

UNIVERSITY HALL MANAGEMENT SYSTEM

JANNATUL FERDOUS

MASTER OF ENGINEERING IN INFORMATION
AND COMMUNICATION ENGINEERING
NOAKHALI SCIENCE AND TECHNOLOGY
UNIVERSITY

UNIVERSITY HALL MANAGEMENT SYSTEM

JANNATUL FERDOUS

The project report submitted in fulfillment of the requirements
for the award of the degree of
Master of Engineering in Information and Communication Engineering

Department of Information and Communication Engineering

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

January 2021

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project and in our opinion, this project is adequate in terms of scope and quality for the award of the degree of Master of Engineering in Information and Communication Engineering.

Name of Supervisor: DR. MD. ASHIKUR RAHMAN KHAN

Position: PROFESSOR& CHAIRMAN

Date:

Name of Co-Supervisor: ZAYED-US-SALEHIN

Position: ASSISTANT PROFESSOR

Date:

STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except for quotations, references and summaries which have been duly acknowledged. This project has not been accepted for any degree and is not concurrently submitted for award of other degree.

Jannatul Ferdous

Roll: BKH1811ME111F

Date:

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Professor Dr. Md. Ashikur Rahman Khan for his germinal ideas, invaluable guidance, continuous encouragement and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science, and his belief that a M.Sc. program is only a start of a life-long learning experience. I appreciate his consistent support from the first day I applied to graduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my native mistakes, and his commitment to my future carrier. I also would like to express very special thanks to my co-supervisor Zayed-Us-Salehin for his suggestions and co-operations throughout the study. I also sincerely thanks for the time spent proofreading and correcting my many mistakes.

My sincere thanks go to all my lab mates, members of the staff of the Information and Communication Engineering Department, NSTU, who helped me in many ways and made my stay at NSTU pleasant and unforgettable.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions which were crucial for the successful completion of this study.

ABSTRACT

UNIVERSITY HALL MANAGEMENT SYSTEM is a web based application intended for the students of all places which could be beneficial for them on the long run. The main objective of this application is to make it interactive and its ease of use. From this application students will be facilitated by providing the accommodation information. They will be informed about the residential facilities in hall whether they will reside there or not and if they can, they will be more specific by knowing room number, quantity of seats and contact info. Through this website students will be able to apply for the hall seat to the Hall provost.. The main emphasis lies in providing mail service, Students application first sent to department chairman as an anonymous message, after confirmation it will be sent to the provost, the confirmation mail will be sent to the students after the provost confirm the application. Only the registered chairman and hall provost can view the details of students and confirm their application via mail. Students can also receive the confirmation mail. Hall provost can add student in the hall if hall seats are available otherwise student cannot be allowed to hall. It will be hassle-free and less time consuming for the students. “UNIVERSITY HALL MANAGEMENT SYSTEM” is designed and implemented using the HTML (Hypertext Mark-Up Language), CSS (Cascading Style Sheet), PHP (PHP Hypertext Pre Processor), and MySQL database at the front-end and back-end. The system was tested after the development and found that it was appropriate to meet all the prerequisites indicated.

TABLE OF CONTENTS

SUPERVISORS DECLARATION	iii
STUDENTS DECLARATION	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xii

CHAPTER 1 INTRODUCTION

1.1	Introduction	1
1.2	Motivation of the project	1
1.3	Background	2
1.4	Problem Statement	3
1.5	Project Objectives	3
1.6	Scope of the Study	4
1.7	Challenges presented by the project	5
1.8	summary	5

CHAPTER 2 LITERATURE REVIEW

2.1	Introduction	6
2.2	Related Work	6
2.3	Summary	8

CHAPTER 3 SYSTEM REQUIREMENT AND FEASIBILITY STUDY

3.1	Introduction	9
3.2	Software Requirements	9
3.3	Operating System	11

3.4	Feasibility Study	11
3.4.1	Technical Feasibility	11
3.4.2	Economic Feasibility	12
3.4.3	Operational Feasibility	12

CHAPTER 4 PROJECT DESIGN AND METHODOLOGY

4.1	Introduction	13
4.2	Design	13
4.3	Project module	13
4.4	Methodology	19
4.4.1	Planning	19
4.4.2	Designing	20
4.4.3	Coding	20
4.4.4	Testing	21
4.4.5	Listening	21
4.5	Diagram	22
4.5.1	ER Diagram	22
4.5.2	Use case diagram	24
4.5.3	Activity diagram	27
4.5.4	Data flow diagram	31
4.6	Database design	36
4.6.1	Design process	36
4.7	Summary	38

CHAPTER 5 RESULTS AND DISCUSSION

5.1	Introduction	39
5.2	User interface	39
5.3	Registration and login implementation	40

5.4	Home page	41
5.5	Dashboard	41
5.6	Edited/Managed items	43
5.7	Final outlook	43
5.8	Result	49
5.9	Scope and limitations	50
5.10	Summary	50

CHAPTER 6 CONCLUSION

6.1	Introduction	51
6.2	Conclusion	51
6.3	Future scope of the project	52

REFERENCES		53
-------------------	--	----

APPENDICES

A	Coding	54-64
---	--------	-------

LIST OF FIGURES

Figure No	Title	Page
4.1	Administrator module	15
4.2	User module	16
4.3	Chairman module	17
4.4	Provost module	18
4.5	Staff module	18
4.6	ER Diagram	23
4.7	Use case diagram for admin	24
4.8	Use case diagram for user	25
4.9	Use case diagram for chairman	26
4.10	Use case diagram for provost	27
4.11	Activity diagram for admin	28
4.12	Activity diagram for user	29
4.13	Activity diagram for provost	30
4.14	Context diagram	31
4.15	Data flow diagram for admin	32
4.16	Data flow diagram for user	33
4.17	Data flow diagram for chairman	34
4.18	Data flow diagram foe provost	35
5.1	Admin registration	40
5.2	Login layout	40
5.3	Home page	41
5.4	Dashboard for admin	42
5.5	Dashboard for department chairman.	42

5.6	Dashboard for hall provost	42
5.7	Department chairman addition form	44
5.8	Hall provost addition form	44
5.9	Departments addition form	44
5.10	Halls addition form	45
5.11	Department chairman details.	45
5.12	Hall provost details.	45
5.13	Departments added by admin	46
5.14	Halls added by admin	46
5.15	Hall seat application form	47
5.16	Students details for department chairman and hall provost	47
5.17	After confirm student request by department chairman	47
5.18	View student request for hall seat	48
5.19	After confirm student request by hall provost	48
5.20	Room and seat detail	48
5.21	Add student to room	48
5.22	Seat confirmation message for student	49
5.23	Seat allotment for student	49
5.24	Payment option	49

LIST OF ABBREVIATIONS

HMS	Hall Management System
DFD	Data Flow Diagram
UML	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In this chapter, the introductory study of the project is performed. As this project is focuses on the University hall management system; hence background study on these sectors are performed first. Then the problem statement is presented and based on this, the project objectives are defined. At the end of this chapter, the scope of this project is presented.

1.2 MOTIVATION OF THE PROJECT

The purpose for our inspiration is the present patterns of web application reconciliation and intuitive highlights. There was a time when people used to waste their valuable times just to get a portion of information. Now the technology is more Advance than compare to any previous times. One of the blessings of technology is web application. It allows users to interact with the system from anywhere as long as they are connected to the internet. As we already discussed above that internet has made the user's interaction through the system easier, so this web application can connect to respective servers for accessing data which will surely help students to get proper information or reserve hall seat for accommodation online without physical labor. We are sitting back where the innovation getting to be more intelligent. We would prefer not to squander our occasions by doing pointless difficult work and attempt to accomplish something gainful with the assistance of innovation to spare our opportunity. I have the most enthusiasm for helping individuals from my youth. This urge me a great deal to make an immaculate site. One of the primary motivations behind the site is to encourage the disconnected understudies online in light of the fact that understudies can't invest their valuable energy in outside endeavoring to discover the best arrangement.

1.3 BACKGROUND

As in the modern world whole data are processed digitally, the computer technology users bring the increasing need for electronic services and their security. One of the things that changes is the technological advancement that makes people feel, comfortable and meaningful. In this paper, the development of University hall management system is proposed. For the past few years the number educational institutions are increasing gradually. Thereby the halls are also increasing for the accommodation of the student studying in this institution. And hence there is a lot of attestation on the person who are managing the hall and documents are not usually used in this context. All these halls at present are managed manually by the authority. This particular project deals with the problem on managing a hall and avoids the problems which occur when carried manually.

This website is organized by the Admin. The whole work is managed by the admin. Admin add and manage department chairman, Hall provost, Hall stuff. In this web based system admin provides Hall related academic information including notices, Hall eligibility, and the quantity of seats of the Hall individually. Students will be facilitated by providing the accommodation information. They will be informed about the residential facilities in hall whether they will reside there or not and if they can, they will be more specific by knowing room number, quantity of seats and contact info. Through this website students will be able to apply for the hall to the Hall provost and they will also receive confirmation mail from the provost on whether they will get a seat. Identification of the limitation of the existing system to the designing of computerized system that will be the congenial to the existing system which is more user oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

- more human error
- Database redundancy
- difficult to handle
- Repetition of the same tasks
- More time consumption
- Low security and difficult to allotment

1.4 PROBLEM STATEMENT

Hall Management System (HMS) is a system which helps in managing various activities in the Hall. Many works that need to do manually by the hall provost, office staff. There is a great cost and physical Diligence associated with collecting the precise information and Arrange accommodation for another zones students.

- i. Earlier, students were deprived of a lot of information in a timely manner, especially in rural areas. They need to go to the institution for collect information. Through this website students will be able to know hall related all the information such as rules, system of application, seat quantity, contact info about provost, stuff.
- ii. In the previous system, hall office staffs need to record all kind of information about living student and process manually which involves more man power.
- iii. Earlier students, provosts and stuff had to be harassed to allocate seats. Students had to stand in line for hours.
- iv. Registration for hall allocation is done manually, thus over time, handling of paper files becomes cumbersome and untidy as the number of students gradually increase.
- v. In addition, applications for allotment of hall seats had to be submitted by hand in the hall office room which takes a long time to reach the provost and the students wait anxiously. Through this website, Students will be able to submit an application to the provost and they will be able to know through email whether they will get a seat and reduce their wastage of time.

From that point of view, we are thinking university hall management system. The proposed system is such a system.

1.5 PROJECT OBJECTIVES

The principle goal of this task is to collaborate with students and their parents with the necessary information so that they do not face any embarrassment. The hall management system application can highlight the necessity of booking seats.

- i. To develop such a website that interested student get easy way to find their information and accommodation and manage a proper database system to ensure data safety and proper management.
- ii. To make easier the hall apply system, simple, reliable, and corrective for the students. Moreover, less time consuming as compared to manual work.
- iii. Provide an interface that will allow the administrators to manage: chairman, provost, stuff, application for hall, add department, hall, room number and seat, notices.
- iv. Provide an interface that will facilitate user apply for the hall, they can write an application to the provost and submit.
- v. Provide an interface that will allow the department chairman and hall provost to confirm the student request for the hall seat.
- vi. Provide an interface that will allow the students to receive confirmation mail from provost

1.6 SCOPE OF THE STUDY

The scope of the project is implementation of hall seat confirmation and allocation through email. Technology is more advances than compare to any previous times. One of the blessings of technology is web application. It allows users to interact with the system from anywhere as long as they are connected to the internet. So this web application can connect to respective servers for accessing data which will surely help students to get proper information or reserve hall seat for accommodation online without physical labor.

- i. This website is to make easier the hall apply system, simple, reliable, and corrective for the students. Moreover, less time consuming as compared to manual work.
- ii. Secure storage and retrieval of user details from the database.
- iii. Enable secure login of user, that is to say non- legitimate user should never be allowed to login to the tool.
- iv. Maintaining and manipulating records in database through functions like user registration, Hall apply, seat allocation, hall room registration and student reports.
- v. Validate and verify input and output data.

- vi. Fast processing and immediate results with high security.

1.7 CHALLENGES PRESENTED BY THE PROJECT

To find out the gap of relevant studies conducted earlier, literatures have been reviewed to explore the potentiality of the apply system. Then the study is conducted by collecting data from the secondary sources and then trend of accommodation information, apply for hall are analyzed with the help of collected data. The challenges of the system are identified and some policy implications are provided to address the challenges as well.

1.8 SUMMARY

Finally, in this chapter we discuss about how we developed an application for University student to get accommodation facility easily.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

A literature review discusses the published information in a particular subject area, and sometimes information in a particular subject area within a certain time period. It can be just a simple summary of the sources, but it usually has an organizational pattern and combines both summary and synthesis. The University Hall Management system is a system that manages the details of the student staying in the hall, application details, seats and room details. Halls are the part of the institutions which will help the students to stay here who are coming from far place and who are foreign student.

2.2 RELATED WORK

Honey Alwani proposed online hostel management system [Developed on Sep 11, 2015]. This web application describes the hostel management to improve their services for all the students of the hostel. This also reduce the manual work of the persons in admin panel and the bundle of registers that were search when to find the information of a previous student, because through this system can store the data of those students who had left the hostel. Through this you can check the personal profile of all the current students. The students of the hostel will be recognized from the ID number allocated at the room rental time. In the last this system will improve the management work in the hostel [1].

Henry Peter Gommans, George Mwenda Njiru, Arphaxad Nguka Owange proposed Rental House Management System [Developed on 11, November 2014] they develop a rental house management system that can solve all the problems experienced with the current manual system. When the user login to the system automatically show three forms: houses form, rent payment form and tenants" registration form. Each form has several command buttons; new, save, cancel,

delete, next, previous and exit. With the command buttons user can manipulate the database. If user want to view data in the database you just click next or previous and the data will be displayed for them [2].

Rahul Vandary proposed Hostel Management System [Developed on April,2009] he develop a institution hall management system that can solve the existing system. Admin can allot different student, control the status of the fee payment, edit the details of the students and update their record, add notice of hall. In his system all work was done by admin [3].

Oliwagbemiga Shoewu and Olsegun Omitola proposed an Automated Hostel Facility Management system [Developed on September 23, 2014] they developed an automated hostel facility management system. When admin log in to the system he can allot student, determine the students payment of the fee. There have student module and mess module [4].

College Hostel Management Software developed by Initio has six modules such as the library module, the transport module, the hostel module, the inventory module, the enquiry module and the visitors tracking module [5].

Above, all those sites we cannot find all facilities together. There is no confirmation mail service for the user (Chairman, Provost, Students). In our project students can find all the services together without any problem. Our website is different from others because it simplifies all information like residence hall information, hall eligibility, Room number, availability of seats, capacity of the room, contact info etc.

- In my project it gives enough information about accommodation.
- Fast processing and immediate results with high security.
- Minimizing human effort and cost efficient databases.
- Navigation through the site is easy.

Since the reservation is made in real time, the user receives immediate assurance that what he wants is or is not available. Also, because user conduct their own transactions and receive confirmations, they are free to make changes during or even after the process, with a minimum of complicating factors. As we do not have any informative website about hall, So we are inspired to make such a website. We are hopeful that students will be facilitated by this website.

2.3 SUMMARY

This system is designed in convenience of the hall management which helps them to maintain the records of the user of this system. It helps them from the manual work from which it is difficult to find out the records of the students, room capacity, available of seats. In the existing system user can't apply for hall seat through online, the action is perform manually which was time consuming. But in the proposed system user can apply for hall with their details and receive confirmation mail about seat allocation. They no longer have to wait anxiously.

CHAPTER 3

SYSTEM REQUIREMENT AND FEASIBILITY STUDY

3.1 INTRODUCTION

In this chapter we discuss what kind of software is used for designing this project. The system is designed with the help of HTML, CSS, JavaScript, Bootstrap. At the end of this chapter the feasibility study of this project is presented.

3.2 SOFTWARE REQUIREMENTS

FRONT END: HTML, CSS, JavaScript, Bootstrap

HTML (Hyper Text Markup Language)

HTML is the standard markup language for creating web pages. HTML stands for Hyper Text Markup Language. HTML describes structure of web pages using markup. HTML elements are represented by tags. With cascading Style Sheet and JavaScript it forms a triad of cornerstone technologies for the World Wide Web.

CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) is a language that describes the style and presentation of an HTML or XML document. CSS describes how HTML element should be displayed on screen, on paper, in speech or on the media.

JAVASCRIPT

JavaScript is a scripting or programming language that allows for implement complex things on web pages. Every time a web page does more than just sit there and display static information for user to look at displaying timely content updates, interactive maps, scrolling video jukeboxes etc.

BOOTSTRAP

Bootstrap is the most popular HTML, CSS and JavaScript framework. It is also a front-end development framework that enables us to quickly build fully responsive websites.

BACK END: PHP, MySQL

PHP (Hypertext Preprocessor)

Hypertext Preprocessor (PHP) is a technology that allows software developers to create dynamically generated web pages, in HTML, XML, or other document types, as per client request. PHP is open source software.

MySQL

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language. It is platform independent and therefore can be implemented and used across several such as Windows, Linux server and is compatible with various hardware mainframes. It allows combination, extraction, manipulation and organization of data in the database. It is platform independent and therefore can be implemented and used across several such as Windows, Linux server and is compatible with various hardware mainframes. It is fast in performance, stable and provides business value at a low cost.

3.3 OPERATING SYSTEM

For Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional/Windows 10 installation:

- o **Processor:** 800 MHz Intel Pentium III or equivalent
- o **Memory:** 512 MB
- o **Disk space:** 750 MB of free disk space

□ **For Ubuntu 9.10 installation:**

- o **Processor:** 800 MHz Intel Pentium III or equivalent
- o **Memory:** 512 MB
- o **Disk space:** 650 MB of free disk space

3.4 FEASIBILITY STUDY

A feasibility study is used to determine the viability of an idea, such as ensuring a project is legally, operationally and technically feasible as well as economically justifiable. Output of feasibility study summarizes results of the analysis and evaluations conducted to review the proposed solution and investigate project alternatives for the purpose of identifying if the project is really feasible, cost-effective and profitable. In this section discussed about the feasibility study of project.

3.4.1 TECHNICAL FEASIBILITY

Every system has to need technical feasibility. The propose system has to technical feasibility though the project is design with PHP as code behind and MySQL as backend, it is easy to install in the systems whenever needed. It is more efficient, easy and user friendly to understand by almost everyone. Huge amount of data can be handled efficiently using MySQL as backend. Hence this project has good technical feasibility. As per the technical aspect this project is technically suitable for the user as well as the developer.

3.4.2 ECONOMICAL FEASIBILITY

Economic feasibility is mainly concern with the cost incurred during the implementation of the software. Since the project is developed using PHP and MySQL which is more commonly available and even the cost involved in the installation process is not high. Similarly it is to recruit persons for operating the software. Since almost all the people are aware of PHP and MySql. Even if we want to train the person in these areas, the cost involved in training is also very less. Hence this project has good economic feasibility.

3.4.3 OPERATIONAL FEASIBILITY

As this PHP component is very easy to operate for the software engineers and easy to use. So this project has a good operational feasibility. Hence this project is technically, economically as well as operationally feasible.

CHAPTER 4

PROJECT DESIGN AND METHODOLOGY

4.1 INTRODUCTION

In this chapter, the proposed model describes about the elaboration of the whole system such as design, architecture and data flow. As I mentioned above that the constituent of the system is software. The system is designed with the help of standard System Architecture that describes software system in terms of their software components and their interactions. Illustration of tools and materials required to implement the system and the working procedure of each modules comprising the system are discussed in detail.

4.2 DESIGNE

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. We used many of programming language and graphics tool to design the web application. The background and feature is designed by Adobe Photoshop, illustrator and HTML, CSS. And backend programming is done by PHP. The Database is designed by JavaScript and Mysql.

4.3 PROJECT MODULE

A module is a collection of source files and builds settings that allow you to divide your project into discrete units of functionality. Your project can have one or many modules band. One module can use another module as a dependency. Each module can be indecently built, tested, and debugged. Module provides a container for your project's source code, resource file, and

such as the module-level build file. Each module can be independently build, tested, and debugged.

The Project outcome consists of eleven main phases:-

- i) USER REGISTRATION:** The first part is to registered user. At first the user registers the website, then they log in with their password.
- ii) AUTHENTICATION:** In this part validate the registration page. People who unregistered are not eligible for access.
- iii) APPLY FOR SEAT:** Here student can apply for seat with their personal information.
- iv) ADD CHAIRMAN:** Here admin can add chairman and manage chairman
- v) ADD PROVOST:** Here Admin can add provost and manage provost.
- vi) ADD ROOM:** Here registered stuff can add new room with number of seat.
- vii) ADD STUFF:** Here Admin can add hall stuff
- viii) NOTICE BOARD:** All the halls have their notice boards. Any change in the hall will be shown in this.
- ix) ADD HALL:** Here Admin can add hall.
- x) CHECK REQUEST:** Here department chairman and hall provost view hall application and approve application.
- xi) SEAT QUANTITY COUNTING:** After booked a room, automatically counted rest of the seat quantity.

The system after careful analysis has been identified to be presented with the following modules and role. The modules involved are:

- Administrator
- User
- Chairman
- Provost
- Staff

Administrator Module

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the system. The administrator has all the information about all the users.

This module is divided into different sub-modules.

- Manage information.
- Manage chairman.
- Manage provost.
- Manage stuff
- Manage booking

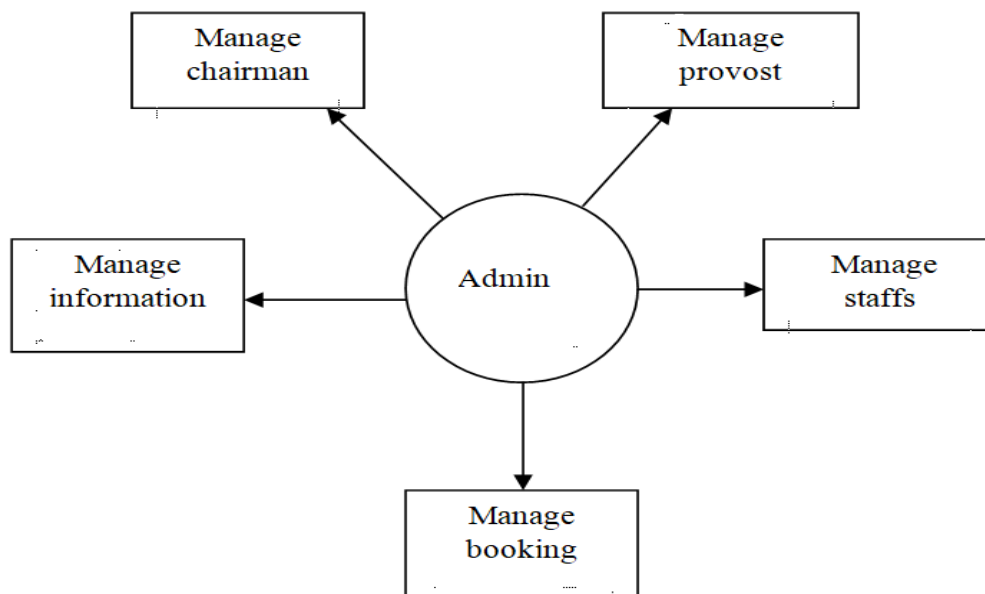


Figure 4.1: Administrator module

User Module

A new user will have to register in the system by providing essential details in order to view the information in the system Here user is student. User can

- Apply for hall seat.
- View confirmation mail.
- View seat allotment.
- Pay hall fee

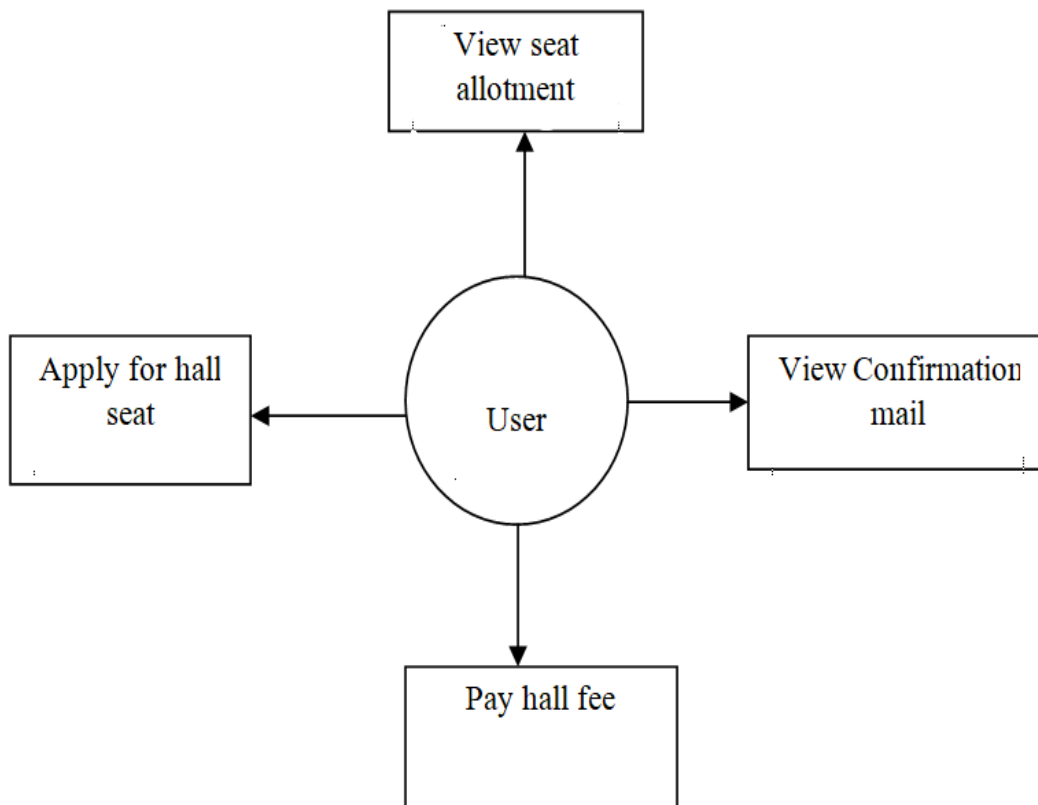


Figure 4.2: User module

Chairman Module

Chairman must login with his user name and password to the system. Chairman can

- View Hall application request.
- View total student.
- Confirm request.

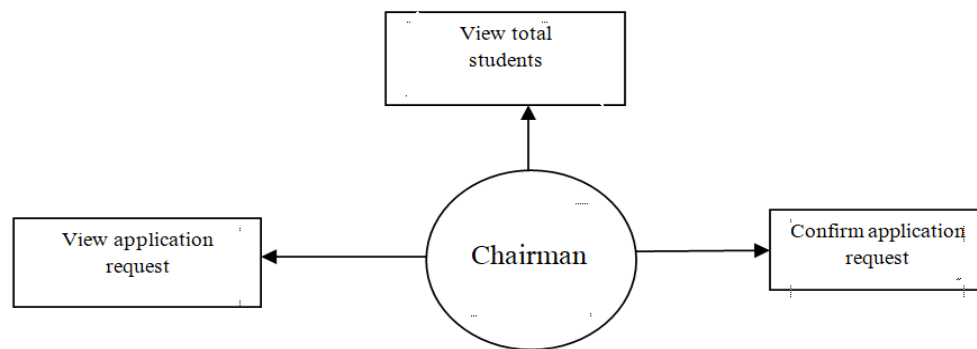


Figure 4.3: Chairman Module

Provost Module

Provost must login with his user name and password to the system. Provost can

- View hall application request.
- View total student.
- View total seats.
- View available seats.
- Confirm request.

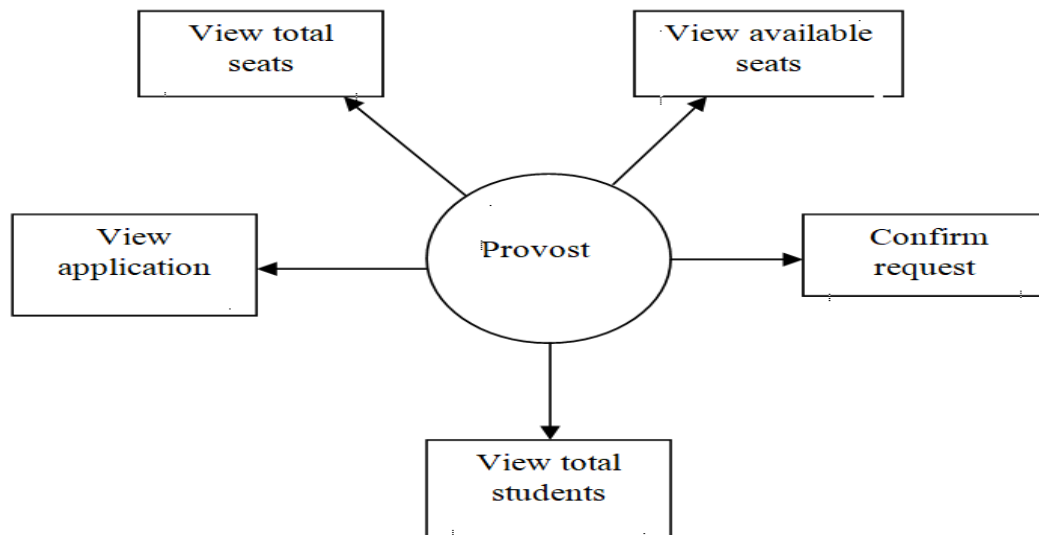


Figure 4.4: Provosts module

Staff Module

Staff must login with his user name and password to the system. Staff can

- Add room
- Add notice

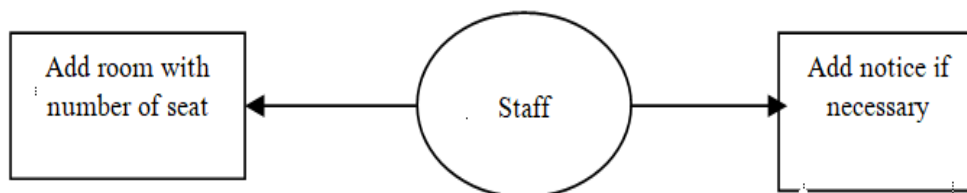


Figure 4.5: Staff module

4.4 METHODOLOGY

Change in software requirements is a common in every software development. As a developer we need to make sure that whatever the changes take place our system should meet the overall requirements of the stakeholders. Here we are going to choose XP (Extreme Programming) methodology to develop this system.

Extreme Programming is one of the agile methodology for system development. Here changes in requirements while iteration is possible compare to others like (Scrum, RAD etc.) [7]. It improves the quality and responsiveness of system application. Back in the 1990s, the rise of the Internet necessitated a change in software development. If a company's success depended on the speed at which the company could grow and bring products to market, businesses needed to dramatically reduce the software development life cycle. Extreme programming also introduces a number of basic values, principles and practices on top of the agile programming framework [8].

There are 5 phases in this agile methodology. They are:

- Planning
- Designing
- Coding
- Testing
- Listening

4.4.1 PLANNING

The first step we are going to take is planning. Here the users of our system are Admin, volunteer and the registered user. We will convert their requirements into the iteration which is usually a small part of the system functionalities. Here we will also focus on dead line of the iteration so that we will be able to complete the requirements on time. This stage is where the UX magic happens. User don't need to be exhaustive or overly technical—they only need to provide enough detail to help the team determine how long it'll take to implement those features. With Lucid chart, user can create a basic flowchart and easily record and share the desired

functionality from there, the team creates a release schedule and divides the project into iterations (one to three weeks long).

4.4.2 DESIGNING

After the planning we will start designing the overall system. Here we will include object-oriented design models (UML) to illustrate the overall functionalities of the system. This design models will show the active users of the system and their involvement in every process. As for examples: registered student login, user registration, write application, submit and so on. From the point of view of simplicity, of course one could say that system development doesn't need more than coding, testing and listening.

If those activities are performed well, the result should always be a system that works. In practice, this will not work. One can come a long way without designing but at a given time one will get stuck. The system becomes too complex and the dependencies within the system cease to be clear. One can avoid this by creating a design structure that organizes the logic in the system. Good design will avoid lots of dependencies within a system; this means that changing one part of the system will not affect other parts of the system.

4.4.3 CODING

Then the time finally comes to implement code. XP practices collective code ownership: Everyone reviews code and any developer can add functionality, fix bugs, or refactor.

As the project is web based, we will use HTML, CSS, and JavaScript for front end and PHP for back end programming, for database storing MYSQL. It will also help us to reduce the conflict and optimize the functionalities of the system. For collective code ownership to work, the team should:

- i. Choose a system metaphor (standardized naming scheme).
- ii. Practice pair programming. Team members work in pairs, at a single computer, to create code and send it into production. Only one pair integrates code at a time.

- iii. Integrate and commit code into the repository every few hours. The customer should be available, preferably on site, during this entire process so they can answer questions and establish requirements.

4.4.4 TESTING

Here we will do testing while developing rather than waiting for ending the overall development phases. We will test the codes which will lead me to eliminate the system bugs and make sure that the code passes all the unit test before we release it to our stakeholders. Besides, I will also take care of user's acceptance testing which is completely based on what I gathered the system requirements from stockholders. I will provide the results of the testing and also demonstrate the system to make it more familiar and user friendly. Testing, also known as component testing refers to tests that verify the functionality of a particular section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

4.4.5 LISTENING

One of the advantages of using this methodology is involvement of our stakeholders. We will involve users in development phases just to receive their feedback regarding the requirements and the functionalities of the system. In every iteration user will provide their feedback and when they satisfied we will proceed to next iteration. Programmers must listen to what the user need the system to do, what "business logic" is needed. They must understand these needs well enough to give the user feedback about the technical aspects of how the problem might be solved or cannot be solved. Communication between the user and programmer is further addressed in the planning game.

4.5 DIAGRAM

A diagram is a valid expression, using visual elements, of details. Diagrams have been used on the cave walls since ancient times, but during the Revolution, they became more widespread. The method uses for three-dimensional representation that is projected onto a two-dimensional layer often. Sometimes the term graph is used as a synonym for diagram.

4.5.1 ER DIAGRAM

ER diagram which shows that helps to construct databases effectively. Characteristics of ER diagrams are generally modeled as a rectangle with the label of the attribute, connected to the person or interaction that includes the attribute. The end product can typically be classified into two further steps within the hierarchical model to determine the grouping of information within the structure, generally determine the fundamental objects where the data is processed, and then determine the relationships between such information groups or entities. The Entity-Relationship Diagram (ERD) is a graphic illustration of various data that explains how these data are connected using concepts. Specific databases that are used in software development and IT networks are most commonly associated with ER diagrams. During the design phase of a development cycle, ER diagrams are often used to describe various system elements and their interactions with each other. By displaying the relationships among entity sets in such a system, an Entity-Relationship (ER) diagram demonstrates the logical structure of a database. ER diagrams are often used because they help to organize data contained in a database and can be converted into tables directly. A connection is an interaction between two or more sets of individuals. In an ER diagram, a relationship is defined by a diamond. Diagram Entity-Relationship (ER) is a pictorial arrangement of information in databases, showing how data is associated. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity. There is a tradition. Figure 4.6 shows the ER diagram for this system

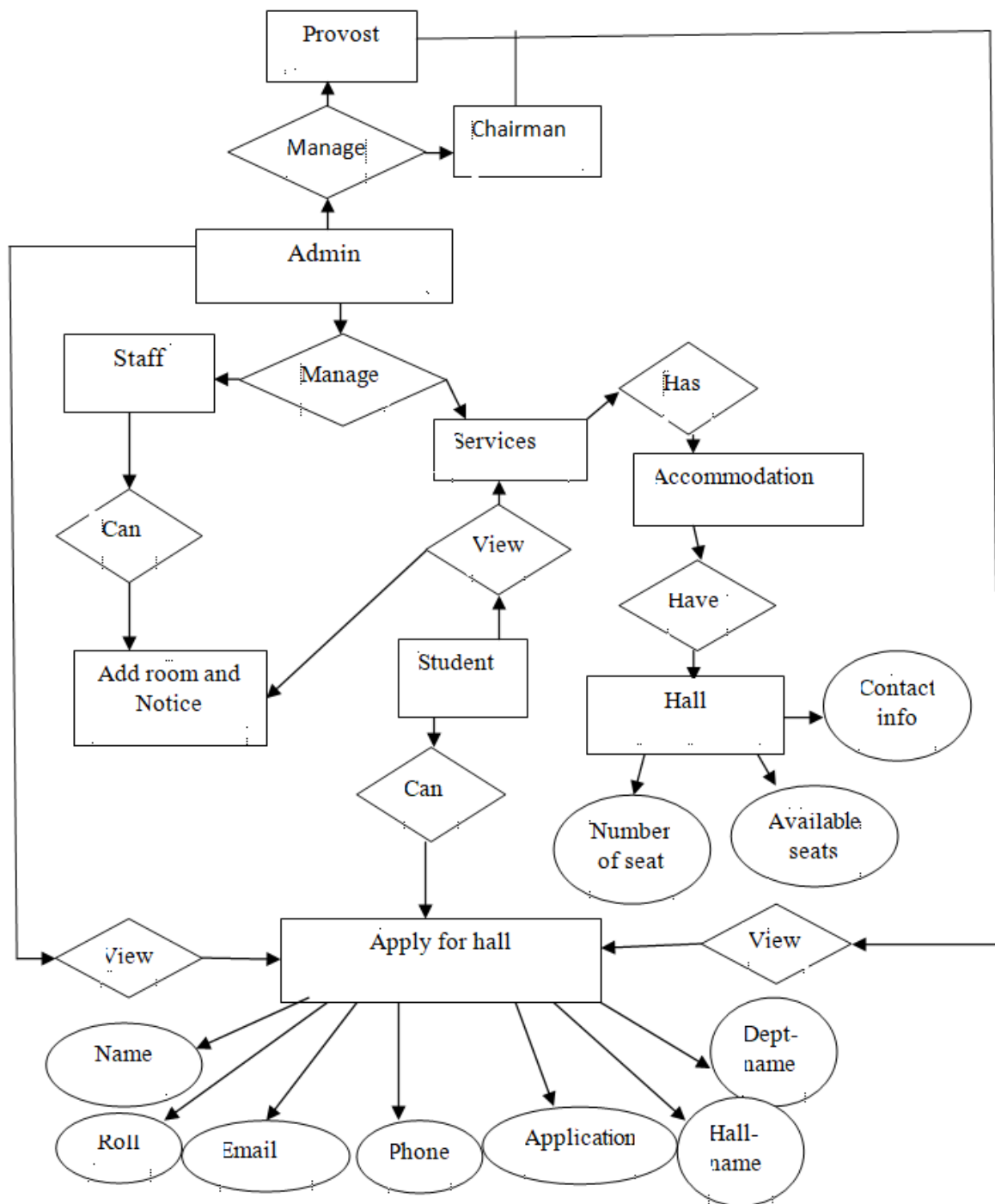


Figure 4.6: ER Diagram

4.5.2 Use Case Diagram

A use case defines the interactions between external actors and the system under consideration to accomplish a goal. Actors must be able to make decisions but need not be human: "An actor might be a person, a company or organization, a computer program, or a computer system-hardware, software, or both.

(a) Admin Diagram

Here in Use case diagrams consist of 2 objects, Actor and Use case. Here admin is an actor that who performs a role in one given system. Administrator Use Case Diagram use only for admin who is responsible for maintaining the project by inputting result or update time to time. Administrator use case diagram is shown in figure 4.7

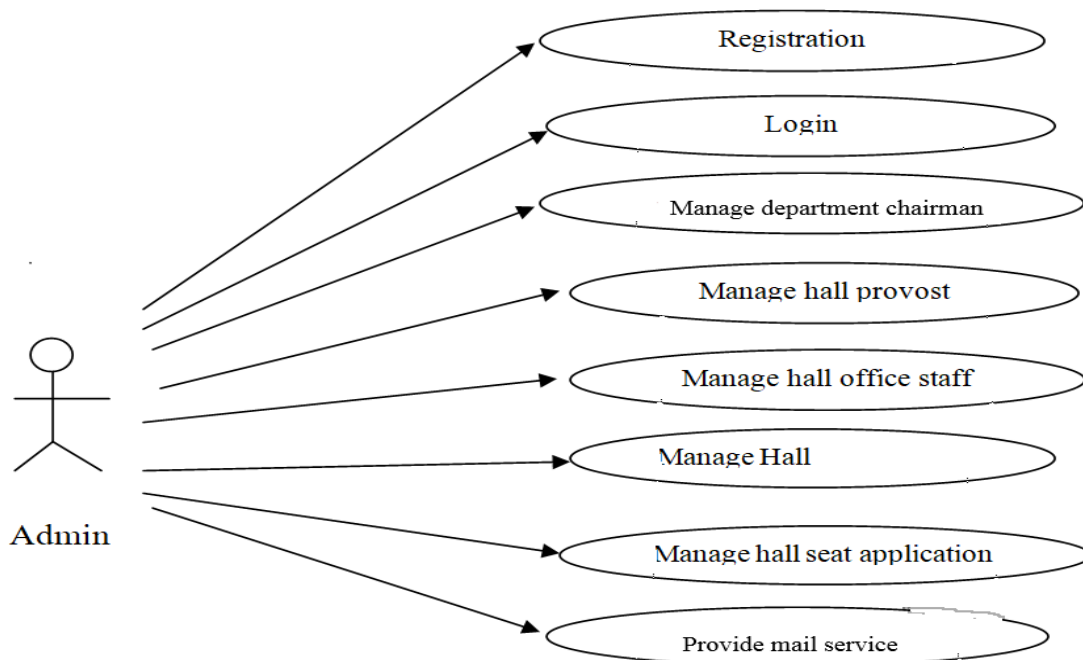


Figure 4.7: Use case diagram for admin

(b) User Diagram

User use case diagram represents that how a user will get facilities in this system. Here user is only student. They can view information such as residence hall by clicking each name of hall where they can see hall name, hall category, seat quantity, contact info, notice board of hall. They can apply for hall seat, receive confirmation mail from provost and view their seat allotment. User use case diagram is shown in figure 4.8

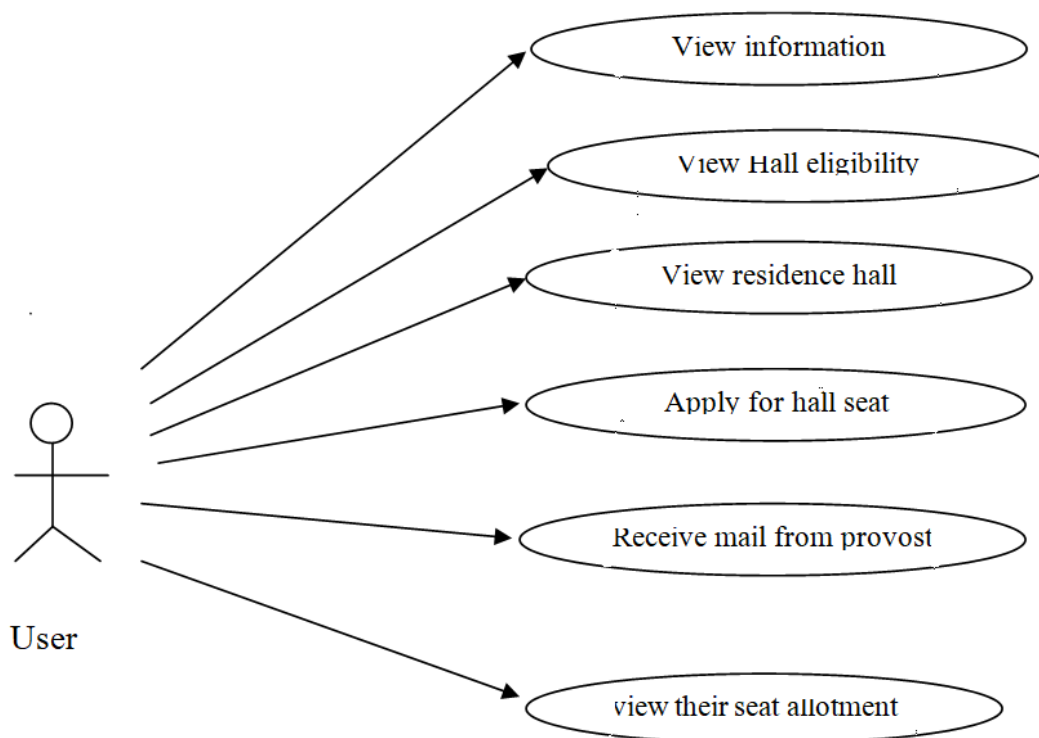


Figure 4.8: Use case diagram for user

(c) Diagram for Chairman

Use Case Diagram for chairman represent that how department chairman perform his tasks in this system. Chairman first log in to the system then he can view the student request for hall seat, view details of student and confirm their request. Then he should log out from the system. Figure 4.9 shows the use case diagram for department chairman.

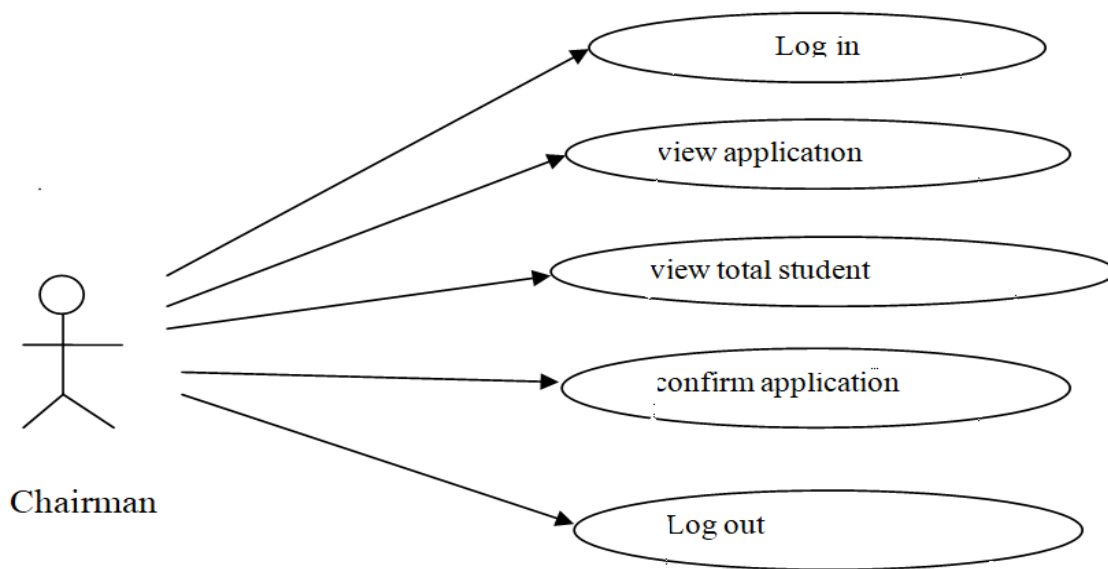


Figure 4.9: Use case diagram for chairman

(d) Diagram for Provost

Use Case Diagram for provost provides an idea of what the provost does in this system. Provost first login to the system then he can view student request for hall view their details and confirm student request. Then the provost entered the room number, view total seats, available seat and add student to room if there is any empty seat. Figure 4.10 shows the use case diagram for hall provost.

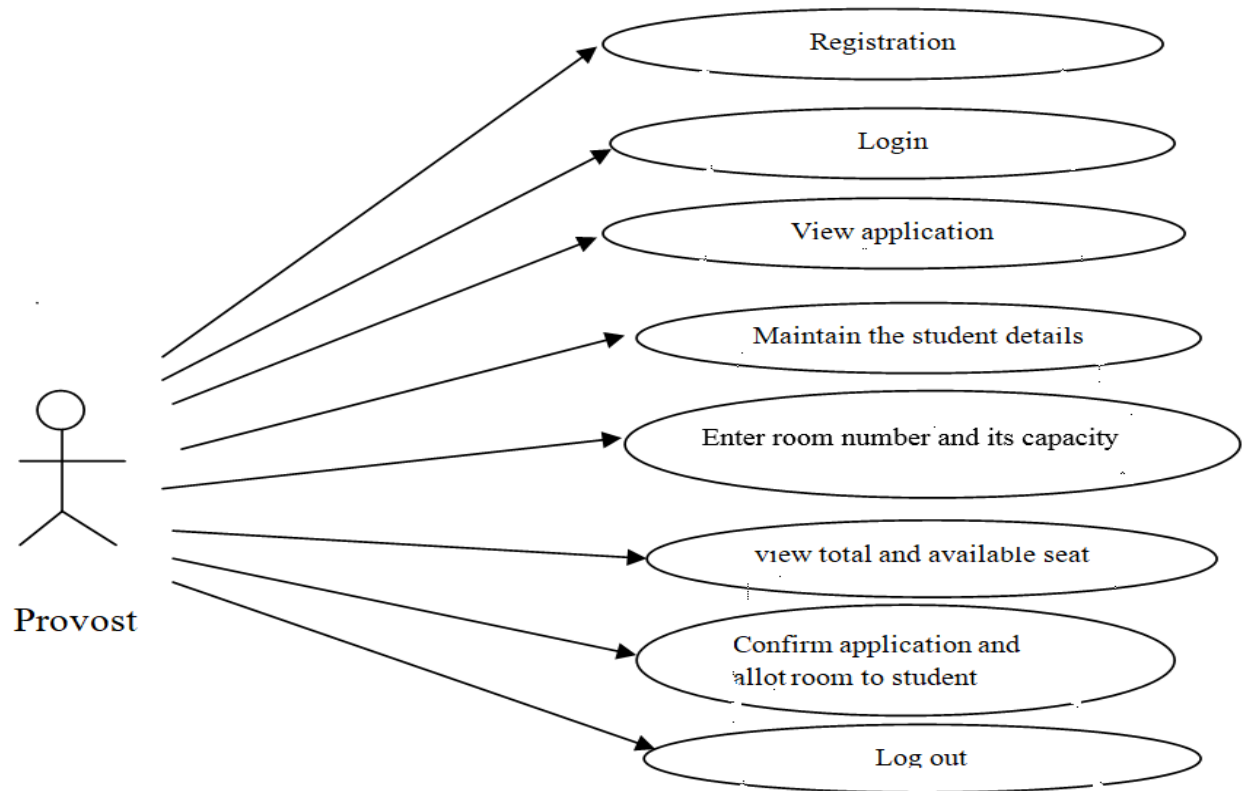


Figure 4.10: Use case diagram for provost

4.5.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. The basic purposes of activity diagrams are similar to other diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

(a) Diagram for Admin

Administrator Activity Diagram uses only for admin who is responsible for maintaining the project by inputting result or update time to time. Here in the activity diagram start the activity diagram with the start node. The first step is to login into the application so that processes the activity name as login which is connected by the control flow arrow. Admin should register to the system to login into the application which is the precondition. If login success displays the dashboard of application then the next process is to update all the activity and maintain the system. The administrator activity diagram is shown in figure 4.7

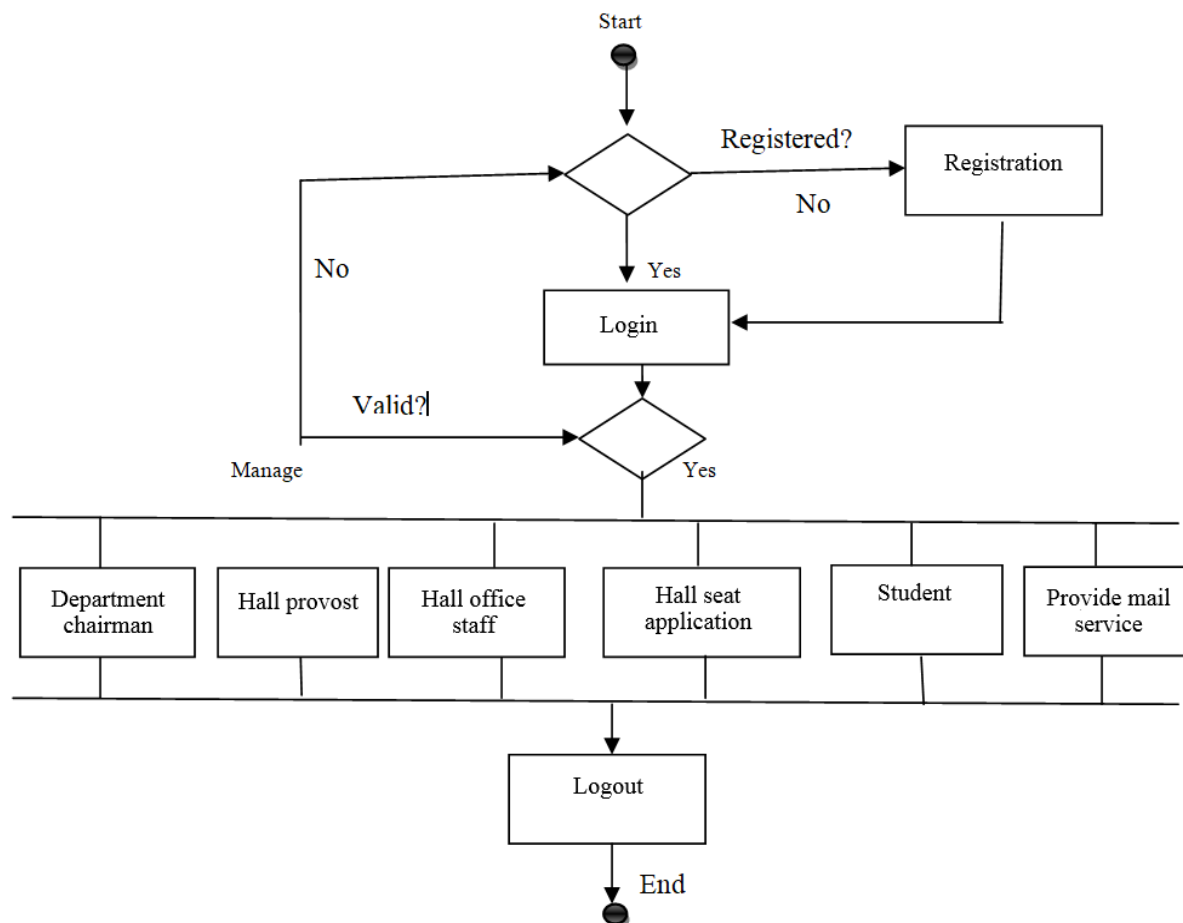


Figure 4.11: Activity diagram for admin

(b) User Diagram

User Activity diagram represents that how a user will get facilities in this system. Here user is only student. Here in the activity diagram start with start node. The first step is to login into the application so that processes the activity name as login which is connected by the control flow arrow. Users should register to the system to login into the application which is the precondition. If login success displays the dashboard of the application then the next process is to view all the activity and they can find their desired information by clicking the desire section. All students can apply for hall seat but who is unregistered cannot view the confirmation mail. The user activity diagram is shown in figure 4.12

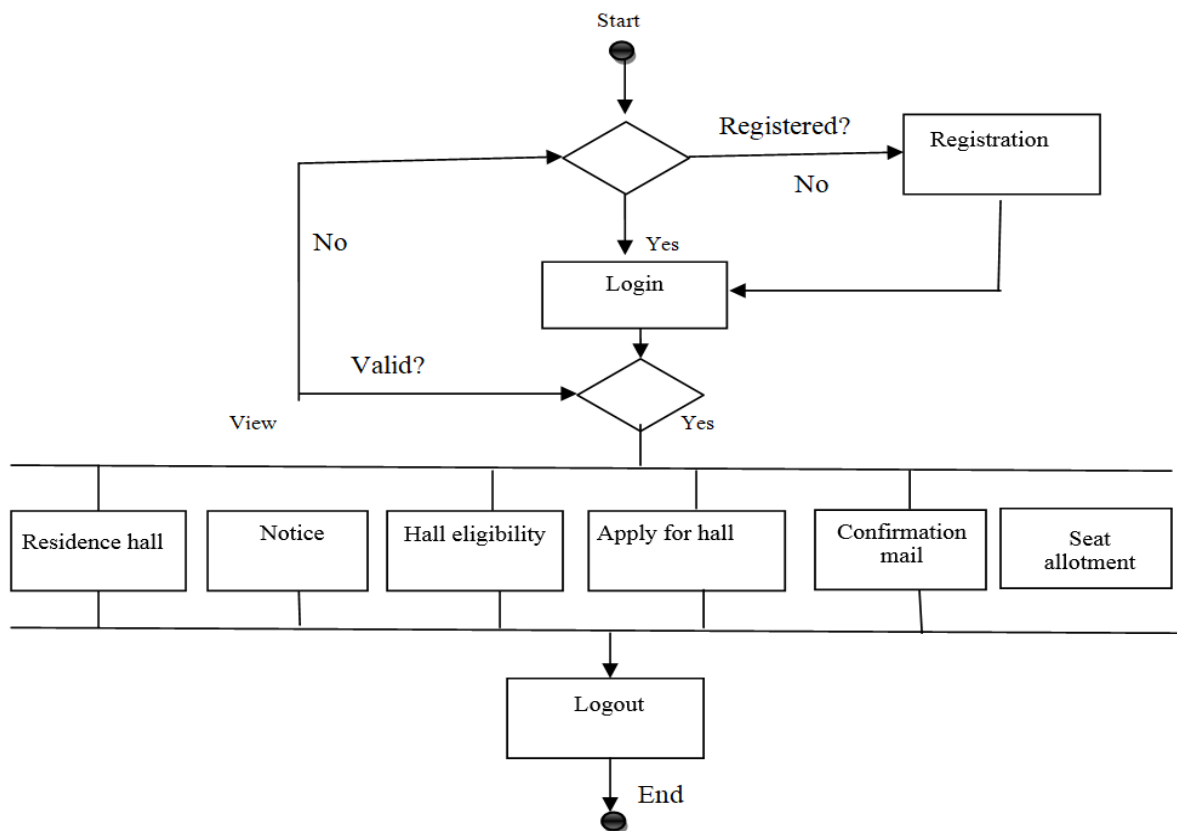


Figure 4.12: Activity diagram for user

(c) Diagram for Provost

Here in the activity diagram start the activity diagram with the start node. The first step is to login into the application so that processes the activity name as login which is connected by the control flow arrow. Provost should registered by the admin to the system to login into the application which is the precondition. If login success displays the dashboard of application then the next process is to update all the activity and maintain the system. The Activity diagram for hall provost is shown in Figure 4.13

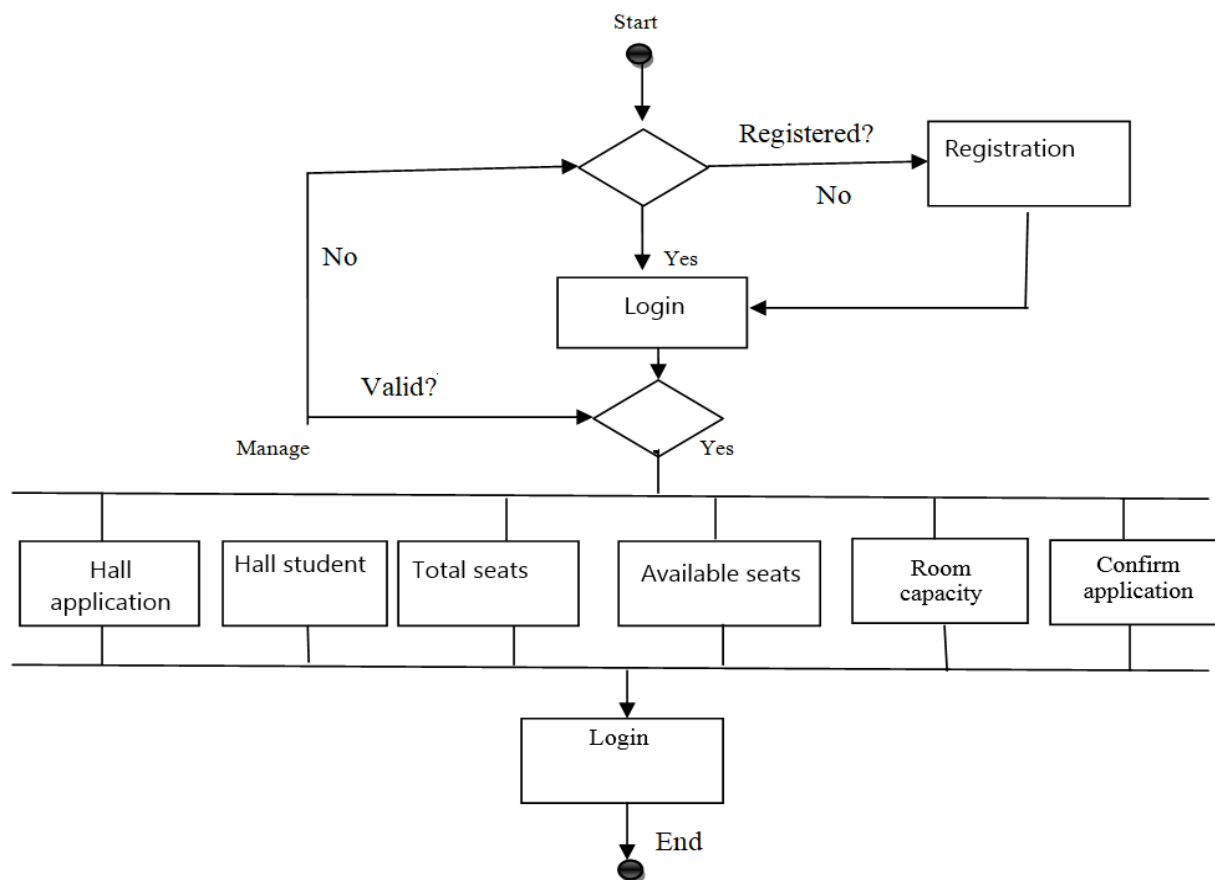


Figure 4.13: Activity diagram for provost

4.5.4 DATA FLOW DIAGRAM

DFD may show a number of processes with data flowing into and out of each process. If there is a need to show more detail within a particular process, the process is decomposed into a number of smaller processes in a lower level DFD. In this way, the Content Diagram or Context-Level DFD is labeled a “Level-0 DFD” while the next level of decomposition is labeled a “Level-1 DFD”, the next is labeled a “Level-2 DFD”, and so on.

Level 0 DFD: Context model

Context Diagrams and Data-Flow Diagrams were created for systems analysis and design. But like many analysis tools they have been leveraged for other purposes. For example, they can also be leveraged to capture and communicate the interactions and flow of data between business processes. So, they don’t have to be restricted to systems analysis.

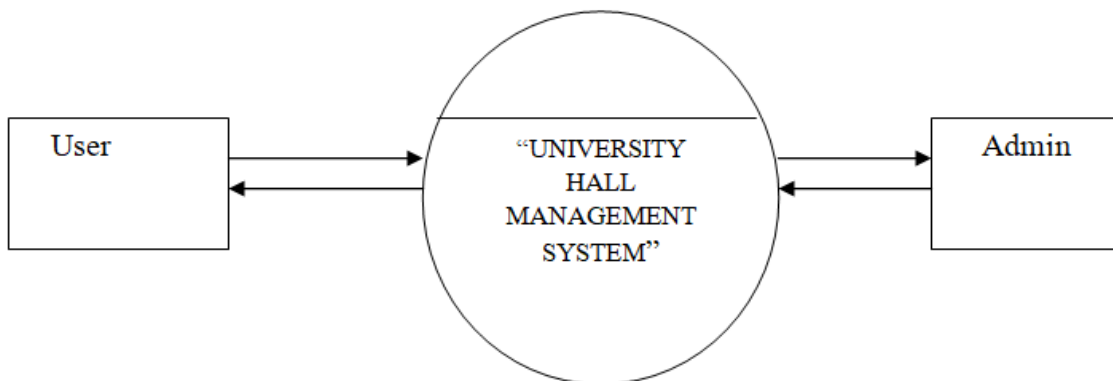


Figure 4.14:Context diagram

Level 1 DFD Admin-data flow diagram

The Admin side DFD describes the functionality of Admin. Admin is a responsible person who runs the project. After login to the system admin can first manage chairman, provost, staffs, students, and maintain the whole tasks. The admin level 1 data flow diagram is shown in Figure 4.15

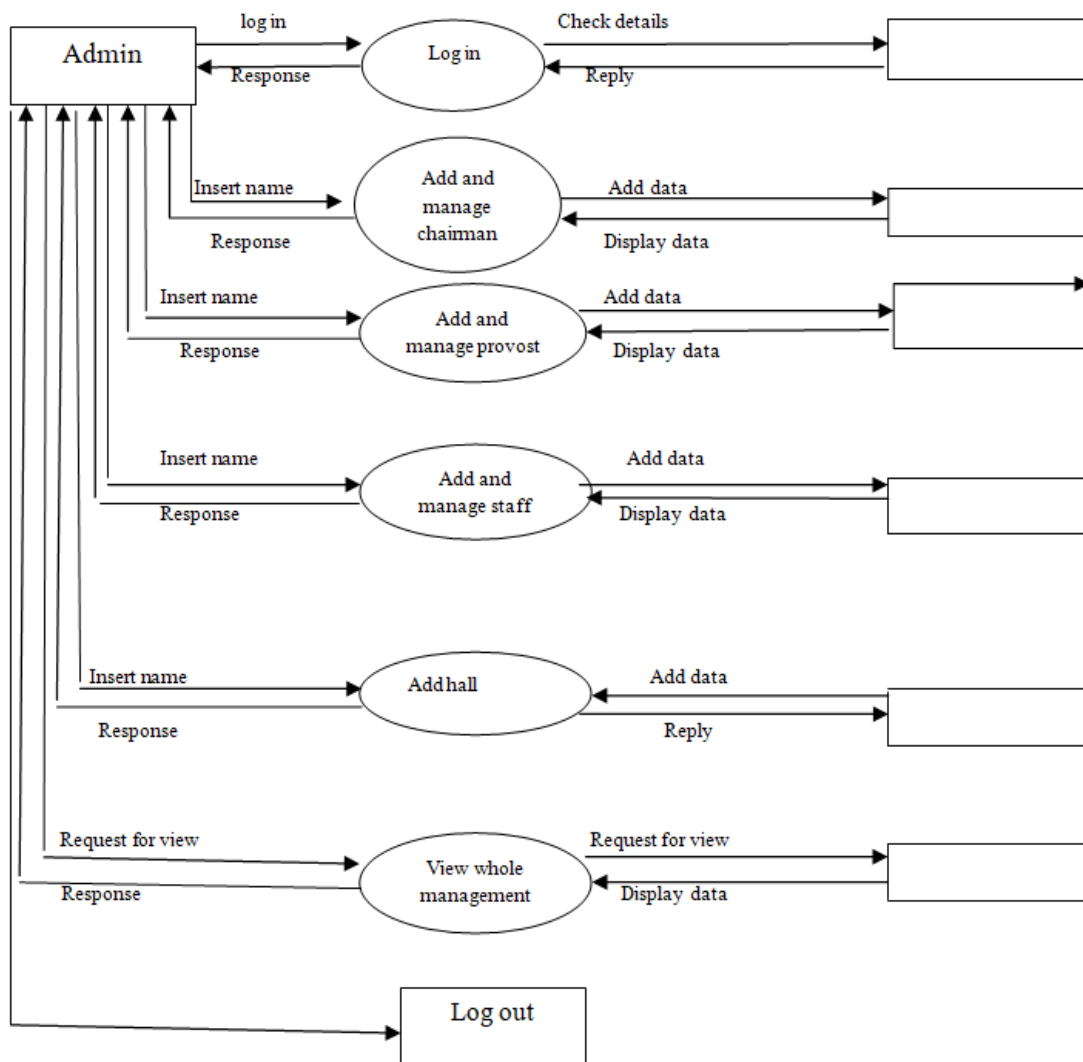


Figure 4.15:Data Flow Diagram for admin

Level 1 DFD User-data flow diagram

In this level 1 DFD there are several data stores and data-flows between processes and the data stores. The User side DFD describes the functionality of the user. A user is a person who uses the project services. Here user is student. They can view accommodation information, notice board of hall, apply for hall, view hall seat confirmation mail from provost and also view their allotment. The user level 1 data flow diagram is shown in 4.16

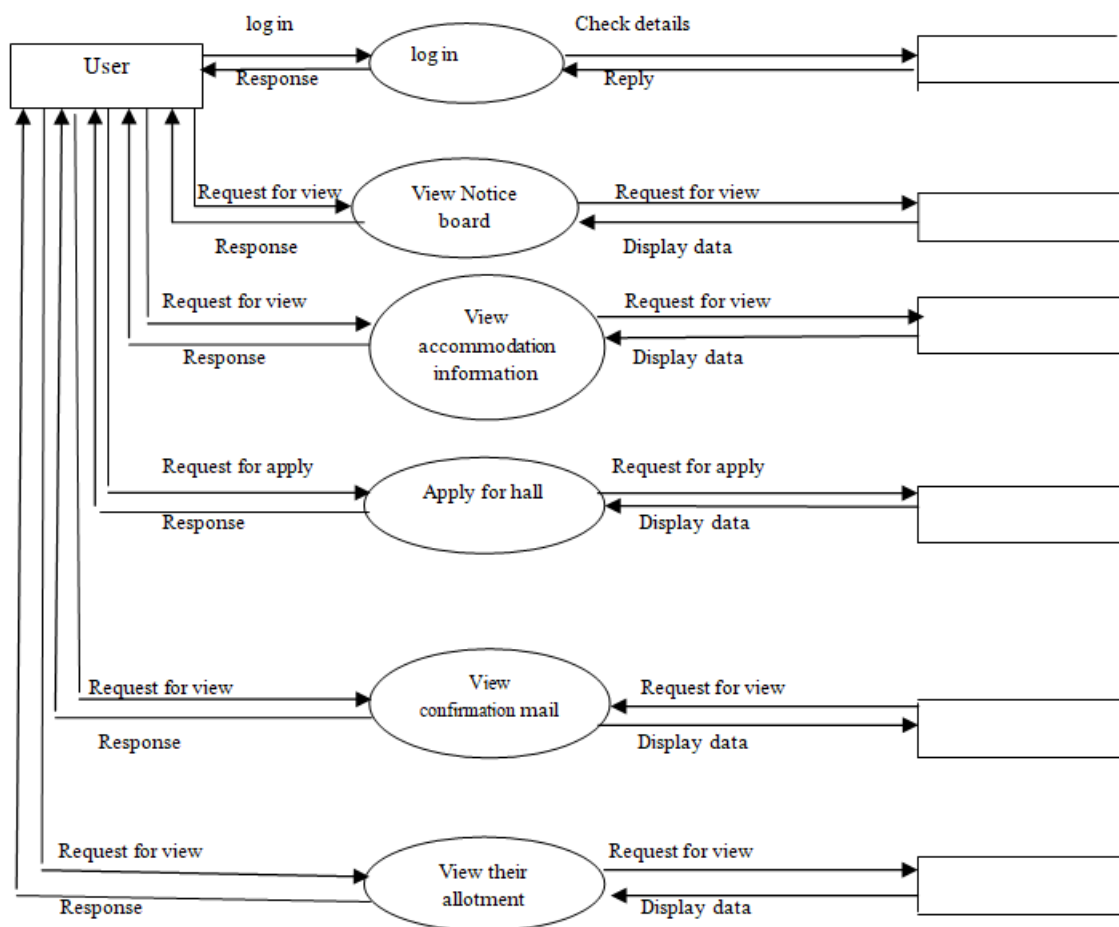


Figure 4.16: Data Flow Diagram for user

Level 1 DFD Chairman-data flow diagram

In this level 1 DFD there are several data stores and data-flows between processes and the data stores. The User side DFD describes the functionality of the chairman. He can view the application for hall, total students then confirm the application for hall. Then log out from the system. The chairman level 1 data flow diagram is shown in 4.17

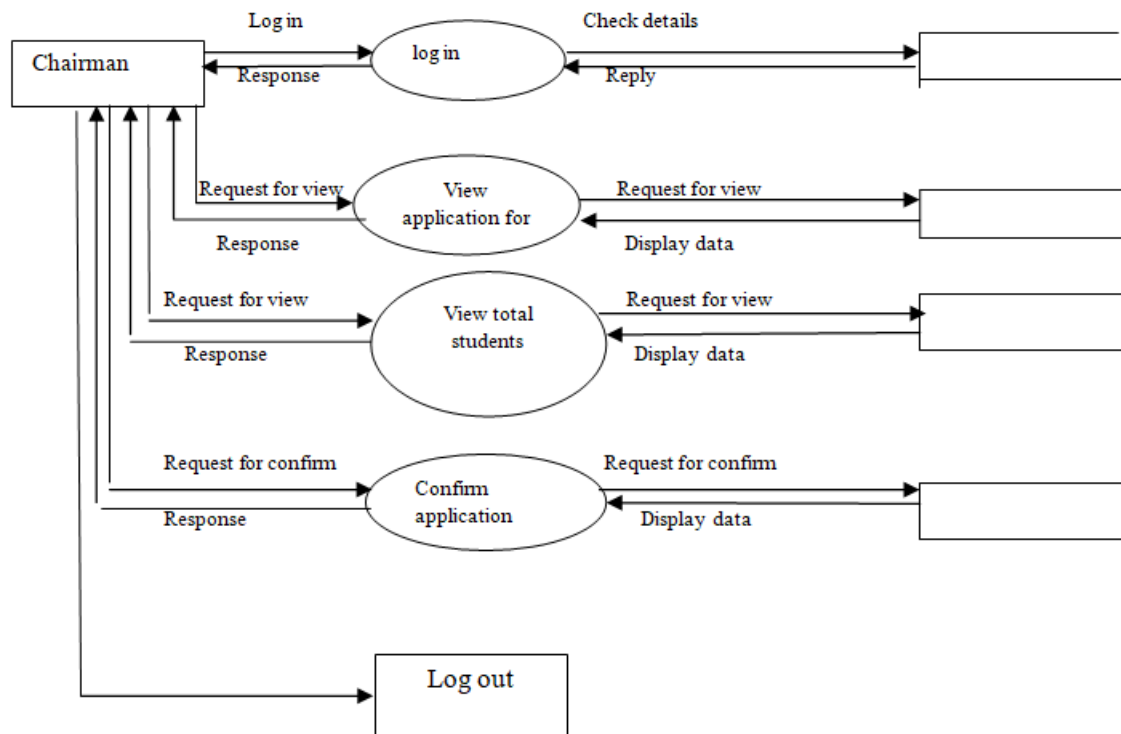


Figure 4.17: Data Flow Diagram for chairman

Level 1 DFD Provost-data flow diagram

In this level 1 DFD there are several data stores and data-flows between processes and the data stores. The User side DFD describes the functionality of the Provost. He can view the application for hall, total students then confirm the application for hall. After confirmation he can view the room capacity and add students to room. Then log out from the system. The Provost level 1 data flow diagram is shown in 4.18

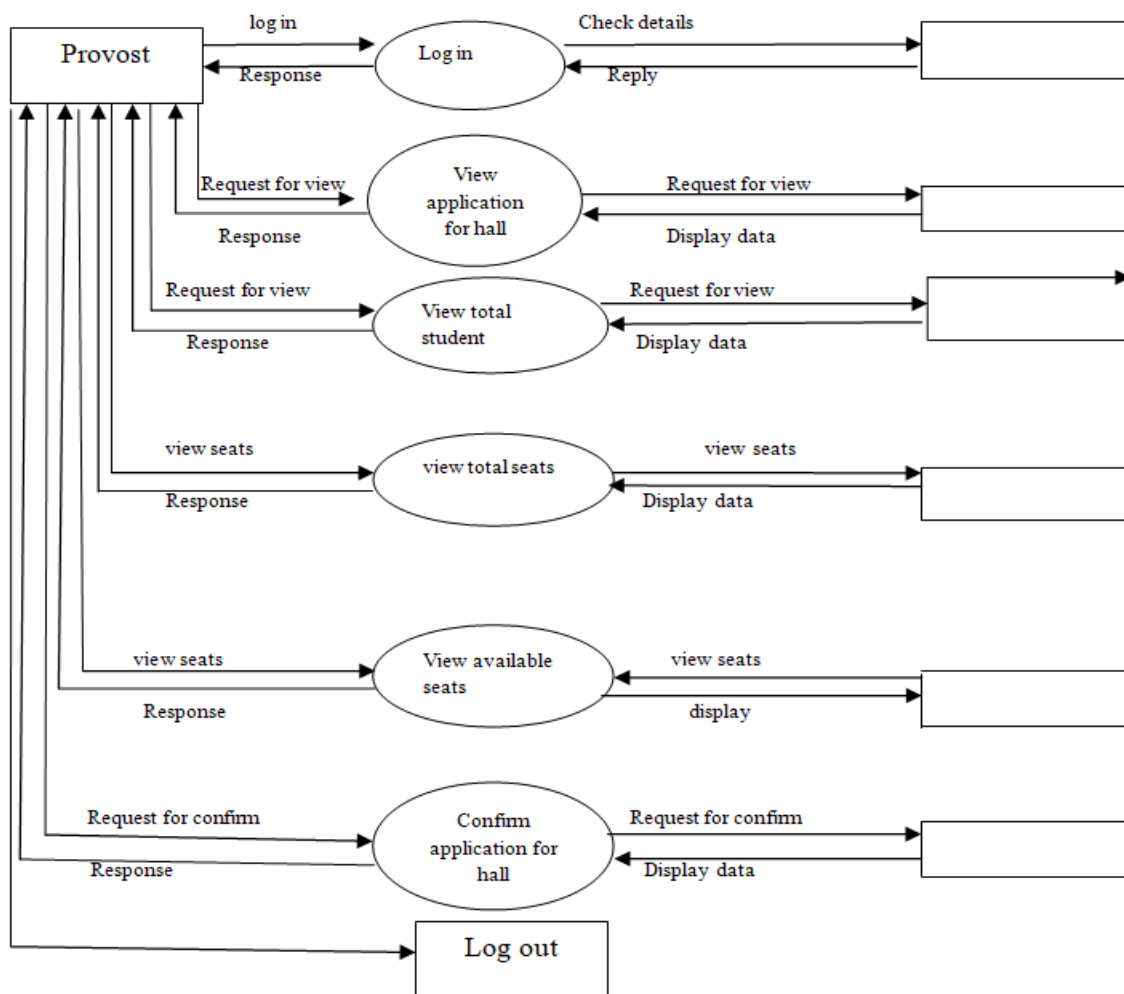


Figure 4.18: Data Flow Diagram for provost

4.6 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database [16]. This logical Data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the data base structures used to store the data. In the relational model these are the tables and views the process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- Determine the relationships between the different data elements.
- Superimpose a logical structure upon the data on the basis of these relationships.

4.6.1 DESIGN PROCESS

1. Determine the purpose of the database - This helps prepare for the remaining steps
2. Find and organize the information required - Gather all of the types of information to record in the database, such as student name, ID
3. Divide the information into tables - Divide information items into major entities or subjects, such as Items or Orders. Each subject then becomes a table.
4. Turn information items into columns - Decide what information needs to be stored in each table. Each item becomes a field and is displayed as a column in the table. For example, a Registration table might include fields such as name and email.
5. Specify primary keys - Choose each table's primary key. The primary key is a column, or a set of columns, that is used to uniquely identify each row. An example might be Item ID or Order ID.
6. Set up the table relationships - Look at each table and decide how the data in one table is related to the data in other tables which is establish Relational Database Management System (RDBMS). Add fields to tables or create new tables to clarify the relationships, as necessary.

- Refine the design - Analyze the design for errors. Create tables and add a few records of sample data. Check if results come from the tables as expected. Make adjustments to the design, as needed.

The Database table is look like this. The database table for user, student are shown in below

SELECT * FROM `users`

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

		id	status	name	email	phone	rolle	dept_id	hall_id	std_id	email_verified_at	password
☐	Edit Copy Delete	2	1	nstu	nstu@gmail.com	NULL	0	0	0	0	NULL	\$2y\$10\$Gb.1Kx
☐	Edit Copy Delete	28	0	tahan	aa8951706@gmail.com	87654343209	0	0	0	25	NULL	\$2y\$10\$V5VAU
☐	Edit Copy Delete	5	1	abdul	abdul@gmail.com	NULL	0	0	0	0	NULL	\$2y\$10\$T.JVR6
☐	Edit Copy Delete	6	1	kamal	kamal@gmail.com	NULL	0	0	0	0	NULL	\$2y\$10\$zs8wXf
☐	Edit Copy Delete	25	0	jannat	kamrulhasan94465@gmail.com	1773138682	0	0	0	21	NULL	\$2y\$10\$8YCIYB
☐	Edit Copy Delete	26	0	anas	anas@gmail.com	12345676754	0	0	0	22	NULL	\$2y\$10\$8YEAP
☐	Edit Copy Delete	11	1	sonjoy	kamrulhasan94465@gmail.com	1974578869	2	0	2	0	NULL	\$2y\$10\$NkDmc
☐	Edit Copy Delete	12	1	sumit	sumit@gmail.com	1874563964	2	0	5	0	NULL	\$2y\$10\$xYwEF
☐	Edit Copy Delete	13	1	Md. Ashikur Rahman khan	jannat.nstubd@gmail.com	1836350710	1	1	0	0	NULL	\$2y\$10\$eujcjLv
☐	Edit Copy Delete	14	1	stuff	stuff@gmail.com	1773138682	3	0	2	0	NULL	\$2y\$10\$yOydK

Table 1.1 User login

SELECT * FROM `students`

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

		id	name	email	roll	phone	session	dept_id	hall_id	status	gender	application	image
☐	Edit Copy Delete	1	Rahim	rahim@gmail.com	ASH1611006M	12345678900	201-15	2	1	2	0	abc	public/UploadImage/600b
☐	Edit Copy Delete	2	mukta	mukta@gmail.com	muh3467895f	1876112677	2014-15	2	1	2	1	need a seat	public/UploadImage/600c
☐	Edit Copy Delete	3	sadia	sadia@gmail.com	muh345671295	1976666665	2758-45	1	1	0	1	need a seat	public/UploadImage/600c
☐	Edit Copy Delete	4	afsana	afsana@gmail.com	bkh18123453	1728987643	2349-45	1	1	0	1	dhdjifnc	public/UploadImage/600c
☐	Edit Copy Delete	5	demo	seat@gmail.com	ASH1622006M	1234567890	2014-14	2	2	0	0	seat	public/UploadImage/600c
☐	Edit Copy Delete	6	bges	bge@gmail.com	ASH110045M	1234567878	2014-15	3	2	2	0	seat	public/UploadImage/600c
☐	Edit Copy Delete	7	oishy	oishy@gmail.com	bkh345678	2867854329	498-34	3	1	2	1	jjjkg	public/UploadImage/600c
☐	Edit Copy Delete	8	mithi	mithi@gmail.com	bkh3456	1984765236	2345-67	3	1	2	1	i need a seat	public/UploadImage/600c
☐	Edit Copy Delete	9	tahsin	tahsin@gmail.com	bkh18115	1874666666	2017-18	3	1	2	0	i need a seat	public/UploadImage/600c

Table 1.2 Student table

4.7 SUMMARY

From the above study now we know the Design process, Hardware, Software, and the working of the system. Design process plays an important role in project development. Students application first sent to department chairman as an anonymous message, after confirmation it will be sent to the provost, the confirmation mail will be sent to the students after the provost confirm the application. Only the registered chairman and hall provost can view the details of students and confirm their application via mail. Students can also receive the confirmation mail. Hall provost can add student in the hall if hall seats are available otherwise student cannot be allowed to hall.

.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 INTRODUCTION

In this chapter, the results we have obtained from our system is presented and discussed. Project implementation or project execution is the phase where visions and plans become reality. Actually, project implementation consists of carrying out the activities with the aim of delivering the outputs and monitoring progress compared to work plan. Monitoring can be defined as control of the project implementation in order to keep project on track and achieve the end results of the project. Most of the visions and plans of the project what was taken during the project proposal is implemented. .

The result and discussion after implementing the feature of the project is discussed based on two phase. They are:

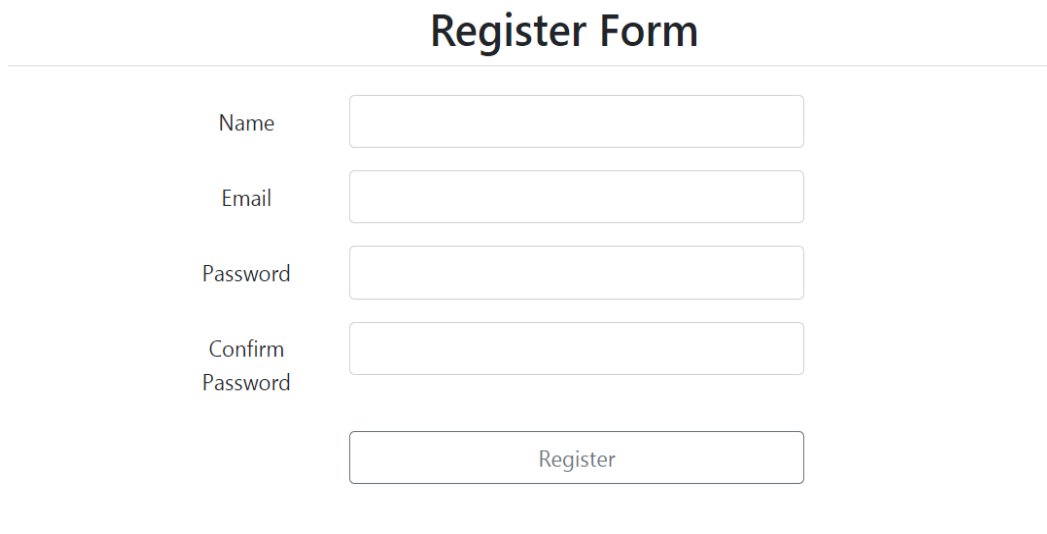
- i. Front end Design and User Interface model
- ii. Back end Design

5.2 USER INTERFACE

The design of the user interface is not just about buttons and taskbar, it is about cooperation between the user and the application or device, it is about cooperation several users. This signifies that the design of the user interface is not about how a system looks but rather how it operates. The screen that appears when the application opens is displayed. It is the one that is login screen in which the user has to log in to the site with his ID and password. Only the registered admin can access the system.

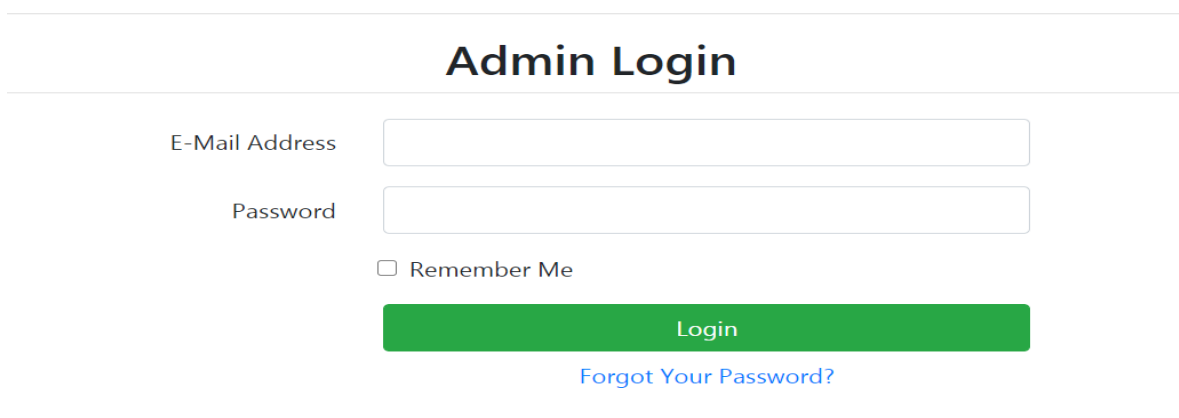
5.3 REGISTRATION AND LOGIN IMPLEMENTATION

The registration and login model is used to input the user name and password. If usernames and passwords don't match, the user can't able to access the site. Mainly a login form is used to enter authentication credentials to access a restricted page or form. Registration form is filled by admin. Only the registered admin can enter and view the whole management system. Here is the screenshot of sign up registration and login model which is shown in figure below.



The 'Register Form' is a web form with a title 'Register Form' at the top. It contains four input fields: 'Name', 'Email', 'Password', and 'Confirm Password'. Each field is represented by a text label followed by a rectangular input box. Below these fields is a 'Register' button, which is a rectangular box with the text 'Register' centered inside it.

Figure 5.1: Admin registration



The 'Admin Login' form has a title 'Admin Login' at the top. It includes two input fields: 'E-Mail Address' and 'Password', each with a text label and a rectangular input box. Below the 'Password' field is a checkbox labeled 'Remember Me'. At the bottom of the form is a green 'Login' button with the text 'Login' in white. Below the button is a blue link that says 'Forgot Your Password?'.

Figure 5.2: Admin login

5.4 HOME PAGE

This is the first screen, that user see when he open the site. The home page of the developed application is presented in Figure 5.4. It was created by PHP and design was done by adobe Photoshop. We choose these back ground for its soft color. And that's the small view of the project. By clicking one of these options user can go next of this application which is shown below.

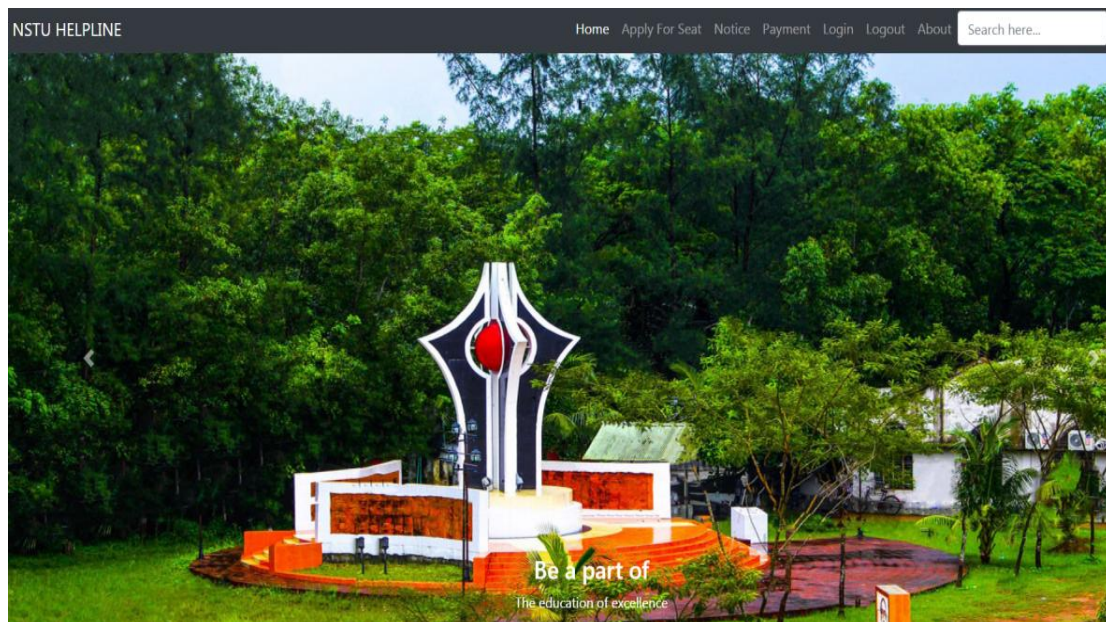


Figure 5.3: Home page

5.5 DASHBOARD

By effective login the Users application's dashboard is shown where user performances are appeared. They perform their task by enter each of the task. The Dashboard is seen in the figure below.

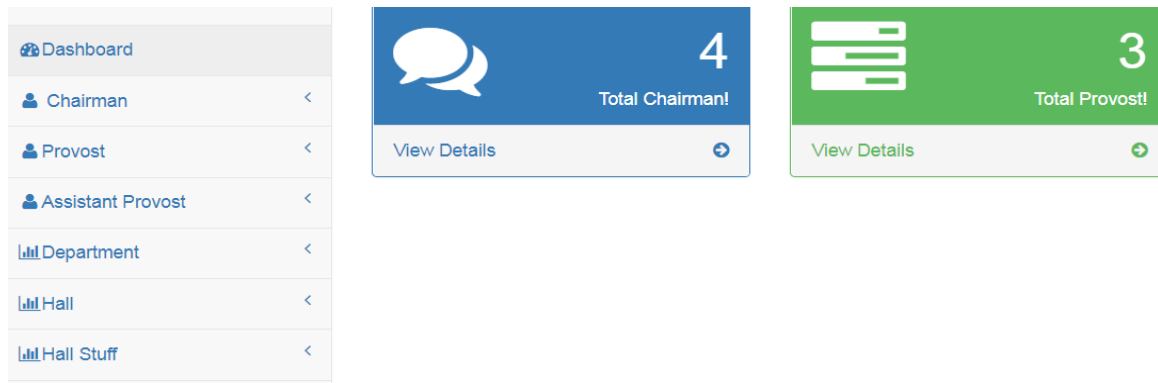


Figure 5.4: Dashboard for admin

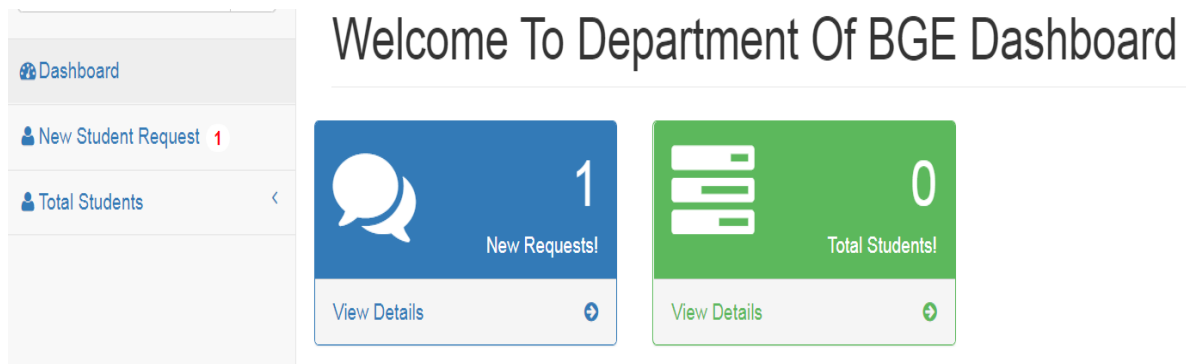


Figure 5.5: Dashboard for chairman

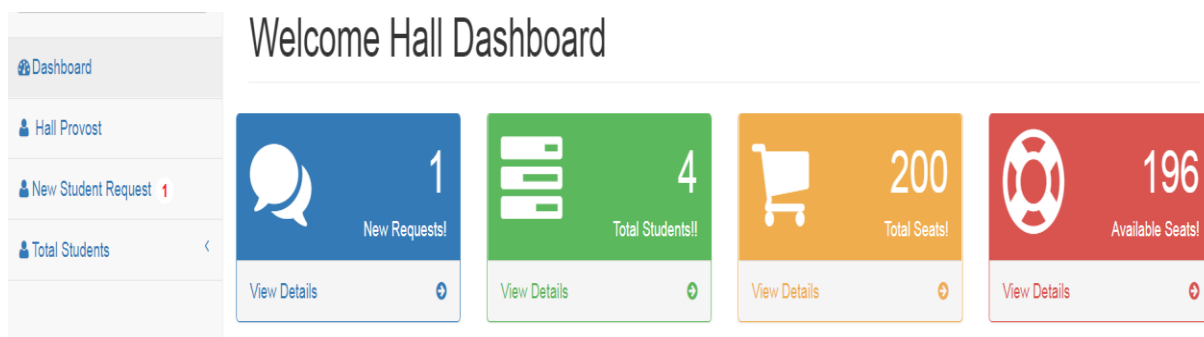


Figure 5.6: Dashboard for hall provost

5.6 EDITED/MANAGED ITEMS

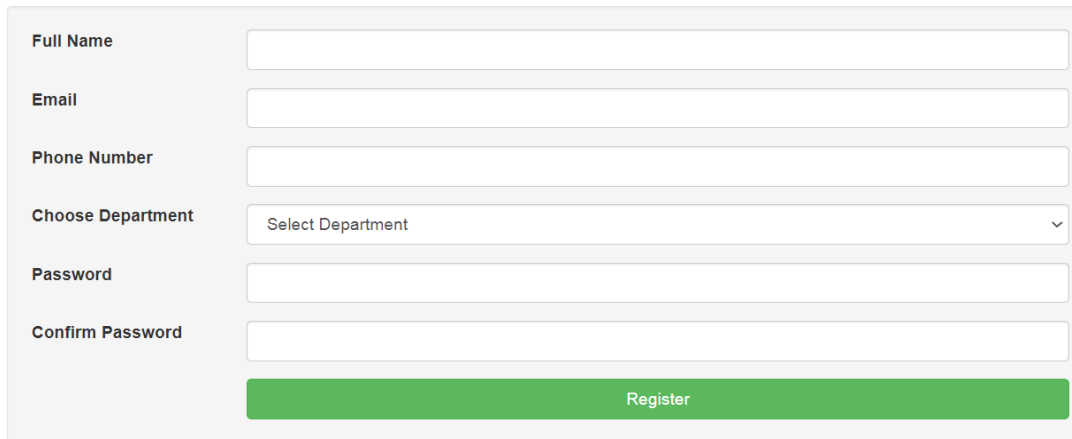
Only the registered admin can be allowed to enter the system. Admin can add department, hall, admin will also add hall staff, chairman, provost with their full name, phone number, email ID and also mention the department name, hall name so that student will be able to know about each hall provost individually. Admin can view total chairman, total provost, manage chairman and provost, application for hall seat, delete hall, delete department to the system.

When user clicks on apply for seat option then this screen will appear. They will then be able to apply for the hall. After submitted the application an email will sent to department chairman. Chairman will check the mail and confirm student request. After confirmation of chairman an email sent to hall provost for his/her confirmation.

Provost can view new student request, total student, total seat, available seats, view details of students and confirm students request for hall seat. Then Provost entered the room number and allot seat to student. If available seats are fulfill there are showing overflow message else there are showing empty message. Then provost can add student to the room. After confirmation of provost an email sent to student. After view confirmation mail from provost then the student will view their seat allotment by clicking the application. In this system payment option is offline payment. Students deposited their fees in the university bank.

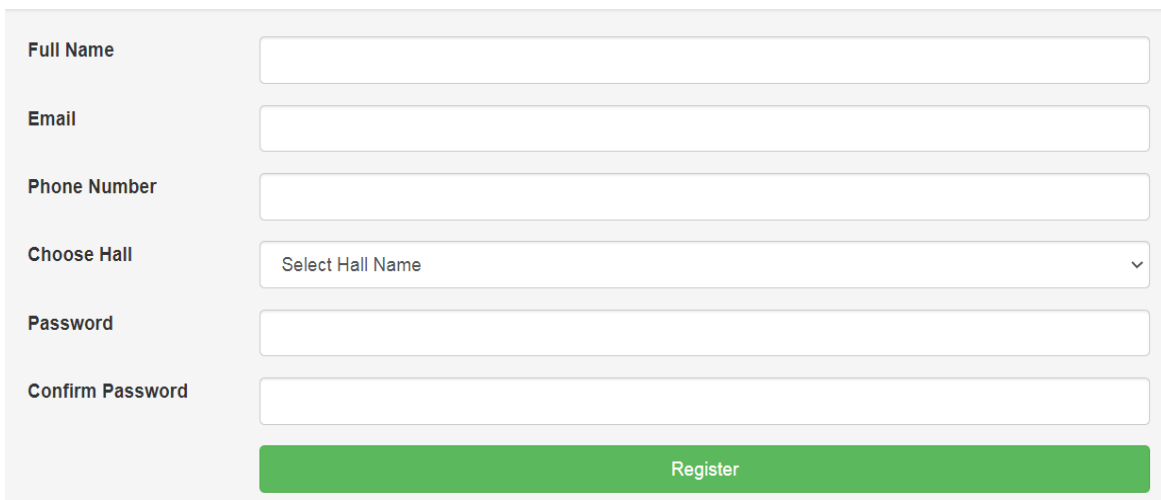
5.7 FINAL OUTLOOK

We can enter the developed application and select option which we have needed. We can select apply for seat, notice board, hall name and view details by clicking on the provided option and see the desired action. Anyone can open the website by just entering the address of website in the address bar of the browser.



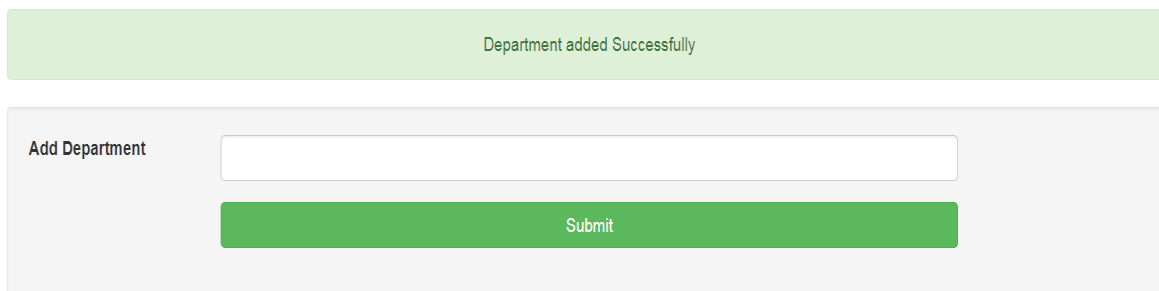
A registration form for a Department Chairman. It features six input fields: Full Name, Email, Phone Number, Choose Department (a dropdown menu with 'Select Department' as the placeholder), Password, and Confirm Password. A green 'Register' button is positioned at the bottom right of the form.

Figure 5.7: Department Chairman Addition form



A registration form for a Hall provost. It features six input fields: Full Name, Email, Phone Number, Choose Hall (a dropdown menu with 'Select Hall Name' as the placeholder), Password, and Confirm Password. A green 'Register' button is positioned at the bottom right of the form.

Figure 5.8: Hall provost addition form



A page for adding a new department. It starts with a green success message: 'Department added Successfully'. Below this is a form with a single input field labeled 'Add Department' and a green 'Submit' button.

Figure 5.9: Department addition page

Add Hall

Available Seat

Hall Image

Choose File

No file chosen

Submit

Figure 5.10: Hall addition page

SN	Name	Phone	Email	Department	Action
1	Md. Ashikul	098109765321	ashik.nstu@yahoo.com	CSTE	
2	md ashik	087663432109	ashik@nstu.edu.bd	ICE	
3	sdasd	087543563214	sad@gmail.com	BGE	
4	Md. Ashikur Rahman khan	01836350710	jannat.nstubd@gmail.com	ICE	

Figure 5.11: Department Chairman Details

SN	Name	Phone	Email	Hall Name	Action
1	Atik	043218890098	atik@gmail.com	Jatir Pita Bangabandhu Sheikh Mujibur Rahman Hall	
2	meherjan	045453266661	meherjan@gmail.com	Abdul Malek Ukil Hall	
3	farabi	012098543215	farabi@gmail.com	Hazrat Bibi Khadiza Hall	
4	jerin	032123456987	jerin@gmail.com	Abdul Malek Ukil Hall	
5	sonjoy	01974578869	kamrulhasan9465@gmail.com	Hazrat Bibi Khadiza Hall	

Figure 5.12: Hall Provost details

SN	Name	Action
1	FTNS	 
2	Microbiology	 
3	PHARMACY	 
4	FIMS	 
5	ESDM	 
6	BBA	 
7	ACCE	 
8	BGE	 
9	CSTE	 
10	ICE	 

Figure 5.13: Departments added by admin

SN	Name	Available Seat	Action
1	Bangamata Sheikh Fojilatunnesa Mujib Hall(FMS)	200	 
2	Jatir Pita Bangabandhu Sheikh Mujibur Rahman Hall	340	 
3	Abdul Malek Ukil Hall	300	 
4	Hazrat Bibi Khadiza Hall	200	 
5	Vasha Shahid Abdus Salam Hall	200	 

Figure 5.14: Halls added by admin

Full Name	
Roll	
Email	
Phone Number	
Session	
Description	
Choose Department	Select Department
Choose Hall	Select Hall
Choose Gender	Select Gender
Image	Choose File No file chosen
Submit	

Figure 5.15: Hall seat application form

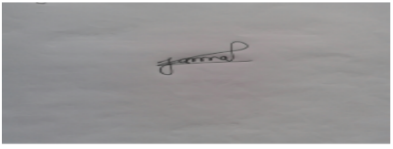
Student Details	
	
SN	1
Name	foez
Roll	298
Email	foez@gmail.com
Phone	1812433655
Session	3421
Name Of the Department	BGE
Name Of the Hall	Jatir Pita Bangabandhu Sheikh Mujibur Rahman Hall
Confirm	

Figure 5.16: Student details for department chairman and provost

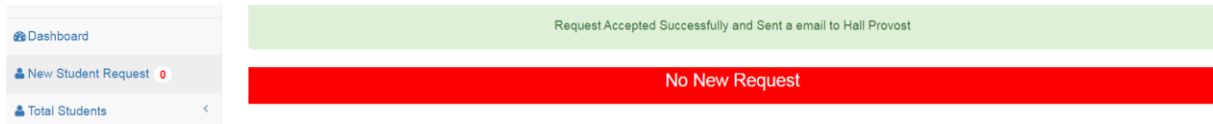


Figure 5.17: After confirm student request by department chairman

Dashboard	SN	Name	Roll	Email	Phone	SEssion	Dept_Name	Hall_name	Action
Hall Provost	1	tahsin	bkh18115	tahsin@gmail.com	1874666666	2017-18	BGE	BKH	Confirm
New Student Request 1									
Total Students									

Figure 5.18: View student request for hall seat

Dashboard	Request Accepted Successfully and Sent a email to Student
Hall Provost	No New Request
New Student Request 0	
Total Students	

Figure 5.19: After confirm student request by hall provost

SN	Room No.	Total Seat	Available Seats	Total Student	Action
1	1	3	0	3	Overflow
2	2	3	0	3	Overflow
3	3	3	0	3	Overflow
4	6	3	0	3	Overflow
5	6	3	3	0	Empty
6	301	24	24	0	Empty
7	302	5	5	0	Empty
8	303	5	5	0	Empty
9	102	4	4	0	Empty
10	203	5	5	0	Empty

< 1 2 >

Figure 5.20: Room and Seat detail

Request Accepted Successfully and Sent a email to Student					
SN	Room No.	Total Seat	Available Seats	Total Student	Action
1	1	3	0	3	Overflow
2	2	3	0	3	Overflow
3	4	5	5	0	Add Student

Figure 5.21: Add student to room

Application Progress Level

Your Application is Successfully accepted. Please check your email for details or [Click here..](#)

Figure 5.22: Seat confirmation message for student

Seat Details

Hall name: Jatir Pita Bangabandhu Sheikh Mujibur Rahman Hall
Room No. 201

Figure 5.23: Seat allotment for student

Payment Option

Please Complete your payment within 6 Months

Bkash

Roket

Offline Bank Payment

Figure 5.24: Payment option

5.8 RESULTS

Our desired result from the above discussion and figures states that this web application can overcome the problem of manual work. It will help the students in getting the seat without fall any embarrassing situation. The University Hall management system is user-oriented application specially designed to make automate, unique and maintain all the process of managing hall. It can exclude more man power compare to the existing system. All the function that hall management encompasses can easily be managed by hall management application software.

5.9 SCOPE AND LIMITATIONS

- i. We can add new features as and when we require.
- ii. Many people do not use internet.
- iii. Reusability of this application is also possible.
- iv. The amount of time and difficulty of using virtual instruction may be significant
- v. University can make use of it for saving students details in database.
- vi. University management group can use it for managing their information, accommodation details.
- vii. This application can easily implement under various situations.
- viii. Lack of proper infrastructure.
- ix. Incidents of local people harassing and torturing the students especially the foreign students in different parts of the country.

5.10 SUMMARY

To evaluate the performance of the proposed methods, the final result is performed in this chapter. The project developed using PHP, MySQL, HTML, CSS JavaScript is based on the requirement specification of the user and the analysis of the existing system. We design this University Hall Management System which is useful especially for the large educational institutions student with university hall student. Hope this site is beneficial for them. We completed the system that is performed easily. Results show the full overview of the project and considerable improvement for hall management system.

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

This chapter describes the conclusion and future scope of this project. First the overall summary of this project work is presented in the conclusion section. Then the potential future research directions are provided in future scope section.

6.2 CONCLUSION

The ‘UNIVERSITY HALL MANAGEMENT SYSTEM’ is designed to provide a web based application that would make easier for the student. The project which we proposed, when we reach the destination, will bring a lot of benefit to huge number of students of NSTU. ‘UNIVERSITY HALL MANAGEMENT SYSTEM’ is a user-friendly computer-based system for managing hall facilities in institutions. It has been designed to automate, manage and look after the overall processing of records of students residing in a large hall. It is capable of managing Enquiry details, Student Details, Payment Details etc. The developed system can provide a solution to manual hall management problems and also provides information such as hall information, hall room information, and also provide mailing facility for the user. The software offers stability, cost-effectiveness and usability. It provides the most flexible and adaptable standards management system solutions for hall. My proposed website will help the student to find out their required information very easily with taking a few minutes. They can also monitor which residence is appropriate for them and they can apply for the hall to the provost and they will also receive a seat confirmation mail.

6.3 FUTURE SCOPE OF THE PROJECT

The flexibility of online system became so appealing that it was no longer about physical locations at all but simply maximizing the flexibility in time and space. There are several possible ways to expand the implementation. A future application of this system lies in the fact that the proposed system would remain relevant in the future. . In case there be any additions or deletion of the services in any type of modification in future can be implemented easily. The data collected by the system will be useful for some other purposes also. We may include hotel and home rent booking system and will work for post-admission facilities in future. Also we include online payment system via Bkash/Rocket in future development.

REFERENCES

1. Automated Online College Admission Management System. [Online] Available: <http://www.ijcstjournal.org/volume-5/issue-3/IJCST-V5I3P1.pdf> [Accessed: 08 January, 2021].
2. International journal of scientific and research publication. Rental House Management System. Available: <http://www.ijserp.org/researchpaper-1114/ijserp-p35101.pdf>. [Accessed: 08 March, 2021].
3. Mr Dineshkumar Mulchandani developed Online Hostel Management System, in partial fulfillment for the award of the degree of Bachelor of technology in Information and technology (June, 2015). [Online]: <https://www.scribd.com/doc/280126229/online-hostel-management-system>. [Accessed: 23 February, 2021]
4. Development of an automated Hostel Facility Management System (September, 2014) [Online]. Available: https://www.researchgate.net/publication/312490400_Development_of_an_Automated_Hostel_Facility_Management_System [Online]. [Accessed: 12 March, 2021].
5. Ritesh Kumar Bista | Aman Jung Karki | Beesu Venkat Mouneesh Reddy | Utkarsh Aakash | Dr. Rajasimha A Makaram | Shilpa Das "Hostel Management System" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-2 | Issue-4, June 2018, <http://www.ijtsrd.com/papers/ijtsrd14110.pdf>.
6. U. Elakkiya, P. Nirmala Priyadarshini Web-Based Hostel Management System For Improving Sustainable Performance Of Educational Institutions. International Journal of Latest Trends in Engineering and Technology

APPENDIX

HOME

This function is used for overall home page. Here two parts, header and footer. Header and footer is designed by html and css. And the main part is done by PHP.

ADMIN

```
<? php
```

```
name space App\Http\Controllers\Admin;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
use Illuminate\Support\Facades\Auth;
```

```
use App\Project;
```

```
use App\Student;
```

```
use App\Hall;
```

```
use App\User;
```

```
use App\Admission;
```

```
use DB;
```

```
class Project Controller extends Controller
```

```
{
```

```
    public function index(){
```

```
        return view('Admin. project. create');
```

```
    }
```

```
    public function store(Request $request){
```

```
        $this->validate($request,[
```

```

        'dept_name'=>'required'
    });

    Project::create([
        'dept_name'=>request('dept_name'),
    ]);
    return redirect()->back()->with('message','Department added Successfully');
}

public function manageProject(){
    $projects = Project::OrderBy('id','desc')->paginate(10);
    return view('Admin.project.index', compact('projects'));
}

public function deleteProject($id){
    $deleteCat = Project::find($id);
    $deleteCat->delete();
    return redirect()->back()->with('message','Department deleted Successfully');
}

public function editProject($id){
    $project = Project::find($id);
    return view('Admin.project.editContain', compact('project'));
}

public function updateProject(Request $request){
    $editCat = Project::where('id',$request->id)->first();
    $editCat->dept_name = $request->dept_name;
    $editCat->save();
    return redirect()->route('project.manage')->with('message','Department updated
Successfully');
}

```

```
}
```

```
public function workProject(Request $request){  
    $this->validate($request,[  
        'user_id'=>'required',  
        'project_id'=>'required',  
    ]);  
  
    Freelancer_project::create([  
        'user_id'=> $request['user_id'],  
        'project_id'=> $request['project_id'],  
    ]);  
    return redirect()->back()->with('message','Project Assign Successfully');  
}
```

```
public function addHall(){  
    return view('Admin.Hall.create');  
}
```

```
public function storeHall(Request $request){  
    $this->validate($request,[  
        'hall_name'=>'required',  
        'avail_seat'=>'required'  
    ]);  
  
    $hall = new Hall();  
    if($request->image==""){  
        $image=0;  
        $hall->hall_name= $request->hall_name;  
        $hall->image = $image;
```

```

        $hall->avail_seat= $request->avail_seat;
        $hall->save();
        return redirect()->back()->with('message','Hall added Successfully');
    }else{
        $image=$request->file('image');
        $name=uniqid().$image->getClientOriginalName();
        $uploadPath='public/UploadImage/';
        $image->move($uploadPath,$name);
        $imageUrl=$uploadPath.$name;

        $hall->hall_name= $request->hall_name;
        $hall->image = $imageUrl;
        $hall->avail_seat= $request->avail_seat;
        $hall->save();
        return redirect()->back()->with('message','Hall added Successfully');
    }

}

}

public function manageHall(){
    $projects = Hall::OrderBy('id','desc')->paginate(10);
    return view('Admin.Hall.index', compact('projects'));
}

public function deleteHall($id){
    $deleteCat = Hall::find($id);
    $deleteCat->delete();
    return redirect()->back()->with('message','Hall deleted Successfully');
}

```

```

public function editHall($id){
    $project = Hall::find($id);
    return view('Admin.Hall.editContain', compact('project'));
}

```

```

public function updateHall(Request $request){
    $editCat = Hall::where('id',$request->id)->first();
    $editCat->hall_name = $request->hall_name;
    $editCat->avail_seat = $request->avail_seat;
    $editCat->save();
    return redirect()->route('hall.manage')->with('message','Hall updated Successfully');
}

```

```

public function studentRequest(){
    $id = Auth::user()->dept_id;
    $students =DB::table('students')
    ->join('halls','students.hall_id','=','halls.id')
    ->join('projects','students.dept_id','=','projects.id')
    ->where('students.status',0)
    ->where('students.dept_id',$id)
    ->orderBy('students.id','desc')
    -

```

```

>select('students.id','students.name','students.email','students.phone','students.session','students.roll','halls.hall_name','projects.dept_name')
    ->get();
    return view('Admin.Chairman.index', compact('students'));
}

```

```

public function confirmRequest($id){
    $studentStatus = Student::find($id);
    if($studentStatus->status==0){

```

```

        $studentStatus->status=1;
        $studentStatus->save();
    }
    return redirect()->back()->with('message','Request Accepted Successfully and Sent a email
to Hall Provost');

}

public function viewDetail($id){
    $project = DB::table('students')
->join('halls','students.hall_id','=', 'halls.id')
->join('projects','students.dept_id','=', 'projects.id')
->where('students.id',$id)
-
>select('students.id','students.name','students.email','students.phone','students.session','students.ro
ll','students.image','halls.hall_name','projects.dept_name')
->first();
    return view('Admin.Chairman.detail', compact('project'));
}

public function totalStudent(){
    $id = Auth::user()->id;
    $user = User::where('id',$id)->first();
    $projects = DB::table('students')
->join('halls','students.hall_id','=', 'halls.id')
->join('projects','students.dept_id','=', 'projects.id')
->where('students.dept_id',$user->dept_id)
->where('students.status',2)
-
>select('students.id','students.name','students.email','students.phone','students.session','students.ro
ll','students.image','halls.hall_name','projects.dept_name')

```

```

->paginate(10);

return view('Admin.Chairman.totalStudent', compact('projects'));
}

public function seatRequest(){
    $id = Auth::user()->hall_id;
    $students =DB::table('students')
    ->join('halls','students.hall_id','=', 'halls.id')
    ->join('projects','students.dept_id','=', 'projects.id')
    ->where('students.status',1)
    ->where('students.hall_id',$id)
    ->orderBy('students.id','desc')
    -
    >select('students.id','students.name','students.email','students.phone','students.session','students.ro
    ll','halls.hall_name','projects.dept_name')
    ->get();
    return view('Admin.Provost.index', compact('students'));
}

public function totalHallStudent(){
    $id = Auth::user()->hall_id;
    $students =DB::table('students')
    ->join('halls','students.hall_id','=', 'halls.id')
    ->join('projects','students.dept_id','=', 'projects.id')
    ->where('students.status',2)
    ->where('students.hall_id',$id)
    ->orderBy('students.id','desc')
    -
    >select('students.id','students.name','students.email','students.phone','students.session','students.ro
    ll','halls.hall_name','projects.dept_name')

```

```

->paginate();
return view('Admin.Provost.totalStudent', compact('students'));
}

public function seatConfirm($id){
    $studentStatus = Student::find($id);
    if($studentStatus->status==1){
        $studentStatus->status=2;
        $studentStatus->save();
    }
    return redirect()->back()->with('message','Request Accepted Successfully and Sent a email
to Student');
}

public function allAplicant(){
    $allAplicant = Admission::paginate(10);
    return view('Admin.Admission.index',compact('allAplicant'));

}
}

```

APPLY FOR HALL SEAT

<?php

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Mail;
use App\Project;
use App\Hall;
use App\User;

```

```
use App\Student;
use App\Admission;
use App\Mail\SendMail;
use DB;
```

```
class HomeController extends Controller
```

```
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index()
    {
        return view('FrontEnd.home.homeContain');
    }

    public function indexs(){
        return view('user.profileContain');
    }

    public function apply(){
        $users = Hall::all();
        $depUser= Project::all();
        return view('FrontEnd.home.applyContain',compact('users','depUser'));
    }
}
```

```
}
```

```
public function StoreApply(Request $request){
```

```
    $this->validate($request,[  
        'name'=>'required',  
        'email'=>'required',  
        'roll'=>'required',  
        'phone'=>'required:min:11',  
        'dept_id'=>'required',  
        'hall_id'=>'required',  
        'session'=>'required',  
        'Application'=>'required',  
        'gender'=>'required',  
        'image'=>'required',  
    ]);
```

```
    // $id=$request->dept_id;  
    // $chairman = User::where('dept_id',$id)->first();  
    // if($chairman->email!=""){  
    //     $chairmanEmail = $chairman->email;}  
  
    $image=$request->file('image');
```

```
        $name=uniqid().$image->getClientOriginalName();  
        $uploadPath='public/UploadImage/';  
        $image->move($uploadPath,$name);  
        $imageUrl=$uploadPath.$name;
```

```
        $user = new Student();  
        $user->name = $request->name;  
        $user->roll= $request->roll;
```

```

$user->email = $request->email;
$user->phone = $request->phone;
$user->session= $request->session;
$user->application = $request->application;
$user->dept_id = $request->dept_id;
$user->hall_id = $request->hall_id;
$user->gender = $request->gender;
$user->image = $imageUrl;

$user->save();
$data = array(

    // 'Email'=>$chairman->email,
    //     'name'=>$request->name,

    );
    // Mail::to($chairman->email)->send(new SendMail($data));

    return redirect()->back()->with('message','Application Submitted Successfully! A Email
Sent to your department chairman. Please Wait For Confirmation');

}

```