffodil International University	
<b>Requirements.....</b>	<b>14</b>
	<b>2.4.1Accessibility</b>

2.4.2	Training Requirements .....	14
2.4.3	Look and Feel Requirements .....	15
2.5	Appearance Requirements.....	15
2.5.2	Operational and Environmental Requirements.....	16
2.6	Expected Physical Requirements.....	16
2.6.1	Requirements for Interfacing with Adjacent Systems .....	16
2.6.2	Release Requirements .....	17
2.7	Legal Requirements.....	17
2.7.1	Standards Requirements.....	17
<b>Chapter 3.....</b>		<b>18</b>
3.1	Use Case Diagram .....	19
3.1	Create account.....	20
3.1.1	Manage account.....	21
3.1.2	Manage Students.....	22
3.1.3	Add departments.....	23
3.1.4	Add exam hall.....	23
3.1.5	Arrange seat info.....	24
3.1.6	Manage profile.....	25
3.1.7	View seat arrangement.....	25
3.2	Activity diagram.....	26
3.2.1	Manage account.....	26
3.2.2	Add departments .....	27
3.2.3	Add exam hall.....	28
3.2.4	Arrange seat.....	29
3.2.5	Manage profile .....	30
3.2.5	View seat arrangement .....	31

<b>3.3Sequence Diagram.....</b>	<b>32</b>
<b>3.3.1Create account.....</b>	<b>33</b>
<b>3.3.2Manager profile.....</b>	<b>34</b>
<b>3.3.3Manage students.....</b>	<b>35</b>
<b>3.3.4Add Department.....</b>	<b>36</b>
<b>3.3.5Add exam hall.....</b>	<b>37</b>
<b>3.3.6Arrange seat info.....</b>	<b>38</b>
<b>3.3.7Manage profile.....</b>	<b>39</b>
<b>3.3.8View seat arrangement.....</b>	<b>40</b>
<b>Chapter 4.....</b>	<b>41</b>
<b>4.1Development tools and technology.....</b>	<b>41</b>
<b>4.2User Interface Technology.....</b>	<b>42</b>
<b>4.3CSS framework or Bootstrap.....</b>	<b>42</b>
<b>4.4jQuery UI.....</b>	<b>42</b>
<b>4.5Programming language.....</b>	<b>43</b>
<b>4.1.1Implemented tools and platform.....</b>	<b>43</b>
<b>4.1.2Integrated Development Environment.....</b>	<b>43</b>
<b>4.1.3Web Server.....</b>	<b>43</b>
<b>4.1.4Compliance Requirements.....</b>	<b>44</b>
<b>4.2.1Database Server.....</b>	<b>45</b>
<b>Chapter 5.....</b>	<b>46</b>
<b>5.1Testing Features.....</b>	<b>46</b>
<b>5.2Features to be tested.....</b>	<b>46</b>
<b>5.3Testing Strategy.....</b>	<b>46</b>
<b>5.4Test approach.....</b>	<b>47</b>
<b>5.5Black box Testing.....</b>	<b>47</b>
<b>5.1.1Equivalent Class Partitioning.....</b>	<b>47</b>



5.1.2Boundary Value Analysis.....	47
5.1.3White box testing.....	48
5.1.4Pass/Fail criteria.....	48
5.2.1Testing Schedule.....	48
5.2.2Trace Ability Matrix.....	49
5.2.3Testing Environment.....	49
5.2.4Testing Environment.....	50
5.2.5Test Cases.....	50
5.3.1Manage account.....	50
5.3.2Manager student info.....	51
5.3.3Add department.....	52
5.3.4Admin add exam hall.....	53
5.3.5Arrange seat info.....	54
5.3.6Manage profile.....	55
5.3.7View seat arrangement.....	56
Chapter 6.....	57
6.1Home page.....	58
6.2Administrator Log in Page.....	59
6.3Admin main page.....	60
6.4Student login page.....	61
6.5Update profile.....	62
6.6User view seat arrangement.....	63
6.7Update Profile.....	64
6.8View Seat Arrangement.....	65
6.9View Student info.....	66
6.10Search Seat Arrangement.....	67
6.11Add department.....	68
6.12Add exam hall.....	69
6.13Add seat arrangement.....	70

<b>6.14Add student.....</b>	<b>71</b>
<b>Chapter 7.....</b>	<b>72</b>
<b>7.1GitHub Link.....</b>	<b>73</b>
<b>7.2Project Summary.....</b>	<b>73</b>
<b>7.3Limitations.....</b>	<b>73</b>
<b>7.1.1Obstacles and Achievements.....</b>	<b>74</b>
<b>7.1.2Future Scope.....</b>	<b>74</b>
<b>7.1.3References.....</b>	<b>74</b>

# **Chapter 1**

## **Introduction**

## **1.1Project Overview**

This project is developed for the School, college, University to simplify examination hall allotment and seating arrangement manual work. It facilitates to access the examination information of a particular student in a particular class.

The purpose of developing this seating arrangement system is to provide a way to allocate exam hall for each student without any clash. Mostly students are facing many problems for finding the exam hall, so a newly invented concept helps for the staffs to generate their exam hall arrangement easily.

This project also allocate particular invigilator for particular hall. It is also very useful for the School, college, University where the software may generate the hall separation and concerned reports.

Hence manual Excel sheet and paper work is automated based on their Departments .

## **1.2Project Purpose**

The main purpose of this project named “Exam hall allocation system” is to make an automation system which might be helpful for millions of users from different perspective by solving their a few problems. So that’s why we are going to develop such project.

### **1.2.1Background**

Suddenly we notice that, people from our country of different educational institution is facing a lot of problem at the time of their Exam seat arrangement. And there are not any options remaining to escape those problem. Thousands of people are being suffering, but there is not any solution. So considering that, we have brainstormed and thinking that our problem is going to solve their problems.

## 1.2.2 Benefits & Beneficiaries

Our applications would be beneficial for some point of view. Now, I am mentioning those below:

1. Develop software such that everybody working in exam hall allocation system can handle easily.
2. Trainer can store & retrieve data easily. And hence, keeping these major target segments in focus, the system was developed.
3. Report can also provided through print outs.
4. Provide a simpler method to store and access information related to exam hall and students.
5. Provide a simple interface which will be easily used without much training.
6. Reduce paperwork and make all related information accessible easily.

I have also mentioned some benefits as well as beneficiaries. So, I think this application is very much helpful for users.

## 1.2.3 Goals

The main goals of this project is to develop an web based. Our proposed system has four modules. And as we have limited time and resource, so this is why we will only develop the module named “Exam hall seating allotment system”. Because we really believe in quality project. This document is also aimed at specifying requirements of software to be developed but it can also be applied to assist in the selection relation between the different stakeholders. The standard can be used to create software requirements specifications directly or can be used as a model for defining the system requirements.

## 1.3 Stakeholders

There are two types of stakeholders in our total “Exam hall seating allotment system” system. Such as:

<b>Students</b>	Student can view his profile, update his profile and can search his exam seat in exam hall.
<b>Admin</b>	Admin can add departments , add exam hall , seat arrangement ,add students ,update students profile, can view all those thing

## 1.4 Project Schedule

We need to prepare a scheduling plan to complete the project on time. It also refers to make communication with what task need to get done within timeframe.

## 1.5 Gantt Chart

Activities	Weeks->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Planning	Ideas	█	█																										
	Problem definition	█	█	█	█																								
	Proposal planning	█	█	█	█	█																							
Requirements	Requirements specification		█	█	█	█	█																						
	Requirements Analysis			█	█	█	█	█																					
QA-1	Quality Assurance							█																					
System	Sketching								█	█																			
	Design Specification								█	█	█																		
	Database design									█	█	█																	
Mid defense													█	█															
Implementation - 1	Administrator's Part & Authentication									█	█	█	█	█	█	█	█	█											
QA-2	Test case					█																							
Implementation - 2	User's & Student's Part																												
Testing	Unit testing								█		█																		
	Black box Testing																												
Delivery & Final defense	Software Release																												
Scheduled Time		█																											
Buffered Time		█																											

## 1.6 Release Plan or Milestone

<b>Activities</b>	<b>Duration in week</b>	<b>Total week</b>
<b>Brainstorming</b>	Week no. (1,2)	2
<b>Problem identification</b>	Week no. (1,2,3)	3
<b>Requirement specification</b>	Week no. (2,3,4)	3
<b>Requirement analysis</b>	Week no. (3,4,5,6)	4
<b>Sketching</b>	Week no. (7,8,9)	3
<b>Design specification</b>	Week no. (8,9)	2
<b>Database design</b>	Week no. (9,10)	2
<b>Order entry and T&amp;A and Planning</b>	Week no. (9,10,11,12,13)	5
<b>Quality assurance(1,2)</b>	Week no. (7,12,13,14,15)	5
<b>Test case</b>	Week no. (12,13,14,15)	4
<b>Impose case &amp; demerits</b>	Week no. (16,17,18,19,20,21,22)	7
<b>Unit testing</b>	Week no. (22,23,24)	3
<b>Black-box testing</b>	Week no. (25,26)	2
<b>Software release</b>	Week no. (27,28)	2

# **Chapter 2**

## **Software Requirement Specification**



## 2. Requirement Specification

The complete requirement specification based on the elicitation process is described in this section.

### 2.1.1 Functional Requirements:

Functional requirements refer to the functions which are mandatory to the system. Functional requirements must be able to perform on the software system. Now, we are going to mention functional requirements with our project.

#### 2.1.2 Create accounts

<b>Requirements 1</b>	<b>Create account</b>
<b>Description</b>	After confirming a Students create accounts needs to plan for exam hall. For that reason student create account . student all information department name ,student id and password .Student successfully create account
<b>Stakeholders</b>	Students

#### 2.1.3 Manage accounts

<b>Requirements 2</b>	<b>Manage account</b>
<b>Description</b>	After confirming a admin/students manage accounts and searching seat planning manager needs to plan for exam hall. Admin add department, Student information collected. Student View seat plan
<b>Stakeholders</b>	Admin , Students

### 2.1.4 Manage Students info

<b>Requirements 2</b>	<b>Manage students info</b>
<b>Description</b>	After confirming a student all information collect admin .Every information is saved to the database of the application.
<b>Stakeholders</b>	Admin

### 2.1.5 Add Departments

<b>Requirements 4</b>	<b>Add departments</b>
<b>Description</b>	Admin confirming add Departments needs to exam seat plan. Admin login with user id password . Data provided properly . After providing legal user id password admin successfully login to the system
<b>Stakeholders</b>	Admin

### 2.1.6 Add exam hall.

<b>Requirements 5</b>	<b>Add exam hall</b>
<b>Description</b>	Admin confirming add exam hall and manager needs to exam seat plan. Admin provide data of exam hall. Admin successfully to the system .it will be add in database

<b>Stakeholders</b>	Admin
---------------------	-------

### 2.1.6 Arrange seat info.

<b>Requirements 6</b>	<b>Arrange seat info</b>
<b>Description</b>	Admin must be know about the arrange seat properly. submit an arrange info. Admin must to be have total seat plan
<b>Stakeholders</b>	Admin

### 2.1.7 Manage profile

<b>Requirements 7</b>	<b>Manage profile</b>
<b>Description</b>	When Admin arrange seat planning , view seat plan info. Submit an update profile student and admin .Admin manage all account system . Admin department name and room number allotment system
<b>Stakeholders</b>	Admin

### 2.1.8 View seat arrangements.

<b>Requirements 8</b>	<b>View seat arrangements</b>
<b>Description</b>	When Students search for room, view seat plan info, seat plan by then the search result must show.After providing student view the seat plan.Submit to view seat arrangement
<b>Stakeholders</b>	Students

## 2.1 Data Requirements:

For defining data requirements, we need to build the model. we need to focus on some major points.

Such as:

- Update profile
- View exam hall

## 2.1 Data Requirements:

For defining data requirements, we need to build the model. we need to focus on some major points.

Such as:

- Admin can add departments
- Add exam hall seat arrangement
- Add students
- Update students
- Can view all those thing

## 2.3 Performance Requirements:

It is important to sustain performance of software system. To ensure performance, we need to sustain some steps. Now, I will explain some aspects by which we are going to prolong the performance of our project.

### 2.3.1 Speed & Latency Requirements:

<b>SLR-1</b>	<b>Create account</b>
<b>Description</b>	When Student create accounts. student all information department name ,student id and password .Student successfully create account
<b>Stakeholders</b>	Admin

### 2.3.2 Precision & Accuracy Requirements:

<b>PAR-1</b>	<b>View seat Arrangement</b>
<b>Description</b>	When Students search for room, view seat plan info, seat plan by then the search result must show .After providing student view the seat plan .Submit to view seat arrangement
<b>Stakeholders</b>	Students

### 2.3.3 Manage Requirements:

<b>CR-1</b>	<b>The system will handle thousands of data</b>
<b>Description</b>	admin/students manage accounts and searching seat planning manager needs to plan for exam hall. Admin add department, Student information collected. Student View seat plan
<b>Stakeholders</b>	Admin

### Dependability Requirements:

The term dependability is measured based on four dimensions. Such as:

- Availability
- Reliability
- Safety
- Security

**Reliability & Availability Requirements:**

<b>RAR-1</b>	<b>The system must be available on 24 X 7</b>
<b>Description</b>	Our system must be available examination day, The system must be updated exam day
<b>Stakeholders</b>	Admin , Students

**Robustness or Fault-Tolerance Requirements:**

<b>RFT-1</b>	<b>The system handles planning manager access without system errors</b>
<b>Description</b>	More than user manager might hit our application system at a time. All their requests must be handled without any fault.
<b>Stakeholders</b>	Students , Admin

**Safety-Critical Requirements:**

There are no safety-critical requirements in our project.

**Maintainability & Supportability Requirements:**

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

**Maintainability Requirements:**

Not available for this project.

**Supportability Requirements:**

Supportability requirements may have related to some extends. Like:

- Testability
- Maintainability
- Configurability
- Serviceability
- Install ability

**Security Requirements:**

The system security as a requirement is very important. System security requirements should be its functional requirement. System security enforces security of an application system.

- Signing in as Students and Admin.
- Signing out as Students and Admin.
- Handling encrypted passwords.

.

### **Access Requirements:**

<b>AR-1</b>	<b>Application provides security mechanism</b>
<b>Description</b>	Every module is designed in such a way that it only give access to the supported and authenticated users.
<b>Stakeholders</b>	students and admin

### **Integrity Requirements:**

Integrity requirements refers to a security system which ensures an expectation of data quality. For that reason, we will store our user passwords as encrypted format which is impossible to decrypt. It is also called hashed password.

### **Privacy Requirements:**

It is very important to ensure privacy of the system users. Privacy requirements enhances to protect stakeholder's privacy. Planners permitted to get access to those data which are being associated by them or by the organization's employees which can be ensured by the planning manager log in system.

### **Usability and Human-Interaction Requirements:**

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



### **Ease of Use Requirements:**

Our application is easy to use and also easily understandable.

<b>EUR-1</b>	<b>Application must be usable for the end users</b>
<b>Description</b>	This app is enough usable to the planning manager by which he can operate this system easily.
<b>Stakeholders</b>	Students , Admin

### **Personalization and Internationalization Requirements:**

There are not any personalization requirements to our system. This maiden version of our application is only be operated by a specific seat

### **Understand ability and Politeness Requirements:**

It is already said that the application which we are going to develop, is understandable enough. The system provides hints to users whether any error occurred or wrong. By reading those errors planning manager can be able to operate the system easily.

### **Accessibility Requirements:**

Documentation are mainly two types. One is internal documentation which is generally written by the application engineers. It is prepared to make development life cycle easier for the system engineers or system analysts.

<b>UDR-1</b>	<b>The system engineer documentation</b>
<b>Description</b>	To develop our application named Exam hall seat allocation system analysis team as well as documentation team.

<b>Stakeholders</b>	System analysts or software developers
---------------------	--

### **2.4.2 Training Requirements:**

Training requirements involved in after service of any application. It is very necessary to properly train up end users to the system so that they would be capable to operate easily

### **2.4.3 Look and Feel Requirements:**

Look and feel requirements mainly refers how the system will look like and how the user interface of our system will display to the Students.

### **Appearance Requirements:**

<b>AR-1</b>	<b>php of mandatory fields must be bold</b>
<b>Description</b>	<b>The mandatory field's label must be bold and all input fields must have placeholder to make it easier for the users.</b>
<b>Stakeholders</b>	<b>All users of the system</b>

### **2.5 Operational and Environmental Requirements:**

Admin can add Departments, add exam hall, seat arrangement, add students, update students profile, can view all those thing. Students view seat arrangement

### **2.5.2 Expected Physical Requirements:**

There are no expected requirements in our system.

### **2.6.1 Requirements for Interfacing with Adjacent Systems:**

There are no requirements for interfacing with adjacent system for our project. Because it is a desktop and web based system.

### **2.6.2 Release Requirements:**

There are no specific release requirements in our system. Because it is a live project.

### **2.7 Legal Requirements:**

Legal requirements normally refer to the terms and conditions or privacy policy of any organizations. The terms and condition of our application is that, no third party software or person are allowed to engage to use our data for their business purpose.

#### **2.7.1 Standards Requirements:**

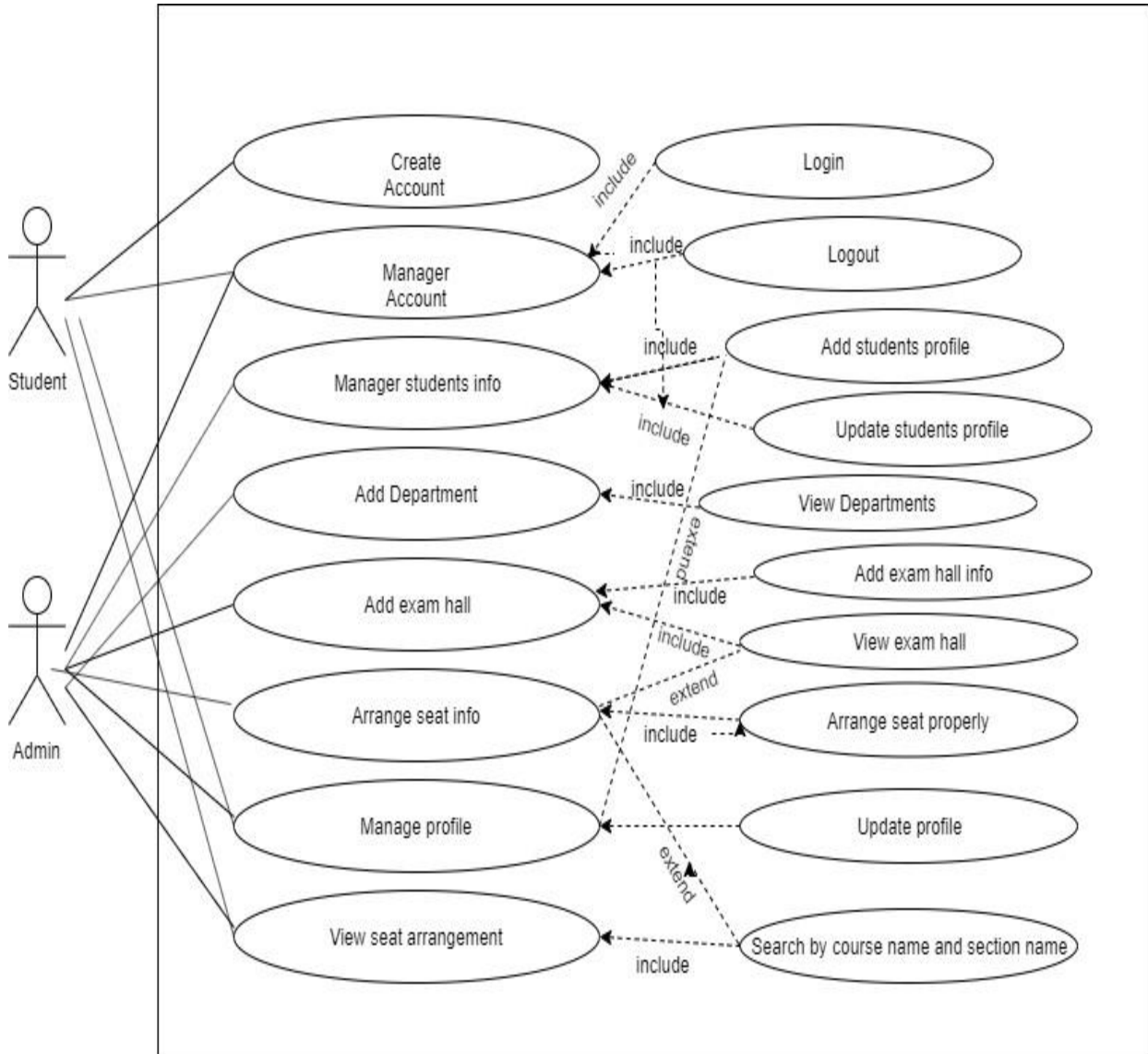
As I said above there are no specific standards requirements for our system.

# **Chapter 3**

## **Requirement Analysis**

### 3.1 Use Case Diagram

We have use case diagram. And there are five actors. Each actor plays different role. And those are already indicated to this use case diagram. This diagram will clarify our system in brief.



### 3.1.1 Create account

<b>Use Case</b>	<b>Create account</b>	
<b>Goal</b>	Successfully create account	
<b>Preconditions</b>	admin must know about students information	
<b>Success End Condition</b>	Admin successfully add students profile Students new accounts created	
<b>Failed End Condition</b>	Admin successfully add students profile can't calculate all date successfully. Students new accounts created can't plan students successfully.	
<b>Primary Actors:</b>	Student	
<b>Secondary Actors:</b>	System	
<b>Trigger</b>	Students must have an create account	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Student name id and department save information create account
	2	Students create account
	3	Student clicks create button for calculating all dates.
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Student name id and department save information can't create account
	2	Students can't search Account
	3	Admin can't click create button for calculating all dates.

### 3.1.2 Manage account

<b>Use Case</b>	<b>Manage account</b>	
<b>Goal</b>	Admin manage all account update profile seat number room number department name all information collected them	
<b>Preconditions</b>	Admin must open the system.	
<b>Success End Condition</b>	Admin manage all account system Admin department name and room number allotment system	
<b>Failed End Condition</b>	Admin manage can't all account system Admin department name and room number allotment can't system	
<b>Primary Actors:</b>	Student and Admin	
<b>Secondary Actors:</b>	System	
<b>Trigger</b>	Student and Admin must have a account for logging in.	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin must open the system
	2	Admin department name and room number allotment system
	3	Admin manage all account system
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Admin doesn't clicks the Admin option
	2	Admin department name and room number allotment can't system
	3	Admin manage can't all account system

### 3.1.3 Manage Students info.

<b>Use Case</b>	Manage Students info
<b>Goal</b>	Admin must about this manage student info properly
<b>Preconditions</b>	Admin manage students info add students profile Admin update students profile.

<b>Success End Condition</b>	Admin can calculate all profile info successfully. Admin successfully update students profile.	
<b>Failed End Condition</b>	Admin can't calculate all machine info successfully. Admin can't plan update students profile successfully.	
<b>Primary Actors:</b>	Admin	
<b>Secondary Actors:</b>	System	
<b>Trigger</b>	Admin manage all students information	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin update student profile option
	2	Students search the exam hall room
	3	Admin clicks plan now button for calculating all students info.
	4	Admin finally clicks plan now button for update Profile
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Admin update can't click the students profile Option
	2	Students can't search the exam hall room.
	3	Admin can't click plan now button for calculating all Students info.
	4	Admin finally can't click plan now button for update profile

### 3.1.4 Add Departments.

<b>Use Case</b>	Add Departments
<b>Goal</b>	Departments add successfully
<b>Preconditions</b>	Add Departments button click. Departments added.
<b>Success End Condition</b>	Admin add the all Departments. Admin can add the departments successfully.
<b>Failed End Condition</b>	Admin can't add the all Departments. Admin can't add the Departments successfully.
<b>Primary Actors:</b>	Admin
<b>Secondary Actors:</b>	System
<b>Trigger</b>	Admin must have an add the Departments



<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin clicks the dyeing option
	2	Admin search the Departments.
	3	Admin clicks plan now button for calculating all Departments info.
	4	Admin finally clicks plan now button for add Departments
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Admin can't click the dyeing option
	2	Admin can't search the Departments.
	3	Admin can't click plan now button for calculating all Departments info.
	4	Admin finally can't click plan now button for add departments

## Add Exam Hall.

<b>Use Case</b>	Add Exam Hall.	
<b>Goal</b>	Add Exam hall successfully	
<b>Preconditions</b>	Add Exam hall button click. Add Exam hall added the room	
<b>Success End Condition</b>	Admin can add exam hall successfully. Admin can seat plan successfully.	
<b>Failed End Condition</b>	Admin can't add exam hall successfully. Admin can't seat plan successfully.	
<b>Primary Actors:</b>	Admin	
<b>Secondary Actors:</b>	System	
<b>Trigger</b>	Admin must have an add the exam hall room	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin clicks the hall room option
	2	Admin search the exam hall room.
	3	Admin clicks plan now button for calculating seat info.
	4	Admin finally clicks seat plan now button for this Room
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Admin can't click the hall room option
	2	Admin can't search the exam hall room.

	3	Admin can't click plan now button for calculating seat info.
	4	Admin can't click seat plan now button for this Room

### 3.1.5 Arrange seat info.

<b>Use Case</b>	Arrange seat info.	
<b>Goal</b>	Arrange seat plan successfully	
<b>Preconditions</b>	Admin button click arrange seat info. Admin seat arrangements information.	
<b>Success End Condition</b>	Admin can arrange seat plan info successfully.	
<b>Failed End Condition</b>	Admin can't arrangement information successfully	
<b>Primary Actors:</b>	Admin	
<b>Secondary Actors:</b>	System	
<b>Trigger</b>	Admin must have an add the arrange seat info	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin clicks the seat info option
	2	Admin search the seat plan.
	3	Admin clicks plan now button for all student seat plan info.
	4	Admin finally clicks the button for Seat info
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Admin can't click the seat info option
	2	Admin can't search the seat plan.
	3	Admin can't click plan now button for all students seat plan info.
	4	Admin finally can't click plan now button for seat Plan

### 3.1.6 Manage profile.

<b>Use Case</b>	Manage profile.
<b>Goal</b>	Profile update successfully
<b>Preconditions</b>	Admin manage system.

	Admin change update profile	
<b>Success End Condition</b>	Admin can update profile successfully. Admin can all information update successfully.	
<b>Failed End Condition</b>	Students can't update profile successfully. Students can't all information update successfully.	
<b>Primary Actors:</b>	Admin	
<b>Secondary Actors:</b>	System	
<b>Trigger</b>	Admin must have an accounts.	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin clicks the update profile option
	2	Students search seat plan .
	3	Admin clicks plan now button for calculating update all information info.
	4	Admin finally clicks plan now button for seat plan
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Admin can't click the update profile option
	2	Admin can't search the seat plan.
	3	Admin can't click plan now button for calculating update profile.
	4	Admin finally can't click plan now button for seat plan

### 3.1.7View seat arrangement.

<b>Use Case</b>	View seat arrangement	
<b>Goal</b>	Arrangements seat successfully	
<b>Preconditions</b>	Seat arrangements button click. Arrangements system poses	
<b>Success End Condition</b>	Student view seat successfully. Student can all seat arrangement successfully.	
<b>Failed End Condition</b>	Student can't view seat successfully. Student can't all view seat successfully.	
<b>Primary Actors:</b>	Student	
<b>Secondary Actors:</b>	System	
<b>Trigger</b>	Student must have an view seat arrangement.	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Student clicks the view seat poses option
	2	Student search view seat arrangements.

	3	Student clicks plan now button for calculating all information info.
	4	Student finally clicks plan now button for view seat Plan
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	1	Student can't click the view seat Option
	2	Student can't search the view seat plan.
	3	Student can't click plan now button for calculating seat plan.
	4	Admin finally can't click plan now button for view seat plan

### 3.2 Activity Diagram

We have prepared some activity diagram according to our use case. These activity diagrams are properly referring the flow of the individual conditions of our project.

#### 3.2.1 Manage account

For any system authorize access is very important. By this a user can secure his data and can avoid another unauthorized user to damage or update the saved data. For our system, Students Admin must log in first. First he should click “Admin/Students” and enter user name and password.

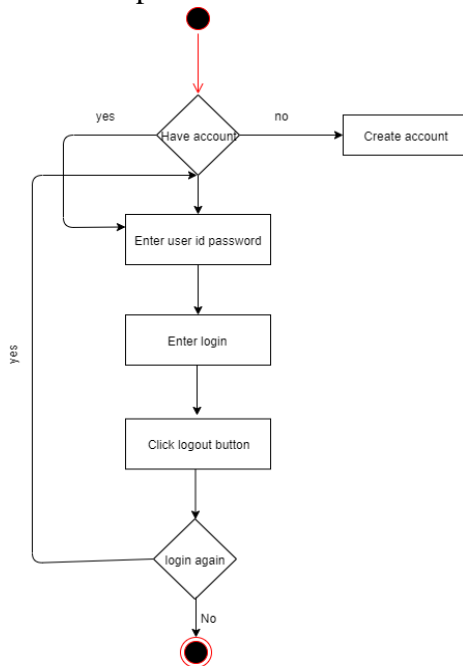


Figure 3.2: Activity diagram for manage account

### 3.2.2 Create account

Secondly students create account. For this Student should click “create account” enter profile info. Then he should click plan now button. After that, he should click “Calculate” button and finally click “Confirm” create account.

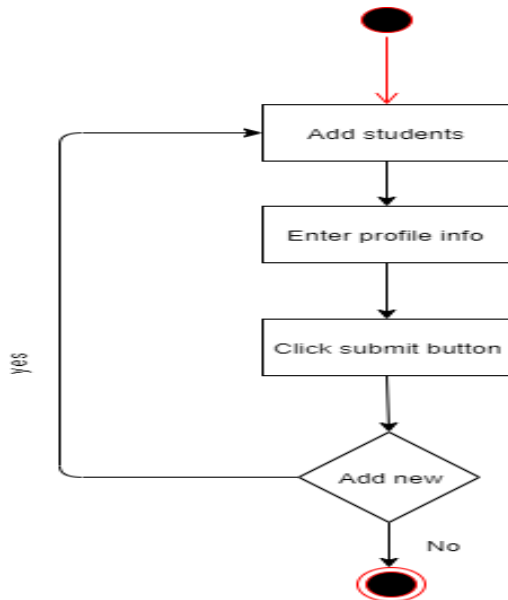


Figure 3.3: Activity diagram of create account

### 3.2.3 Manager Students info

Thirdly Manager Students info add students profile and update students profile. For this manager students should click “update profile” and search info by order id. Then he should click plan now button. After that, he should click “Plan Now” button for manger student info.

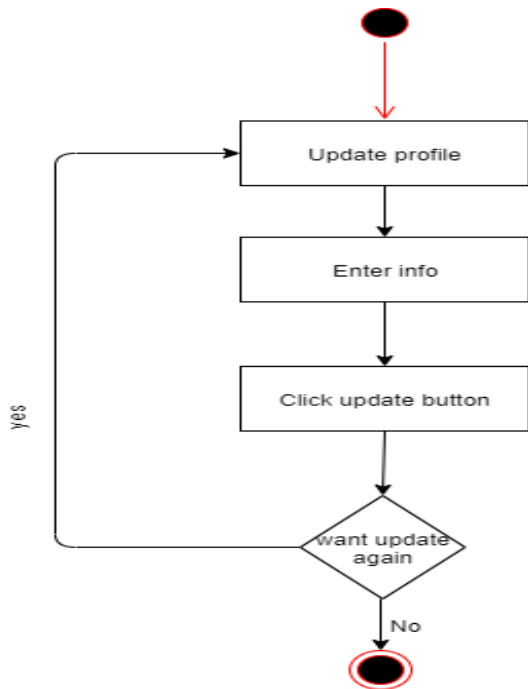


Figure 3.4: Activity diagram of manage student info

### 3.2.4 Add Department.

Fourthly Add department view departments. For this add department click “add department” and enter profile info. Then he should click plan now button. After that, he should click “submit” button for add department.

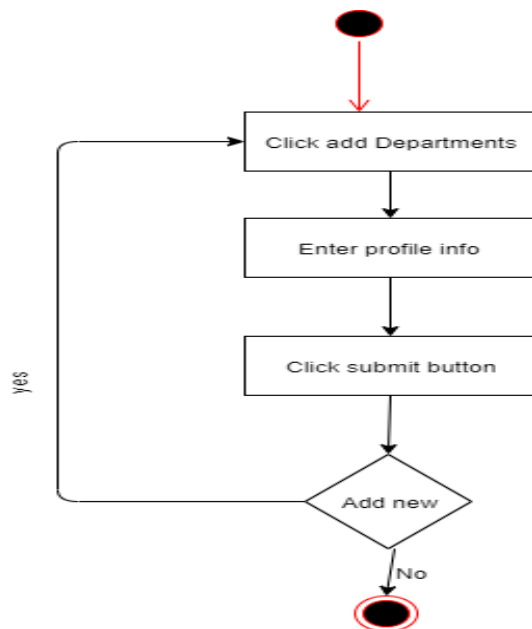


Figure 3.5: Activity diagram Add department

### 3.2.3 Add exam hall

Add exam hall and view exam hall, add exam hall info. For this admin and student “add exam hall” and search enter hall info. Then he should click plan now button. After that, he should click “Plan Now” student and admin.

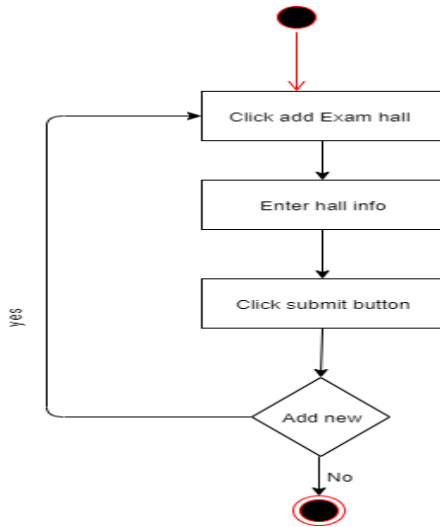


Fig: Activity diagram Add Exam hall

### 3.2.4. Arrange seat info.

Then Admin and student arrange seat properly. For this admin and student should click “Arrange seat” and search info by order id. Then he should click plan now button. After that, he should click “Plan Now” student and admin.

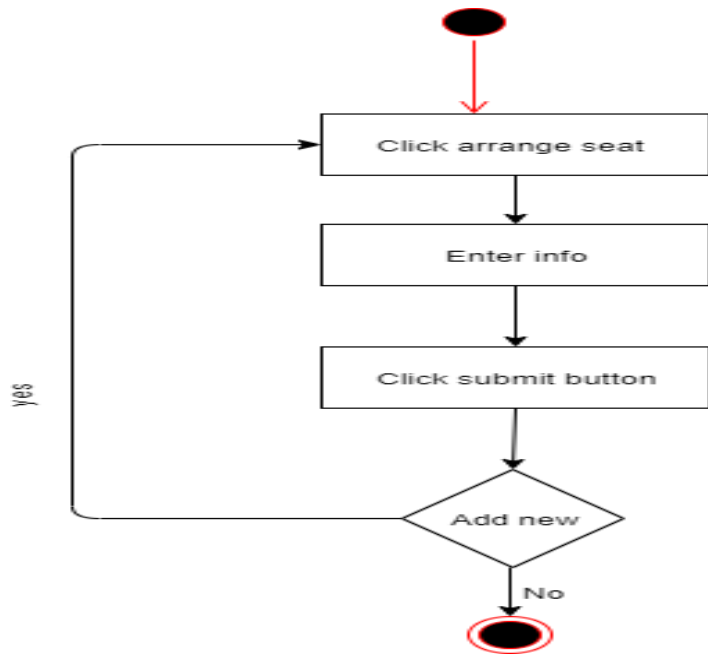


Figure 3.6: Activity diagram of Arrange seat info

### 3.2.5 Manage profile.

For any system authorize access is very important. By this a user can secure his data and can avoid another unauthorized user to damage or update the saved data. For our system, Students Admin must log in first. First he should click “Admin/Student



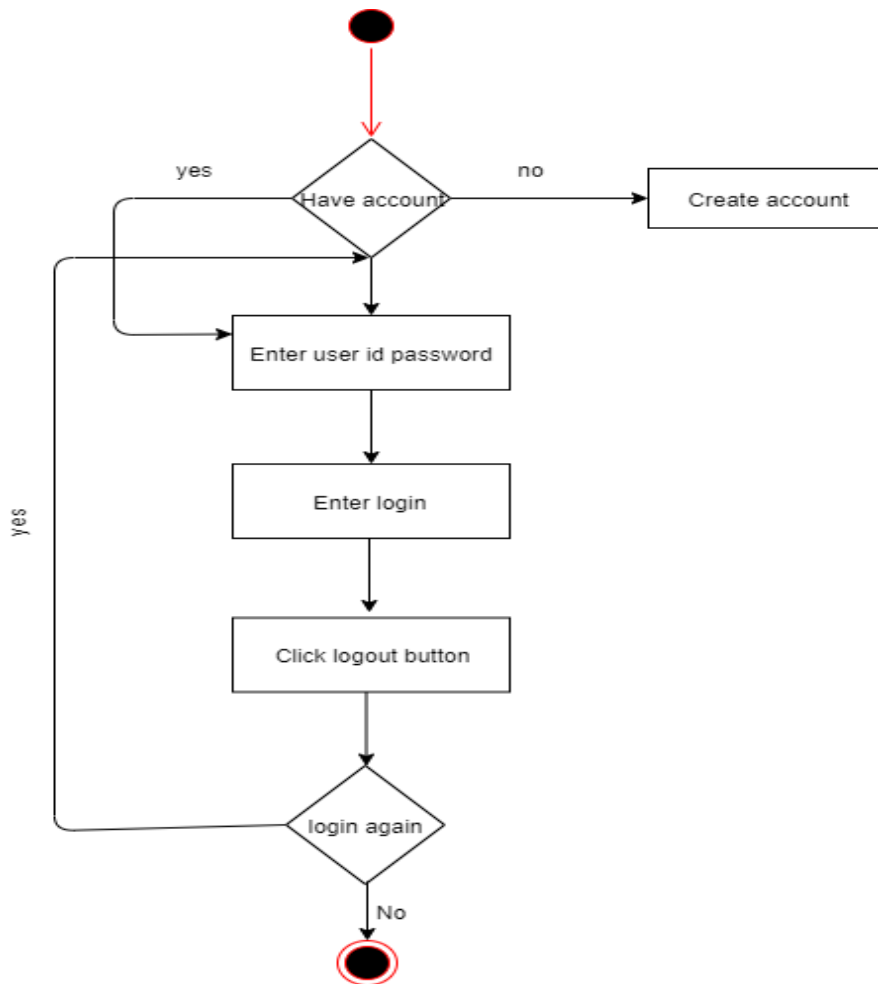


Figure 3.7: Activity diagram of manage profile

### 3.2.5 View seat Arrangement.

Again Student and admin search by course name and section name. For this Student and Admin should click “View seat plan”. Then he should click plan now button. After that, he should click “Plan Now” Admin and student.

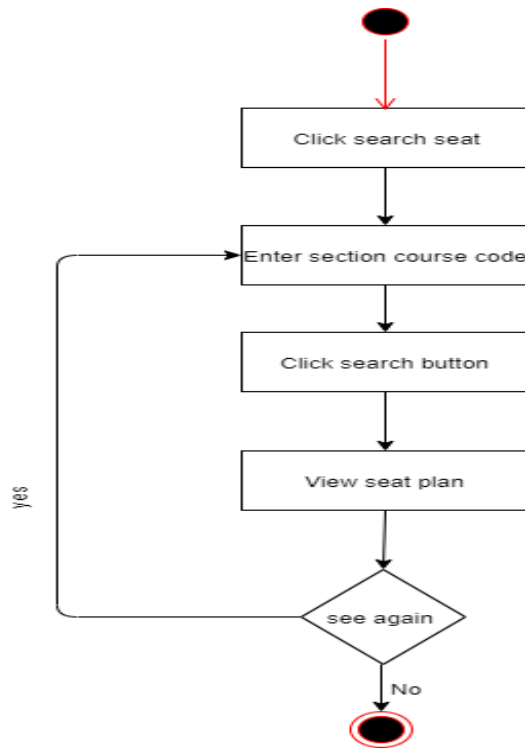


Figure 3.8: Activity diagram of View seat arrangement

### 3.3 Sequence Diagram

Mainly sequence diagrams understand us how the data will be followed in any application. Now we are going to show some sequence diagrams.

#### 3.3.1 Create account.

dit

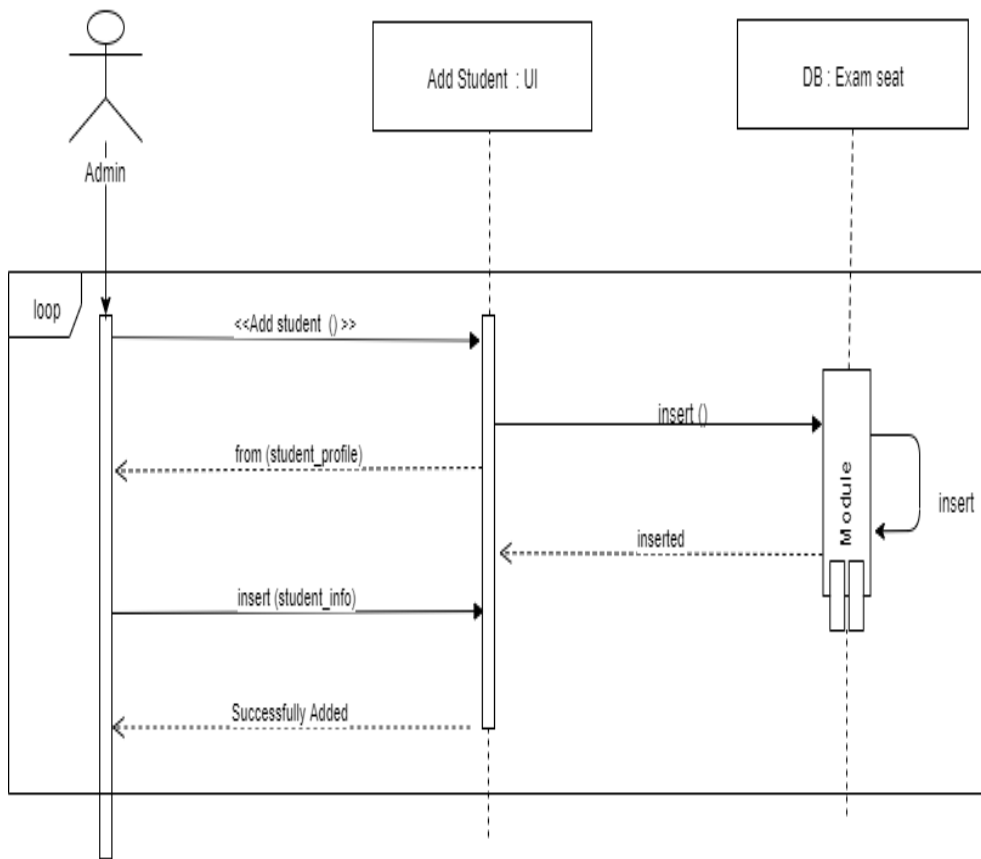


Figure 3.12: Sequence Diagram for create account

### 3.3.2 Manager account

dispatch

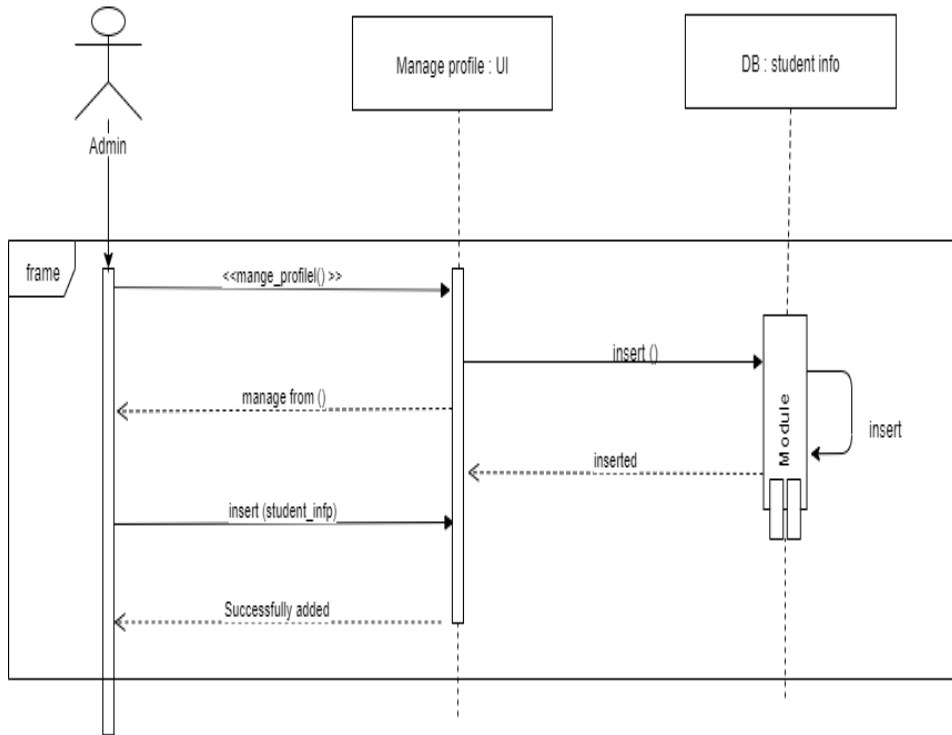


Figure 3.13: Sequence Diagram of Manager account

### 3.3. Manage Students info.

dispatch

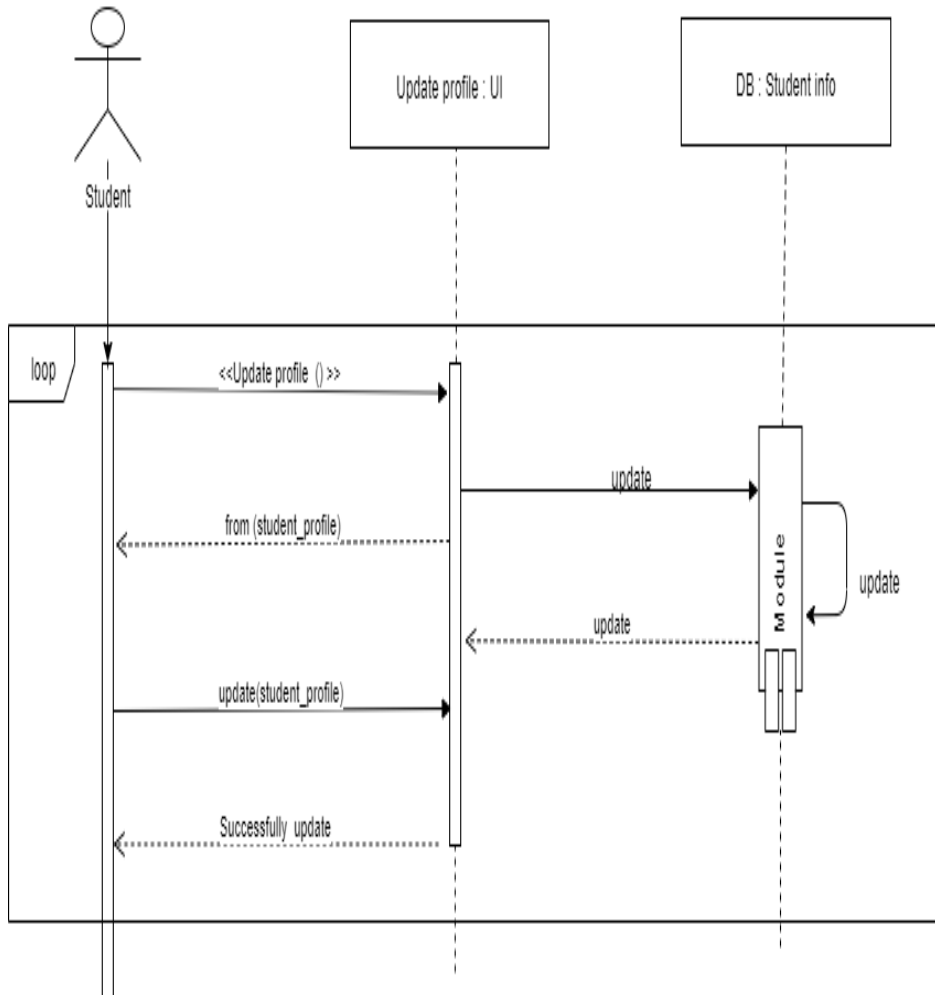


Figure 3.14: Sequence Diagram of manager student info

### 3.3.4 Add Department.

dispatch

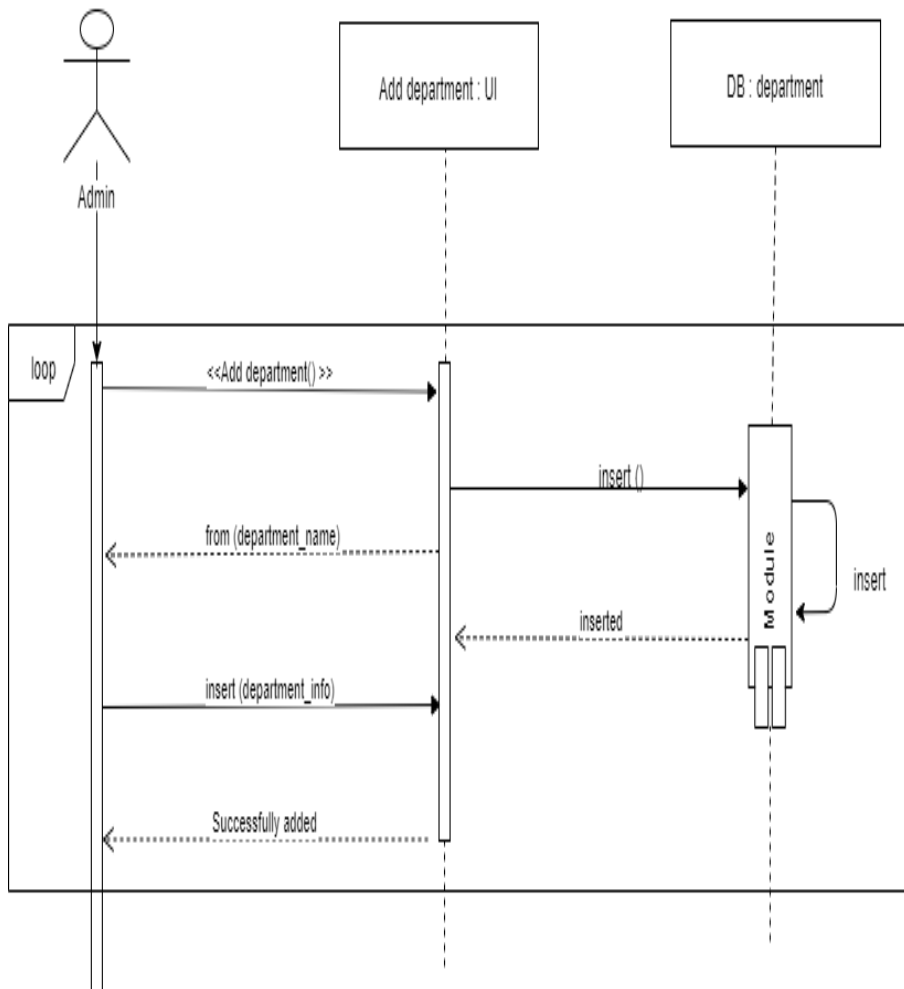


Figure 3.15: Sequence Diagram of add department

### 3.3.5 Add exam hall.

dispatch

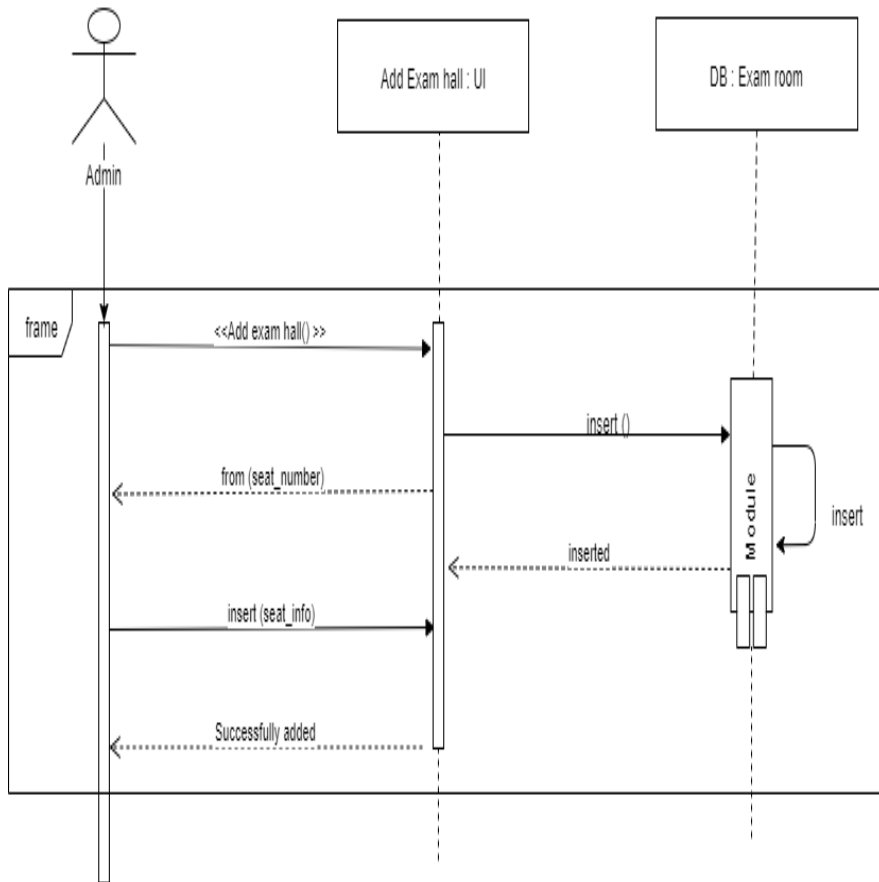


Figure 3.16: Sequence Diagram of Add exam hall

### 3.3.6 Arrange seat info.

dispatch

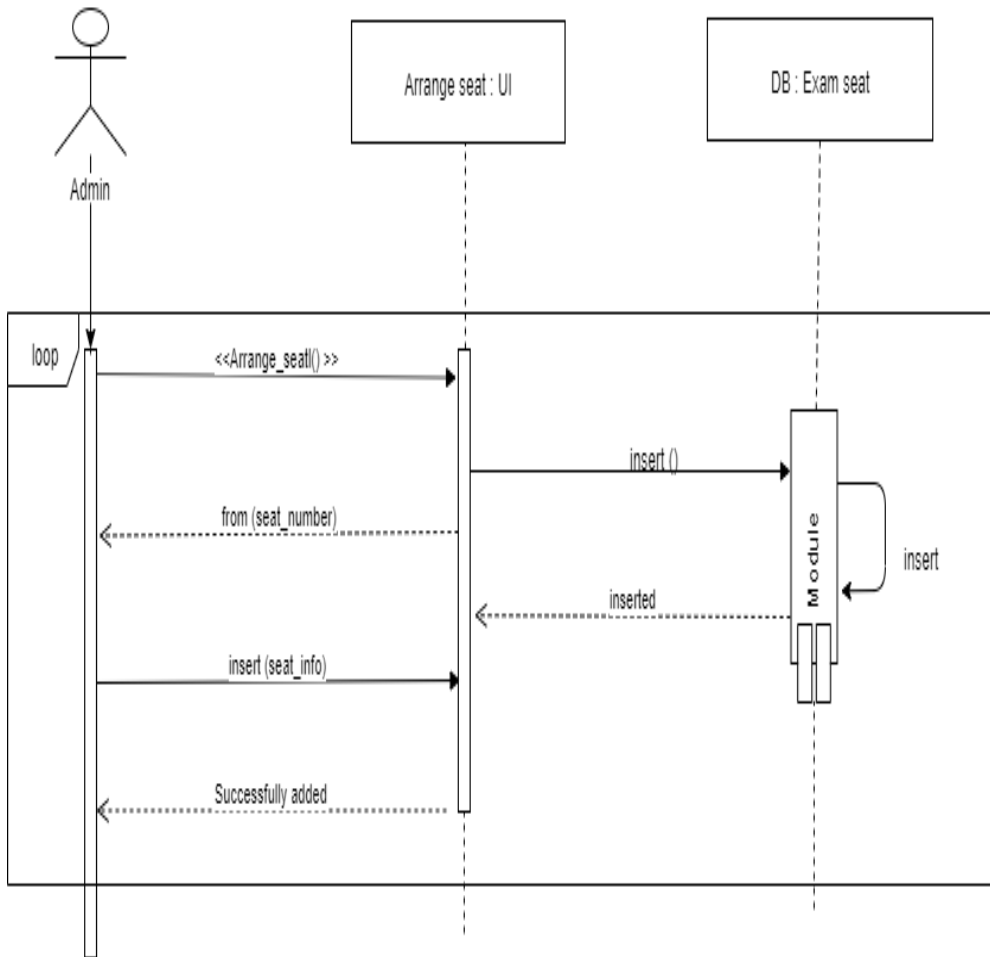


Figure 3.17: Sequence Diagram of arrange seat plan



### 3.3.7 Manage profile.

dispatch

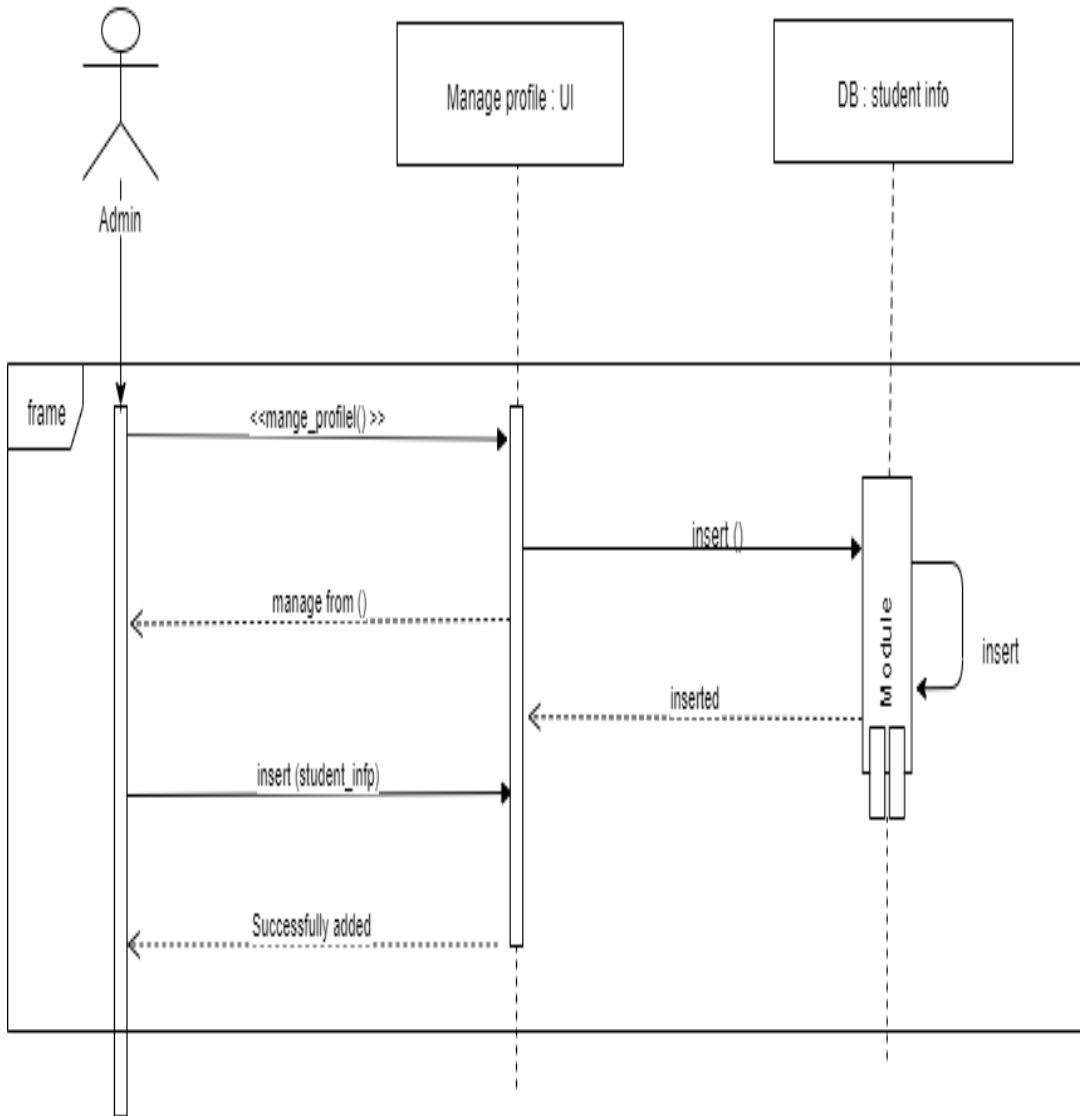


Figure 3.18: Sequence Diagram of manage profile

### 3.3.8 View seat arrangement.

dispatc

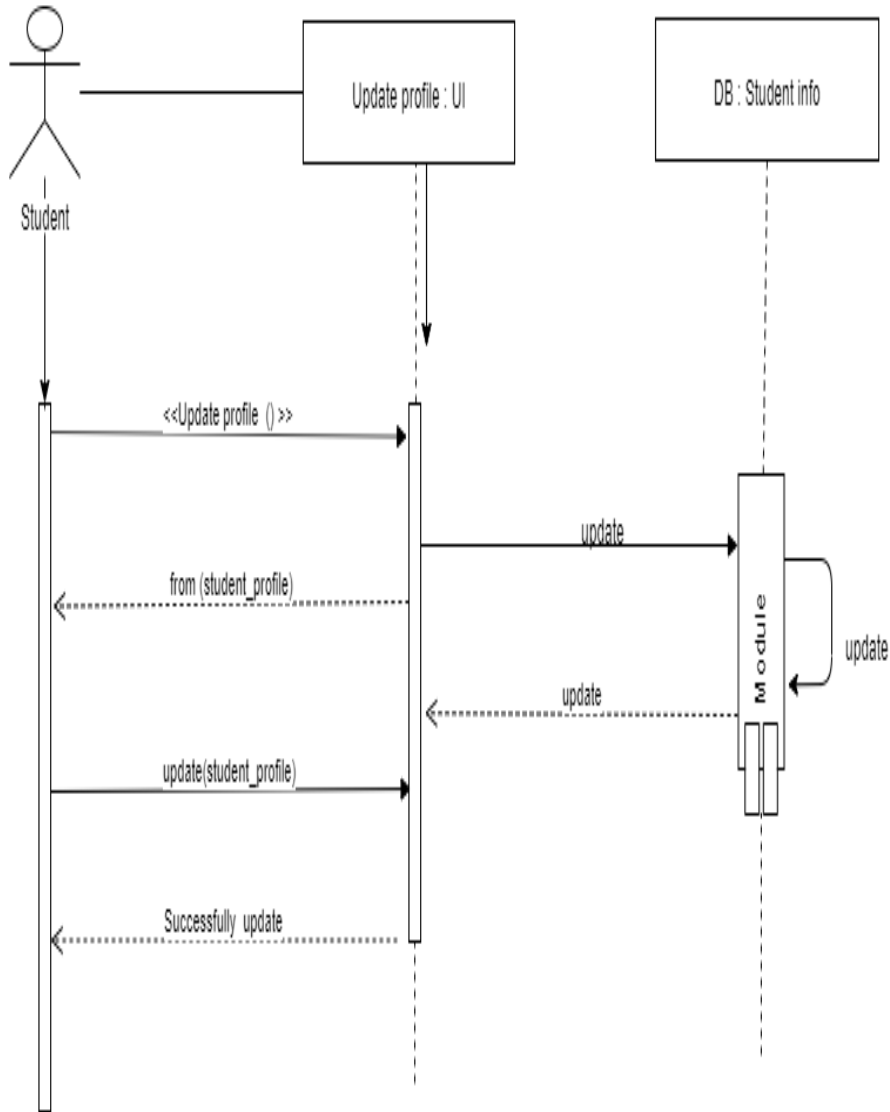
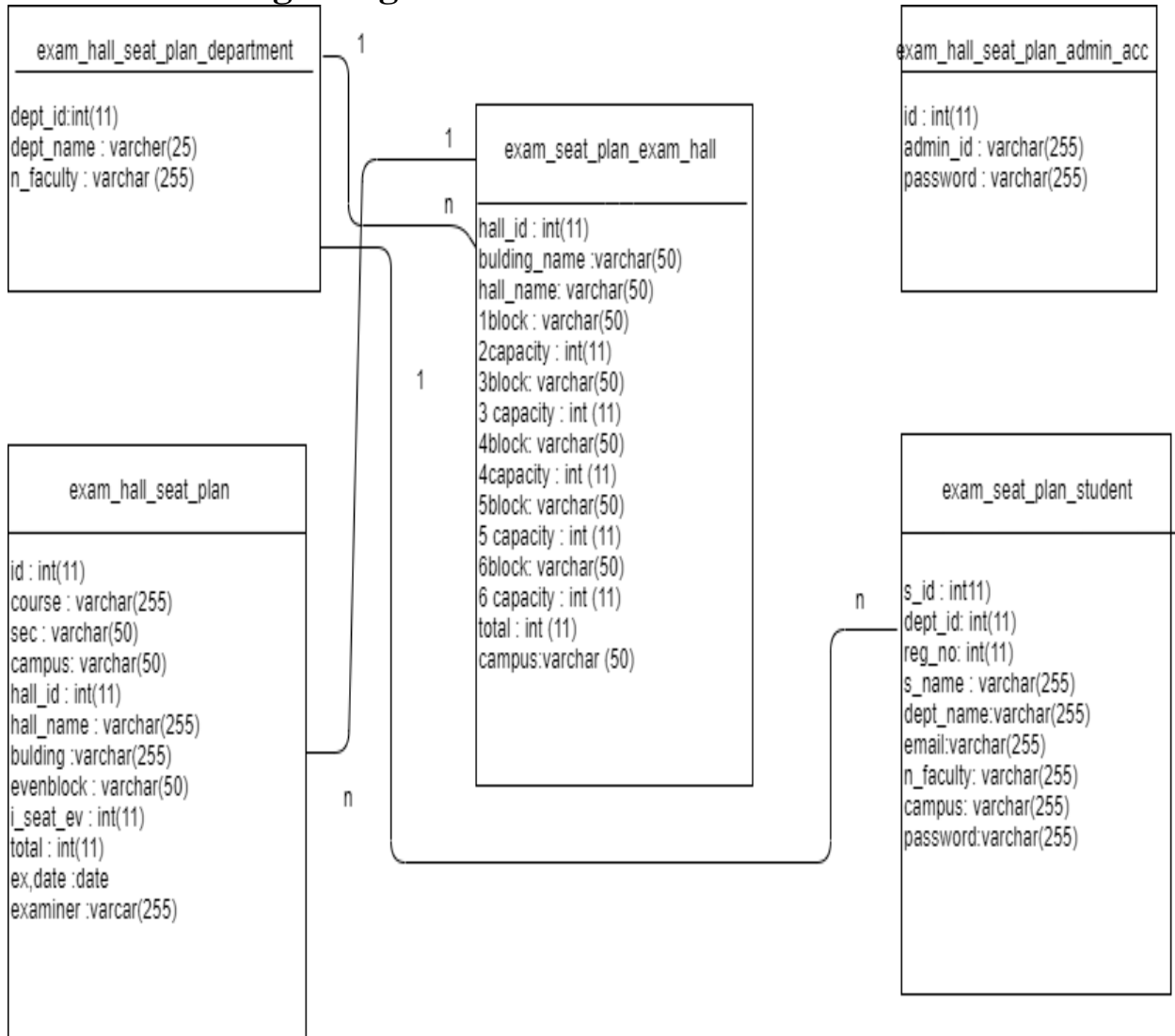


Figure 3.19: Sequence Diagram of view seat arrangement

# **Chapter 4**

## **System Design Specification**

## 4.Database design diagram



Text

**Fig: Database for exam hall seating allotment system**

### 4.1 Development tools and technology

Software development tools are used to develop software. There are different kinds of tools for software developers to make the process of software development very smoothly. But now I will mention some of tools that I am using to develop my project.

## **4.2 User Interface Technology**

First of all, after landing any application, interface of that software is appeared to the user. So, the importance of user interface is very high. For success of any software application, a good looking user interface plays a vital role. User interface includes using good image, graphics, typography, stylesheets, scripting etc.

## **4.3 CSS framework or Bootstrap**

After completing markup, cascading style sheets are designed. It generally explains how HTML elements will display. There are three ways to write stylesheet. One internal CSS, another is external CSS. And last one is inline CSS. But most of time external CSS is used. Because, by using external CSS, all CSS data can be kept in different files. But nowadays CSS framework seems to be very popular. In our application, I have used Bootstrap4. This framework is developed mainly for CSS and JavaScript. This framework is highly used for designing website and web application also. Like Google Chrome, Mozilla Firefox, Opera, Safari, Internet Explorer etc. Bootstrap also provides media query which features responsive layout for different devices with different screen size. Bootstrap provides a set of some files which contains stylesheets which gives basic definition. Bootstrap also provides some JavaScript components also. There are some built in components like JQuery UI. By using Bootstrap, we get both CSS and JavaScript facilities with a single platform. But before start designing an application interface with Bootstrap, one may have some basic knowledge about this framework

## **4.4 JQuery UI**

JQuery UI is a name of library which uses JavaScript as core programming language. It simplifies codes of a programming language named JavaScript. Generally, it interacts with Graphical User Interface (GUI). It also provides visual effects with animation. It can also override cascading style sheets. It is also a subset of JavaScript. By using AJAX no page reloading is required. It is also compatible with any browser like Google Chrome, Mozilla Firefox, Opera, Safari, Internet Explorer etc. It will increase the efficiency

## **4.5 Programming Language**

For developing any application system minimum one programming language is essential. In my application, there are two different programming language is used. One is for front-end side. And another is for server side. The front-end language that I have used to my application is HTML. And the server side programming language name is PHP which stands for Hypertext Preprocessor. Both of them are open source general purpose scripting language.

### **4.1.1 Implemented tools and platform**

As I have said before, there are some tools and technologies that need to be used for developing software. It is very important to determine which tools and platforms are the best match of my requirements. After making a proper decision, one need to start using them.

### **4.1.2 Integrated Development Environment**

IDE stands for Integrated Development Environment. Programmers write code on IDE. After that IDE provide the feature to execute the source code. For developing my web application, I have used an IDE.. To develop my web version, I have used SublimeText which is powered by Sublime HQ Company. It is a commercial IDE for cross platform environment. It is able to suggest code to the programmers also.

### **4.1.3 Web Server**

We have used xampp apache server. It is a free and open source software to use. It can be used on cross platform. It supports a wide range of features and most of them are already implemented as compiled modules. This module can extend the main features or core functionality.

### **4.1.4 Database Server**

For developing our whole project, we have followed Relational Database Management System or RDBMS. And we find that MySql provides the feature of RDBMS. So we should not have any issue to use MySql database. It is also very easy to use. It can also ensure the security, scalability, high performance and many things.

# **Chapter 5**

## **System Test**

## 5.1 Testing Features

Feature testing can be considered as making change to add or modify the new functionality to the existing project. To test the features and functionality, a new test set is to be written for testing purpose. Almost every feature and functionality have different characteristics. Those are designed to make the application more useful, intuitive, reliable, secured, scalable, effective and efficient.

## 5.2 Features to be tested

Features	Priority	Description
Login	1	User must be authenticated by login
Logout	1	Session must be destroyed after logout
Create account	2	Student must be create account
Manager account	2	Admin must be manager account
Add Department	2	Admin data must be add department
Seat arrangements	3	Search by course name and section name
Manage profile	2	Admin and Students update profile.
<b>Technological Features</b>		
Database	1	Database will be used at almost every operation. So this is why, this part must be controlled tightly.

Here, 1 = Low Priority, 2 = Medium Priority, 3 = High Priority

## 5.3 Testing Strategy

Testing strategy is to be considered as a general purpose of testing process. Testing object, testing function methods, total available resources are included to the testing strategy. It is also the indicator of test levels that are to be performed on the whole software development life cycle. Those strategies that are prepared by the quality assurance team should be reviewed by the developers of the application. After that it should be reviewed by the test team leads. Different kinds of testing strategies can be performed according to the type of application system that need to be tested.

## 5.4 Test approach

To complete the test process, testers must take some approach. There are mainly two test approach.



• **Automation testing:** Automation testing is a name of testing technique by which test engineers prepare some scripts according to test plan and after that they use suitable tools to perform testing of the software. Nowadays, almost every software company follow the approach of automation testing.

• **Manual testing:** Manual testing is also a name of technique of testing by searching out the bugs or vulnerability in an application. In this process, test engineers manually test and execute the test cases without having any automation tools.

## **5.5 Black Box Testing**

Black box testing is a software testing method by which test engineers test the software without having knowledge of the internal architecture of the application that need to be tested. It is also known as behavioral testing. Black box testing can be both functional or non-functional. It ignores the internal mechanism of a system. We have decided to perform the equivalent class partitioning and Boundary value analysis techniques to implement.

### **5.1.1 Equivalent Class Partitioning**

Equivalent class partitioning is also known as equivalent partitioning. In this technique input data is to be divided into groups. Those group of data are expected to perform as similar behavior. Each groups works as same as other groups also.

The main advantage of following the approach of equivalent class partitioning is to reduce the total number of test cases from infinite to finite. Another advantage is that it can be applied to all testing levels also.

For legal input value output will produce a meaning data. But for illegal input value, output will not produce meaningful data.

### **5.1.2 Boundary Value Analysis**

Boundary value analysis is a name of software testing by which input data can be included within a boundary range. Following this strategy, test engineers firstly define the boundary value. After that they take a specific boundary and then write the test cases that will be go through the application for testing purpose. After completing those tasks, testers use test cases to test applications software.

### 5.1.3 White Box Testing

White box testing is also a name of testing approach which is also known as clear box testing, glass box testing, open box testing, transparent box testing, code based testing or structural testing. It is opposite to the black box testing. In black box testing, the internal architecture or algorithms is not known to the testers whether the whole software architecture is known to the testers while white box testing.

Testers can also predict the output of every test cases for white box testing. White box testing can be classified into some levels. Such as:

- Unit Testing
- Integration Testing
- System Testing

The main advantage of white box testing is that testing is more throughout and the testing can be started from the very beginning stage.

### 5.1.4 Pass / Fail Criteria

Pass or fail criteria will be set by the test engineers. They will prepare the pass / fail criteria on the basis of which input data are worked and which are not works well. Those data that are worked well will considered as pass criteria. And rest of input data will be considered as fail criteria.

Now I will give the pass / fail criteria below.

- System crash will not be considered as pass case.
- If any criteria pass 100% times, then it will be considered as pass criteria only.
- If data can't be displayed to the application properly, then it is also to be considered as fail criteria.

### 5.2.1 Testing Schedule

Test Phase	Time
Testing plan create	1 week
Test specification	2 week
Unit testing	During development time
Component test	1 week
Test Phase	Time
Integration testing	1 week
Validating use cases	1 week
Testing user interfaces	1 week
Load testing	1 week
Performance testing	1 week
Release to production	1 week

## 5.2.2 Trace Ability Matrix

Project Manager			Business Analyst Lead		
QA Lead			Target Implementation Date		
BR#	Category Functionality / Activity	Requirement Description	Use Case Reference	Test Case Reference	Comments
Br-1	Functional	Create account	Use case 3.1.1	Test case 5.6.1	
BR-2	Functional	Manage account	Use case 3.1.2	Test case 5.6.3	
BR-3	Functional	Manager student Info	Use case 3.1.3	Test case 5.6.4	
BR-4	Functional	Add department	Use case 3.1.4	Test case 5.6.5	
BR-5	Functional	Add exam hall	Use case 3.1.5	Test case 5.6.6	
BR-6	Functional	Arrange seat Info	Use case 3.1.6	Test case 5.6.7	
BR-7	functional	View seat arrangement	Use case 3.1.7	Test case 5.6.7	

### **5.2.3 Testing Environment**

Testing environment means to prepare the environment with hardware and software so that test engineers can be able to execute test cases as required. Besides hardware and software usage, network configuration might be needed to execute test plans.

For making the environment for testing, some key area need to setup. Those are:

- Test data
- Database server
- Client's operating system
- Front end running environment
- Browser
- System and application
- Network
- Hardware with server operating system

### **5.2.4 Test Cases**

A test case refers to some rules and regulations or conditions by which it can be determined whether a system can be able to meet the works or requirements under test cases properly. We know, there is a chance to have some fault or break holes in any application. This is very common scenario. And those issues are solved usually by software testing approaches. But if we don't care of those issues, then the full application development might be ruined. So proper testing must be needed. For testing our application, I have prepared some test cases. Now, I am going to provide them.

### 5.3.1 Manage account

<b>Test case #1</b>		<b>Test case name:</b> Manage account			
<b>System:</b> Exam hall seating allotment system		<b>Subsystem:</b> N/A			
<b>Designed by:</b> Khalequzzaman miraj		<b>Designed date:</b> 2/12/19			
<b>Executed by:</b> Khalequzzamaqn miraj		<b>Executed date:</b> 2/12/19			
<b>Short description:</b> The Admin manage account application.					
<b>Pre-conditions:</b> Admin manage all account system Admin department name and room number allotment system					
<b>Step</b>	<b>Username</b>	<b>Password</b>	<b>Expected result</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Admin	1234	Invalid	Pass	Good
2	Miraj		Password can't be blank	Pass	Required field
3	Rakib		Username can't be blank	Pass	Required field is correct
4	Miraj	1234	Successfully login to the application and redirected to the dashboard	Pass	Good
<b>Post-conditions:</b> Admin including hall seating will successfully to the application.					

### 5.3.2 Manage student info

<b>Test case #2</b>		<b>Test case name:</b> Manage Student info		
<b>System:</b> Exam hall seating allotment system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Khalequzzaman miraj		<b>Designed date:</b> 2/12/19		
<b>Executed by:</b> Khalequzzaman miraj		<b>Executed date:</b> 2/12/19		
<b>Short description:</b> Admin including Students info need to fill up all that fields which as marked as required.				
<b>Pre-conditions:</b> Manage student all details				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Admin collect information	Collection info	Pass	correct
2	Admin store the information	Application will save those information.	Pass	Correct
<b>Post-conditions:</b> Every information is saved to the database of the application program properly.				

### 5.3.3 Add department

<b>Test case #3</b>		<b>Test case name:</b> Add department		
<b>System:</b> Exam hall seating allotment system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Khalequzzaman miraj		<b>Designed date:</b> 2/12/19		
<b>Executed by:</b> khalequzzaman miraj		<b>Executed date:</b> 2/12/19		
<b>Short description:</b> Add Department.				
<b>Pre-conditions:</b> Admin must have an account				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>

1	Admin login with user id and password	Login successfully	Pass	Correct
2	Data provided properly	Data saved successfully	Pass	Correct
<b>Post-conditions:</b> After providing legal user id password admin successfully login to the system				

### 5.3.4 Admin add exam hall

<b>Test case #4</b>		<b>Test case name:</b> Add exam hall		
<b>System:</b> Exam hall seating allotment system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Khalequzzaman miraj		<b>Designed date:</b> 2/12/19		
<b>Executed by:</b> Khalequzzaman miraj		<b>Executed date:</b> 2/12/19		
<b>Short description:</b> Admin Add exam hall.				
<b>Pre-conditions:</b> Admin must gave Exam hall information				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Admin provide data of exam hall	provide all exam hall required data	Pass	Correct
2	Admin successfully to the system	Application show this exam hall info	Pass	Correct
<b>Post-conditions:</b> After providing add exam hall ,it will be add in database.				

### 5.3.5 Arrange seat info

<b>Test case #5</b>		<b>Test case name:</b> Arrange seat info		
<b>System:</b> Exam hall seating allotment system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Khalequzzaman miraj		<b>Designed date:</b> 2/12/19		
<b>Executed by:</b> Khalequzzaman miraj		<b>Executed date:</b> 2/12/19		
<b>Short description:</b> Admin Arrange seat info.				
<b>Pre-conditions:</b>				
Admin must be know about this Arrange seat properly.				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Admin provide data seat info	Arrange seat properly	pass	Correct
2	Submit an Arrange seat info.	Application show this seat plan all data.	Pass	Correct
<b>Post-conditions:</b> Admin must to have total seat plan				



### 5.3.6 Manage profile

<b>Test case #6</b>		<b>Test case name:</b> Manage profile		
<b>System:</b> Exam hall seating allotment system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Khalequzzaman miraj		<b>Designed date:</b> 2/12/19		
<b>Executed by:</b> Khalequzzaman miraj		<b>Executed date:</b> 2/12/2019		
<b>Short description:</b> Admin Update profile seat number room number department				
<b>Pre-conditions:</b> Admin manage all account system Admin department name and room number allotment system				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	All data is not provided	Application manage required data	Pass	Correct
12	Submit an update profile student and admin	Update profile successfully	Pass	Correct
<b>Post-conditions:</b> After providing Manage profile.				

### 5.3.7 View seat arrangement

<b>Test case #7</b>		<b>Test case name:</b> Create Post order		
<b>System:</b> Exam hall seating allotment system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Khalequzzaman miraj		<b>Designed date:</b> 2/12/19		
<b>Executed by:</b> Khalequzzaman miraj		<b>Executed date:</b> 2/12/19		
<b>Short description:</b> student view seat arrangement				
<b>Pre-conditions:</b> Student must be authenticated by university.				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Student show the seat	view seat to provide all required data	Pass	Correct
12	Submit to view seat arrangement.	Application provided this search by course name and section all data.	Pass	Correct
<b>Post-conditions:</b> After providing student view the seat plan.				

## **Chapter 6 Requirement Analysis**

## 6. User Interface

First of all, after landing any application, interface of that software is appeared to the user. So, the importance of user interface is very high. For success of any software application, a good looking user interface plays a vital role.

### 6.1 Home page

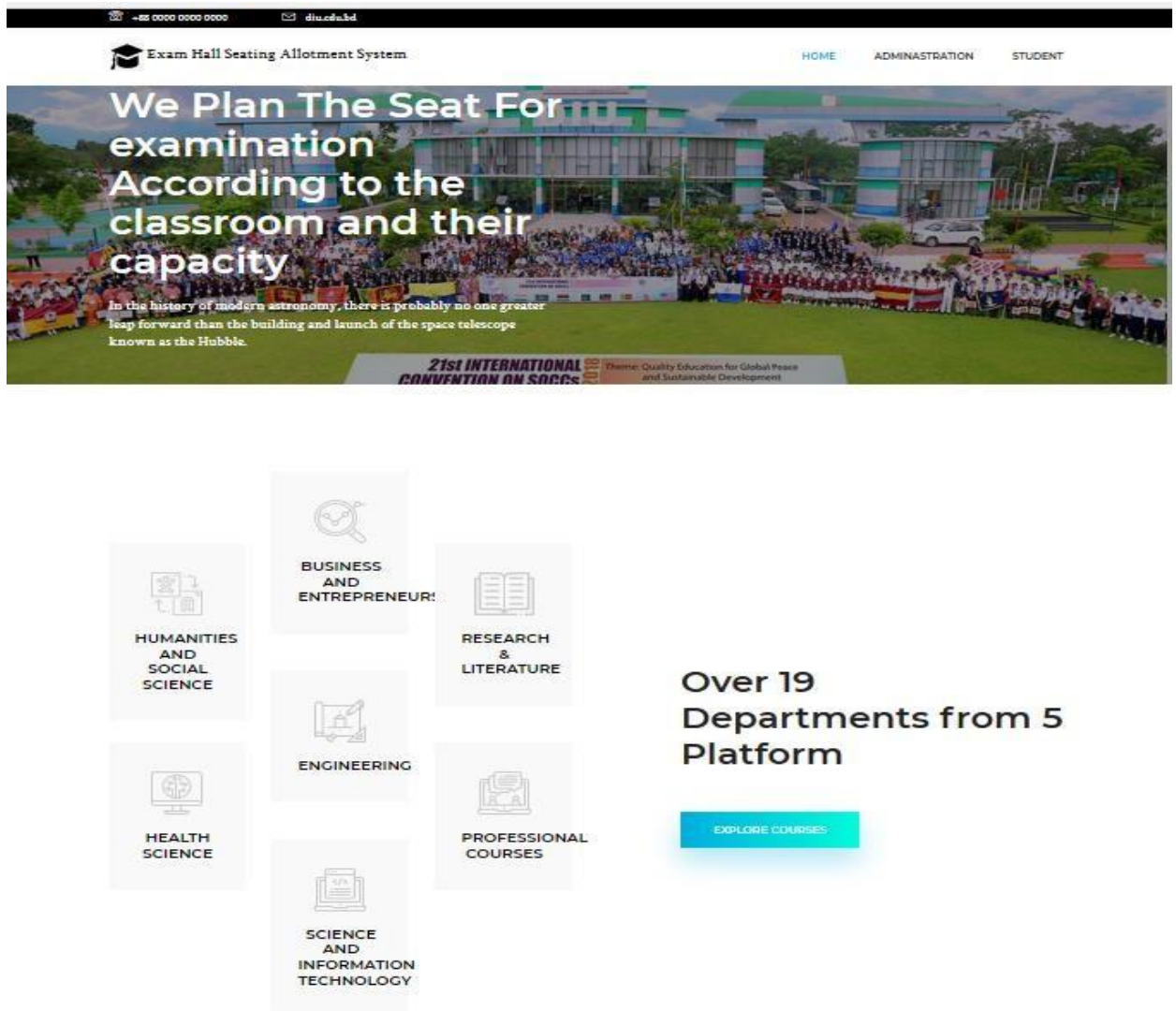


Figure 5.1: Home page

## 6.2 Administrator Log in Page

Exam Hall Seating Allotment System

HOME ADMINISTRATION STUDENT

This is a required field **Administrator Log In**

User ID

Enter User ID

Password

Password

Submit

### Campuses

#### Parmanent Campus

DIU Parmanent Campus is a Green Campus, the first of its kind in Bangladesh, which was launched in 2002 by (DIU)

Khagan Bazar, Ashulia, Savar, Dhaka

#### Main Campus

We have Incrative City Campus as a Main Campus. Its situated at Shebhanbag, Dhaka-1207, which was launched in 2002 by (DIU)

102-Shukrabad, Mirpur Road, Dhaka-1207

Copyright © 2018

Figure 5.2: Administrator Log in Page

## 6.3 Admin main page

Exam Hall Seating Allotment System

HOME STUDENT MANAGEMENT ADD DEPARTMENT SEAT PLANNING LOG OUT

### Department Information

Faculty Of Science & Information Technology(FSIT)	VIEW DETAILS
Faculty Of Business & Enterprneurship(FBE)	VIEW DETAILS
Faculty Of Engineering(FE)	VIEW DETAILS
Faculty Of Allied Health Sciencs(FAHS)	VIEW DETAILS
Faculty Of Humanities & Social Science(FHSS)	VIEW DETAILS

### Department Wise Details

#	Department Name	Campus Name	Total Student
1	FSIT	Main Campus	1200

### Programme wise Details

#	Course Name	Department Name	Campus Name	Total Student
1	SWE	FSIT	Main Campus	500

Fig: 5.3 admin main page

## 6.4 Student login page

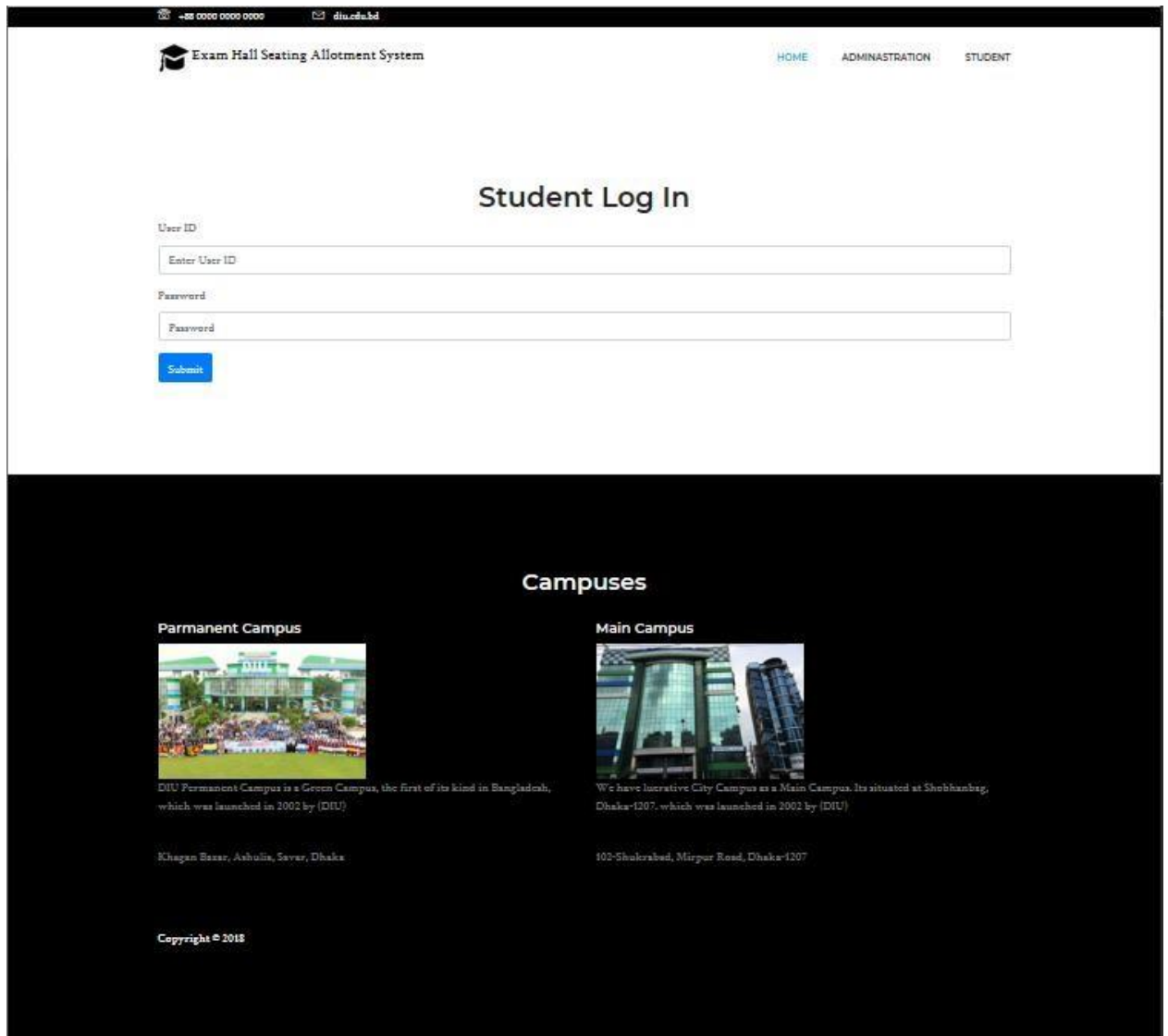


Figure 5.4: Student login page

## 6.5 Update profile

The screenshot shows a web interface for the 'Exam Hall Seating Allotment System'. The header includes a logo, the system name, and navigation links: HOME, UPDATE MY PROFILE, SEE SEAT ARRANGEMENT, and LOG OUT. The main content area is titled 'Add Student Here' and contains a form with the following fields:

- Student ID: 161-35-1548
- Department Name: SWE
- Email Address: khaquzzaman35-1548@diu.edu.bd
- Faculty Name & Initial: Faculty Of Science & Information Technology(FSIT)
- Campus Name(PC/MC): MC
- Password: \*\*\*\*\*

Figure 5.5: update profile



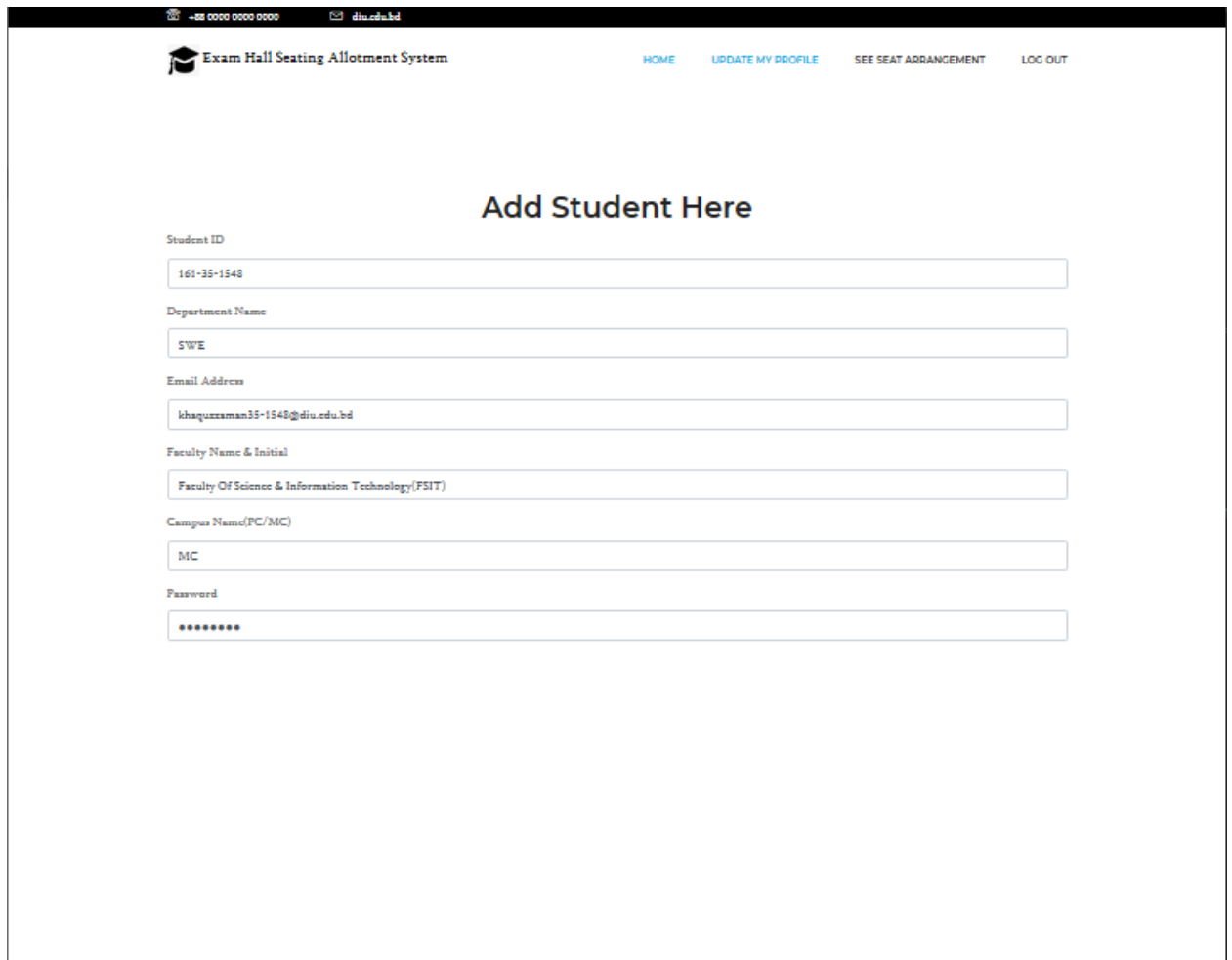
## 6.6 User view seat arrangement

The screenshot displays the 'Exam Hall Seating Allotment System' interface. At the top, there is a navigation menu with links for HOME, STUDENT MANAGEMENT, ADD DEPARTMENT, SEAT PLANNING, and LOG OUT. Below the navigation, there are two search input fields: 'Course Name' and 'Section Name'. A blue 'Search Now' button is positioned below these fields. The main content area is titled 'Seat Arrangement Information' and contains a table with the following data:

Course Name	Section Name	Campus Name	Building Name	Semester	Hall Name	Blocks/Column Name	Total Student	Examiners Name
(SWE-423)Advanced Database With Lab	B	MC	AB	FALL 2019	902	1,3,5	18	HR,HS

Fig: User view seat arrangement

## 6.7 Update Profile

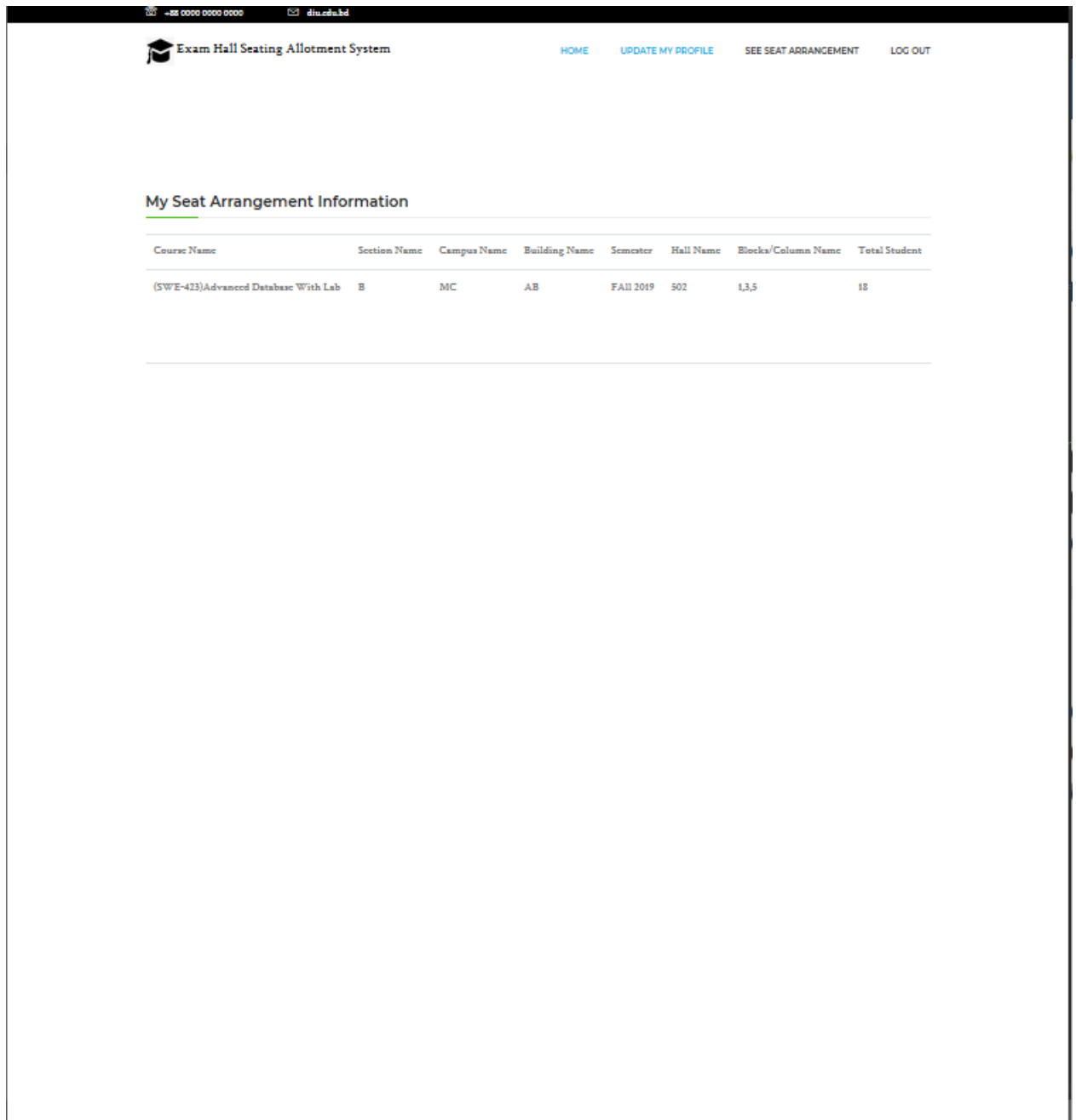


The screenshot shows a web interface for the 'Exam Hall Seating Allotment System'. The header includes a logo, the system name, and navigation links: HOME, UPDATE MY PROFILE, SEE SEAT ARRANGEMENT, and LOG OUT. The main content area is titled 'Add Student Here' and contains a form with the following fields:

- Student ID: 161-35-1548
- Department Name: SWE
- Email Address: khaquzzaman35-1548@diu.edu.bd
- Faculty Name & Initial: Faculty Of Science & Information Technology(FSIT)
- Campus Name(PC/MC): MC
- Password: \*\*\*\*\*

Fig 5.7 Update Profile

## 6.8View Seat Arrangement

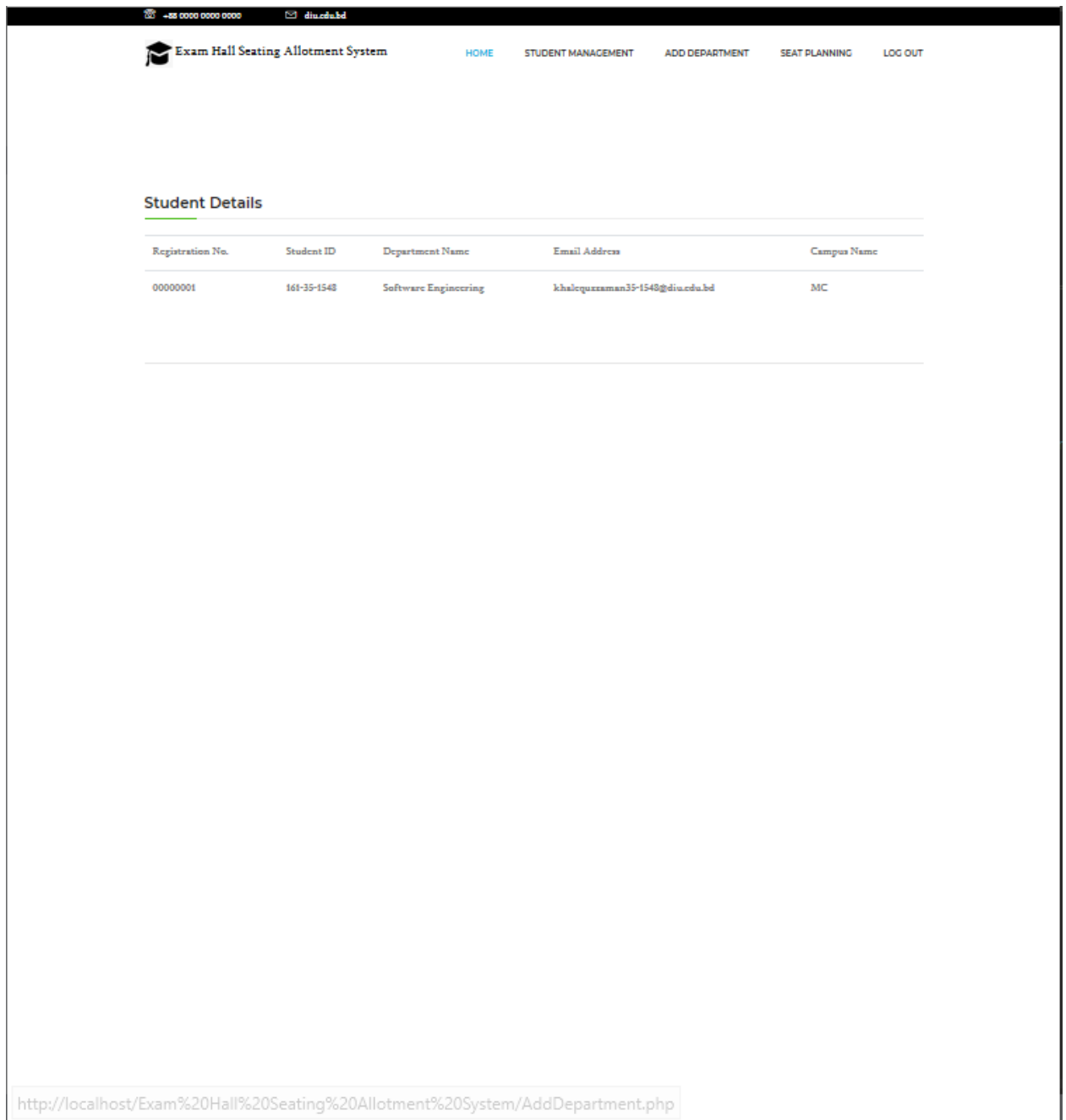


The screenshot displays the 'Exam Hall Seating Allotment System' interface. At the top, there is a navigation bar with the system name and a graduation cap icon on the left, and links for 'HOME', 'UPDATE MY PROFILE', 'SEE SEAT ARRANGEMENT', and 'LOG OUT' on the right. Below the navigation bar, the section 'My Seat Arrangement Information' is highlighted. A table below this section lists the seat arrangement details for a specific course.

Course Name	Section Name	Campus Name	Building Name	Semester	Hall Name	Blocks/Column Name	Total Student
(SWE-423)Advanced Database With Lab	B	MC	AB	FALL 2019	502	1,3,5	18

Fig 5.8View Seat Arrangement

## 6.9 View Student info



The screenshot displays the 'Exam Hall Seating Allotment System' web application. The top navigation bar includes a logo, the system name, and menu items: HOME, STUDENT MANAGEMENT, ADD DEPARTMENT, SEAT PLANNING, and LOG OUT. The main content area features a 'Student Details' section with a table containing one row of student information.

Registration No.	Student ID	Department Name	Email Address	Campus Name
00000001	161-35-1548	Software Engineering	khalequzaman35-1548@diu.edu.bd	MC

The browser's address bar at the bottom shows the URL: <http://localhost/Exam%20Hall%20Seating%20Allotment%20System/AddDepartment.php>

Fig 5.9 View Student info

## 6.10 Search Seat Arrangement

The screenshot shows a web application interface for the Exam Hall Seating Allotment System. At the top, there is a navigation bar with the system name and four menu items: HOME, UPDATE MY PROFILE, SEE SEAT ARRANGEMENT, and LOG OUT. Below the navigation bar, there are two input fields for searching: 'Course Name' and 'Section Name'. A blue 'Search Now' button is positioned below the input fields.

Exam Hall Seating Allotment System

HOME UPDATE MY PROFILE SEE SEAT ARRANGEMENT LOG OUT

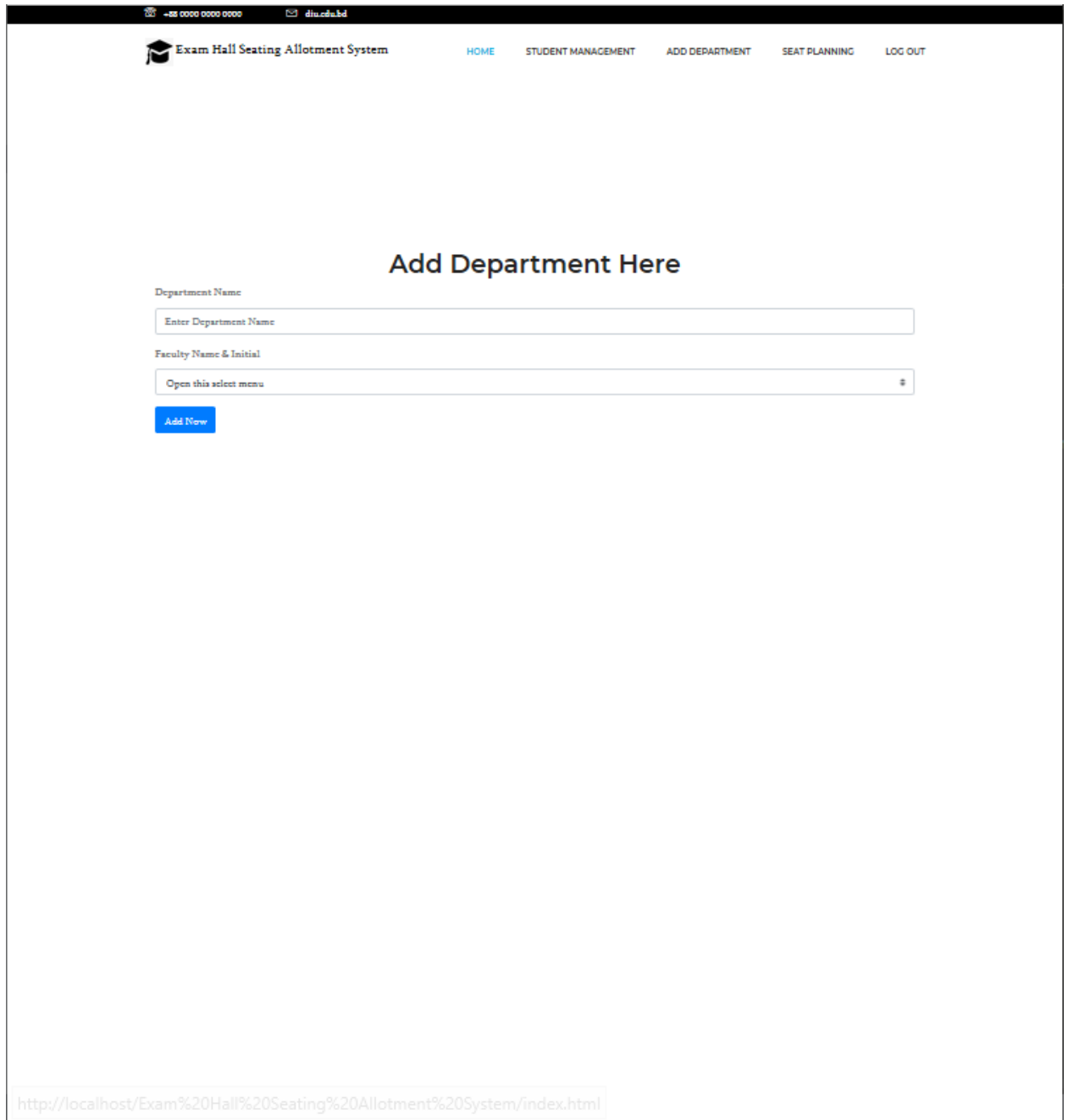
Course Name

Section Name

Search Now

Fig 5.10 Search Seat Arrangement

## 6.11 Add department



The screenshot shows a web application interface for adding a department. At the top, there is a navigation bar with the following items: a logo, the text "Exam Hall Seating Allotment System", and links for "HOME", "STUDENT MANAGEMENT", "ADD DEPARTMENT", "SEAT PLANNING", and "LOG OUT". The main content area is titled "Add Department Here" and contains two input fields: "Department Name" with a placeholder "Enter Department Name" and "Faculty Name & Initial" with a placeholder "Open this select menu". Below these fields is a blue "Add New" button. At the bottom of the page, a browser address bar shows the URL "http://localhost/Exam%20Hall%20Seating%20Allotment%20System/index.html".

Fig 5.11 Add department

## 6.12 Add exam hall

The screenshot shows a web application interface for adding an exam hall. At the top, there is a navigation bar with the following items: a logo, the text 'Exam Hall Seating Allotment System', and links for 'HOME', 'STUDENT MANAGEMENT', 'ADD DEPARTMENT', 'SEAT PLANNING', and 'LOG OUT'. The main heading is 'Add Exam Hall Here'. Below this, there are several input fields: 'Campus Name(PC/MC)' with a dropdown menu showing 'Open this select menu'; 'Building Name' with a text input field containing 'Enter Building Name'; 'Hall Name' with a text input field containing 'Enter Hall Name'; '1st Block Name' with a dropdown menu showing 'Open this select menu'; '2nd Block Name' with a dropdown menu showing 'Open this select menu'; '3rd Block Name' with a dropdown menu showing 'Open this select menu'; and 'Seating Capacity' with a text input field containing 'Enter Seating Capacity'. A blue 'Submit' button is located at the bottom left of the form area.

Fig 5.11 Add exam hall

## 6.13 Add seat arrangement

The screenshot shows a web application interface for 'Exam Hall Seating Allotment System'. The page title is 'Add Seat Allocation Here'. The form contains the following fields:

- Course Name:
- Section Name:
- Campus Name(PC/MC):
- Building Name:
- Hall Name:
- Block Name:
- Total Seating Capacity:
- Examiners name:

At the bottom of the form is a blue button labeled 'Save'.

Fig 5.12 Add seat arrangemen



## 6.14 Add student

The screenshot shows a web application interface for adding a student. At the top, there is a navigation bar with the logo and name 'Exam Hall Seating Allotment System' and several menu items: HOME, STUDENT MANAGEMENT, ADD DEPARTMENT, SEAT PLANNING, and LOG OUT. The main content area is titled 'Add Student Here' and contains a form with the following fields:

- Registration NO. (Text input field with placeholder 'Enter Registration NO.')
- Student ID (Text input field with placeholder 'Enter Student ID')
- Department Name (Text input field with placeholder 'Enter Department Name')
- Email Address (Text input field with placeholder 'Enter Email Address')
- Faculty Name & Initial (Dropdown menu with placeholder 'Open this select menu')
- Campus Name(PC/MC) (Dropdown menu with placeholder 'Open this select menu')
- Password (Text input field with placeholder 'Password')

A blue 'Submit' button is located at the bottom left of the form.

Fig 5.13 add student

# **Chapter 7**

## **Conclusion**

## GitHub Link

### 7.1 Project Summary

I have started to develop this project from April. From the starting of development of this project, it demands hard working, patients, persistency to meet the requirements of stakeholders. After that I have proposed the design. And then started to work. Database plays a vital role for any application software. And so this is why, I have designed the database diagram having tables with proper relationship. After that, I have prepared the user interface and take their approval to continue to the next part. It is to be said that, the interface of my application is very simple and easy to understand. After completing that, I have started to write the core functionality of the project.

Developing project is not end of all tasks actually. There are some other important tasks to perform. And that is testing. It is also known as quality assurance also. Almost at every software company there are a quality assurance team. Their main responsibility is to find the loop holes or vulnerability of software. If there any bug remains before handover to the stakeholders, there is a change to ruin the whole project. So testing plan is very important. And after developing the project, I have tried to assure the quality of this project.

### 7.2 Limitations

For developing this project, I have faced some limitations. Now I will describe those in brief.

- **Sending reports by email:** In our application, there is not having any mailing method directly by which one can contract with others for business purpose. End user need to email or contract with others now by manual way.

### 7.3 Obstacles and Achievements

I believe that if there are not any obstacles to develop a project, then there doesn't have any challenges. Because we know, challenge give us the opportunity to prove ourselves. Obstacles, challenges and achievements are like a path to the success. Before starting this project, I didn't know the actual flow of software development life cycle. By developing this project, I have learnt to know how to have a row requirement from clients. After that I have learnt system analysis, database design and many things. My supervisor helps me a lot from the very beginning of the development of this project.

There are some other obstacles and achievements also that I will describe below.

- **Lack of Actor’s Engagement:** There are different types of stakeholders in our full web application. And each actors uses different functionality. And almost they are busy with their day to day activities. So this is why, T&A and Planning manager can’t plan or update plan in time.

- **Scope Change:** Sometimes, some features need to be changed or modified. Then I need to follow reverse engineering process. And again designed to meet the new requirements. It also made me frustrated sometimes.

### 7.4Future Scope

I have learnt a lot throughout the whole development stage of this project. For making this project developed, I have also meet some young entrepreneurs and enthusiasts also. I am very much thankful to all of them as their idea and discussion gave me some opportunities to make my product complete. It will help me to work with similar type project in future also.

### 7.5References

I have gained some knowledge from some platforms. Obviously I will mention those references. For making my project successful those resources help me a lot. Not I will mention the names below.

<a href="https://blog.stoneriverelearning.com/16-best-websites-for-learning-php-programming/">https://blog.stoneriverelearning.com/16-best-websites-for-learning-php-programming/</a>	PHP programming
<a href="http://www.stakeoverflow.com">www.stakeoverflow.com</a>	Mysql_fetch_Assoc() function error
<a href="http://www.php.net">www.php.net</a>	Joining value
<a href="http://www.mysql.com/products/workbench">www.mysql.com/products/workbench</a>	Joining query
<a href="http://www.w3schools.com">www.w3schools.com</a>	HTML & CSS
<a href="http://www.jquery.com">www.jquery.com</a>	Print function
<a href="https://www.youtube.com/watch?v=2yaUC8PiuI0">https://www.youtube.com/watch?v=2yaUC8PiuI0</a>	Programming
<a href="http://www.wikipedia.com">www.wikipedia.com</a>	Documentation
<a href="http://www.mysql.com">www.mysql.com</a>	Foreign key Creation
<a href="http://www.getbootstrap.com">www.getbootstrap.com</a>	Joining query & CSS
<a href="http://www.github.com">www.github.com</a>	Authentication
<a href="http://www.draw.io/">www.draw.io/</a>	Drawing all diagram