PROJECT REPORT

ONLINE BASED UNIVERSITY ADMISSION SYSTEM FOR UNDERGRADUATES

ARIF

ROLL: ASH1311007M

SESSION: 2012-2013

BACHELOR OF SCIENCE (INFORMATION AND COMMUNICATION ENGINEERING)

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

JULY 2018

ONLINE BASED UNIVERSITY ADMISSION SYSTEM FOR UNDERGRADUATES

SUPERVISOR'S DECLARATION

I hereby declared that I've checked this thesis/project in my opinion, this thesis/project is adequate in terms of scope and quality for award of the degree of Bachelor of Science in Information and Communication Engineering.

(Professor Dr. Md. Ashikur Rahman Khan)

Chairman,

Dept. of Information and Communication Engineering

Noakhali Science & Technology University

Sonapur – 3814, Noakhali, Bangladesh

STUDENT'S DECLARATION

This project is submitted to the Dept. of Information and Communication Engineering of Noakhali Science & Technology University in partial fulfillment of the requirements for having the B.Sc. (Engineering) in Information and Communication Engineering (ICE). So, I hereby declare that this project report has not been submitted elsewhere for the requirement of any kind of degree, diploma or publication.

(Arif)

Dept. of ICE, NSTU

Class Roll No. : ASH1311007M

Session : 2012-13

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

I am very grateful to my project guide Professor Dr. Mohammad Ashikur Rahman Khan for the guidance, inspiration and constructive suggestions that were very helpful for me in the preparation of this project. My project will be incomplete without thanking Saifur Rahman sir, Amzad Hossain sir and Aslam Uddin sir, whose continuous encouragement provided me necessary information to choose the field for project

ABSTRACT

As the number of students appearing for the admission are increasing every year, it requires much effort and time to handle the admission system with man power and paper system. So we are in need of a better system to make the process easier and serves better which could be done by Computerized Student Admission System that facilitates the work of the universities and at the same time it must reduce the work load of the organization with expected quality. Quality in the sense, the system try to avoid the mistakes that are usually happen during the Admission Process. As still in our government universities all the work at the time of admission of the students is done manually by ink and paper, which is very slow and consuming much efforts and time. It is required to Design of a online Admission System, to speed up and make it easy to use system.The Online Admission Application System allows institutes to receive applications through the web and candidates need not collect physical forms. Candidates are required to visit the admission website of the institute. All eligibility criteria, no. of seats, selection criteria etc are mentioned in the website. Candidates are required to fill in the online subject choice form and submit it online. The University Admission System has been developed in order to automate the admission system starting after the admission test to until admission process is completed. The system enables online admissions saving the time of the geographically scattered students. It enables reducing time in activities, centralized data handling and paperless admission with reduced manpower. It improves the operational efficiency and reduces the cost. This is a online web application allows us to access the whole information about the admission process. This application also provides information about the Campus & department review so that the applicants can choice their subject choice form without any hesitation. Online admission system has two basic modules. They are Applicant module & Administrator module. Using applicant module, the applicants will fill up subject choice form & submit necessary documents including cash receipt etc. Using administrator module, the administrators will manage admission processes. Thus making the admission process easy and quick.

TABLE OF CONTENTS

List of Figures:

Chapter 1

INTRODUCTION

Online University Admission System is aimed at developing an online admission application for a University. This system is an online system that can be accessed throughout the organization and outside as well with proper login provided. Our system has two type of accessing modes, administrator and applicant. Admission process activities are managed by an administrator. It is the job of the administrator to admit and monitor the whole process. When an applicant log in to the system, he would only view details of his credentials. He can't perform any administrative changes .The system has two modules. They are applicant's module and administrator module. The applicant can fill up the Subject choice form within a deadline. After the deadline the application will allocate applicants a department corresponding their merit positions. Finally the applicants will be asked for depositing admission fee for confirming the admission.

1.1 PROBLEM STATEMENT

There is a widespread agreement that the university admission process is very complicated, costly, take-time, and inconvenient to both Co-op Staffs and the institution. During the admission process the university authority have to manage the required logistics support & deploy administrative staffs for accomplishing the process. It's too much time consuming as well as costly.

This is due the fact that during the admission process, the institutions has to pause or delay some activities to spend time for admission process. Some staffs have to prepare for admission process. The teachers have to spend time for subject allocation & documents verification process. During that time the regular academic activities are hampered. The applicants have to come to university for subject choice purpose & admission confirmation. It is an inconvenient for an applicant that he/she has to come for knowing the allocated subject, if he/she didn't get his/her desired subject.

During & after the admission process days the academic staff have to preserve & organize the required documents (Transcript, certificate, etc.) in a proper way. They have to keep synchronization with the bank for confirming the corresponding applicant's admission.

In addition, managing & organizing processes are also thorny issues. Mistakes can occur anytime.

Those are some typical examples for the inconvenience and complication of the current manual admission system. They lead institutions to the decision of building Online University Admission System to improve effectiveness, reduce time, cost, etc. for the admission process.

1.2 OBJECTIVES OF THE PROPOSED SYSTEM

The aim of Online Admission System is to computerize admission process without any flaws. It's basic objectives are to extend their reach to geographically scattered students, reducing time in activities, centralized data handling and paperless admission with reduced manpower. It's other factors are cost cutting, operational efficiency, etc. Main challenges are effectively completing the admission process within the time limit and making the process easier for the students as well as university authority. Hence there is a prerequisite for maintaining a database, to store all the details.

The aim of the proposed system is to address the limitations of the current system. Following are the objectives of the proposed system:

I. One of the important objectives of the admission system is communicate with all the students scattered geographically.

II. Reduce the time taken process the applications of students, admitting a student, verify student documents etc.

III. .Transfer the data smoothly to all the departments involved and handle the data centralized way.

IV. Reduce the manpower needed to perform all the admission and administration task by reducing the paper works needed.

V. Reduce the cost involved in the admission process.

VI. Improve the operational efficiency by improving the quality of the process.

2.1 Scope of the Project

- Online Admission System is to allocate seats for the eligible students which would be accessed only on the registered systems.
- Students are to be checked for their eligibility criteria before getting registered.
- A database is maintained for storing & manipulating student's credentials.
- For each event database should be updated and should be ready to provide information for next applicant in no time.

CHAPTER 2

FEASIBILITY STUDY

2.1 Economic Feasibility

Economic analysis is most frequently used for evaluation of the effectiveness of thesystem. More commonly knows as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions ismade to design and implement the system.This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management the management, because veryoften the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives theactual comparison of costs and benefits is much more meaningful in such cases.It is economically feasible, it will only require a single operator to operate the system, who isresponsible for entering the data into the database via a user interface provided to him, whocan also able to show all the data in html tabular form so to provide information regarding thestudents who are either taken admission or to take admission, since it requires only a single person to operate the whole system thus reduces the cost to operate the system.

In the system, the organization is most satisfied by economic feasibility.Because, if the organization implements this system, it need not require anyadditional hardware resourcesas well as it will be saving lot of time.

2.2 Technical Feasibility

Technical feasibility centers on the existing manual system of the testmanagement processand to what extent it can support the system.According to feasibility analysis procedure thetechnical feasibility of thesystem is analyzed and the technical requirements such as softwarefacilities,procedure, inputs are identified. It is also one of the important phases of thesystemdevelopment activities.It is technically feasible, since the whole system is designed into thelatest technologies like PHP and SQL Server which are the most recent technologies todevelop web based systems and design databases.The system offers greater levels of user friendliness combined with greaterprocessingspeed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

2.3 Operational Feasibility

It is Operational feasible, since the system is providing a attractive user interface totheoperator/end user, so he feel very easy to work onto it. Response to operator/end user isvery fast and very good.Since, as we mentioned above that it requires much less amount of cost, it uses computer work so it is very fast to operate and it is very easy for user to work onit.

CHAPTER 3

METHODOLOGY

3.1 Software Development Model for Online Admission System

The usage of a software development model [1] is essential for a software project of any scale; and the as such using and implementing a proper and appropriate development model is essential to the overall success, effectiveness and efficiency of a project. Many well described development models exist, however each project is a unique venture, and thus appreciating this fact, most models are never used as is, there are always some modifications made to the model as the situation of the particular project would require. Often, no single model is exclusively used for a single, sometimes a hybrid of models is created and used as per the project demands, and other times a new purpose-built model is created and used. The key to using a development model is, a model which enables the project team to meet the requirements and goals of the project in the most optimally effective and efficient manner should be the one employed. Therefore we began our search for an appropriate software development model by listing and analyzing our key requirements and goals for development.

For our project which was the development of an Online Admission System we decided that a development model would be needed that met the following criteria:

I. Limited development time, scope of deadline extensions are almost none.

II. Need to maintain proper documentation throughout the development phase.

III. Need to have on the fly testing and quick feedback as time constraints are high.

IV. Need to compensate for the fact that all requirements may not be met at delivery, thus at any given stage we should have a product which meets previous requirements/goals.

3.2 Functional Requirements

The system is a web application. Functional requirements of online admission system are as follows:

- Design a database structure for maintaining required applicants information.
- Import university admission result CSV file to the database.
- Then system admin will entry the total available seats corresponding units & departments. Then he will set a deadline for fill up subject choice form & published that for that applicants.
- The applicants have to sign up in the system & have to fill up subject choice form with in the deadline. At the time of choosing subjects the applicants will be checked for subject eligibility. If he didn't eligible for any subject then he will show a message that will inform why he is not eligible for.
- During the deadline the applicants can change his choice list.
- After deadline applicants as well as system admin can't change any subject order.
- In a certain date the list of permitted applicants corresponding departments will be published.
- Then the permitted applicants have to confirm his admission by depositing money & submitting necessary documents cash receipt with in a deadline.
- The submitted documents will be verified by university authority.
- After the deadline the confirmed admitted applicant list will be published by the system.
- Then system administrator will perform auto migration process & then recall the same previous process until the total seats are filled up.

3.3 Nonfunctional Requirements

Usability Requirements

Usability requirements identified for this system are listed below:

- A logical interface is essential to make easy use of system, speeding up common tasks.
- The product could be used by two categories of people mainly administrator category and other users.

Performance Requirements

- The database should be able to accommodate a minimum of 10,000 records of students.
- At any instant the system should support use of multiple users at a time.
- Available results of the applicants should be presented to them in max of two seconds, so retrieving of data should be reliable.
- As each applicant will be given a maximum time of 30 min, accessing from the database should be done at relevant speed.

Reliability

Some of the attributes identified for the reliability is listed below:

- All data storage for user variables will be committed to the database at the time of entry.
- Data corruption is prevented by applying the possible backup procedures and techniques.

Security Techniques

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below:

- Check data integrity for critical variables.
- Every user should be licensed to use the system under any of the two categories provided i.e. either system administrator or registered applicant.
- Communication needs to be restricted when the application is validating the user or license (i.e. using https).

3.4 TECHNOLOGIES

Development Tools:

Feature	Description
Development environment	Windows, Linux
Design tools	Adobe Photoshop, Microsoft Paint
Front end	HTML5[5], CSS3, Twitter Bootstrap, JavaScript[6], jQuery etc.
Back end	PHP[2][3], MySQL[3]
IDE	phpstorm, Notepad++
Compatible browsers	Mozilla 43+, Chrome: 38+, IE 50+

Server Requirements:

Feature	Description
Operating System	Linux
Web Server	Apache/NGINX
PHPVersion	Minimum 5.4
MySQL version	Minimum 5.0
RAM	Minimum 4 GB
Storage	Minimum 10 GB
Bandwidth	Minimum 100 GB/month

3.5 DATA FLOW DIAGRAMS

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing(structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model.

Level 0:

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. In this project at level 0 it shows the system within the entities like Applicant, User/Admin and Database as shown below.

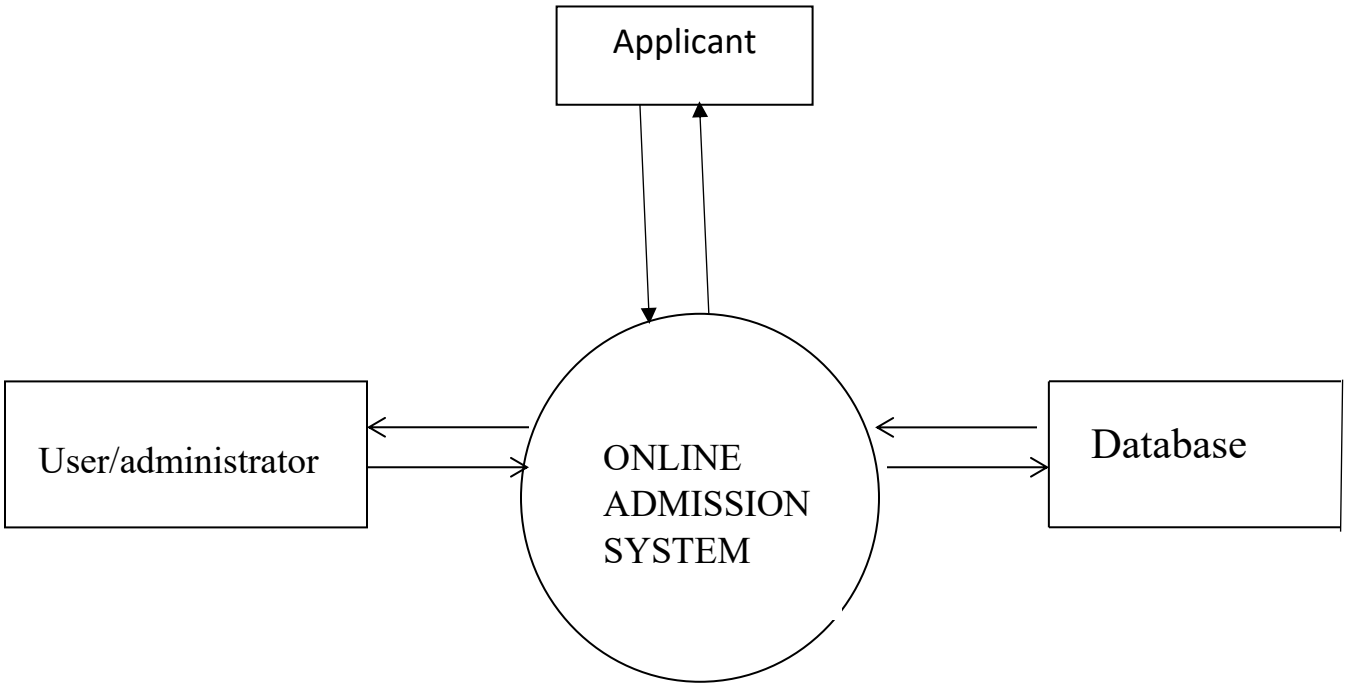


Figure 1: Data flow diagram (Level 0)

Level 1:

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram .It will highlight the main functions carried out by the system, as we break down the high-level process of the Context Diagram into its sub processes.

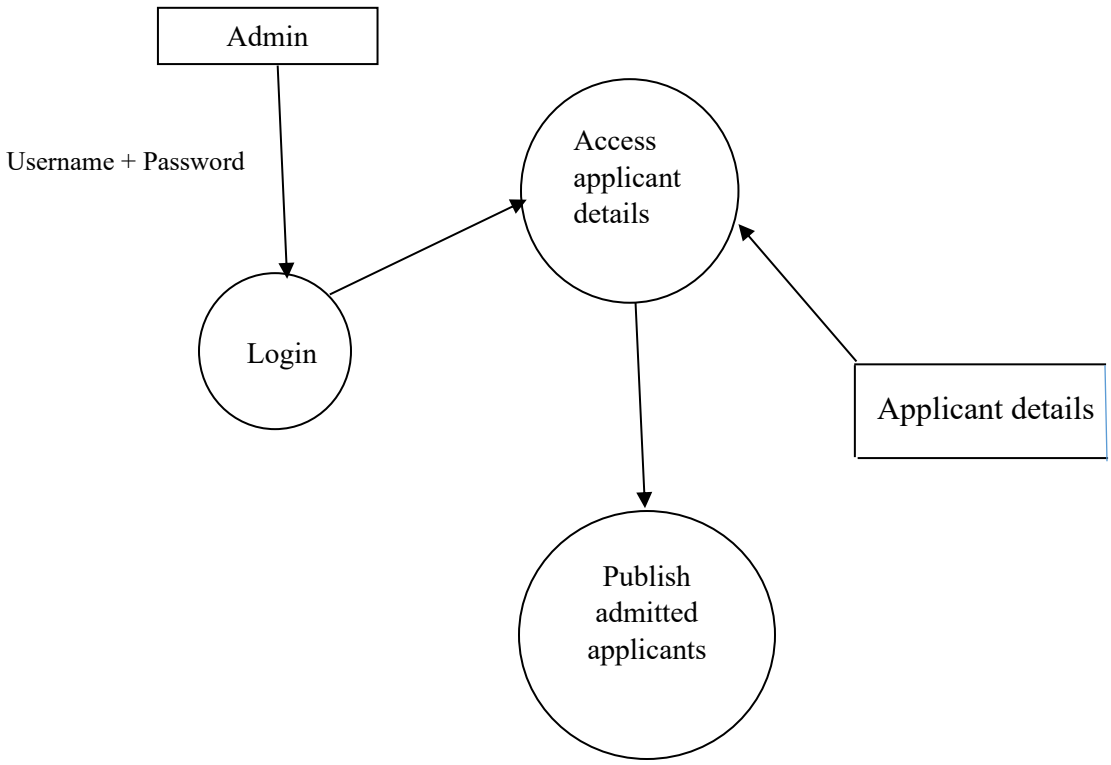


Figure 2: Data flow diagram(Level 1)

Level 2:

DFD Level 2 then goes one step deeper into parts of Level 1. It is required more text to reach the necessary level of detail about the system's functioning. Following this procedure we also draw the ***Level 3*** as shown after this Level 2.

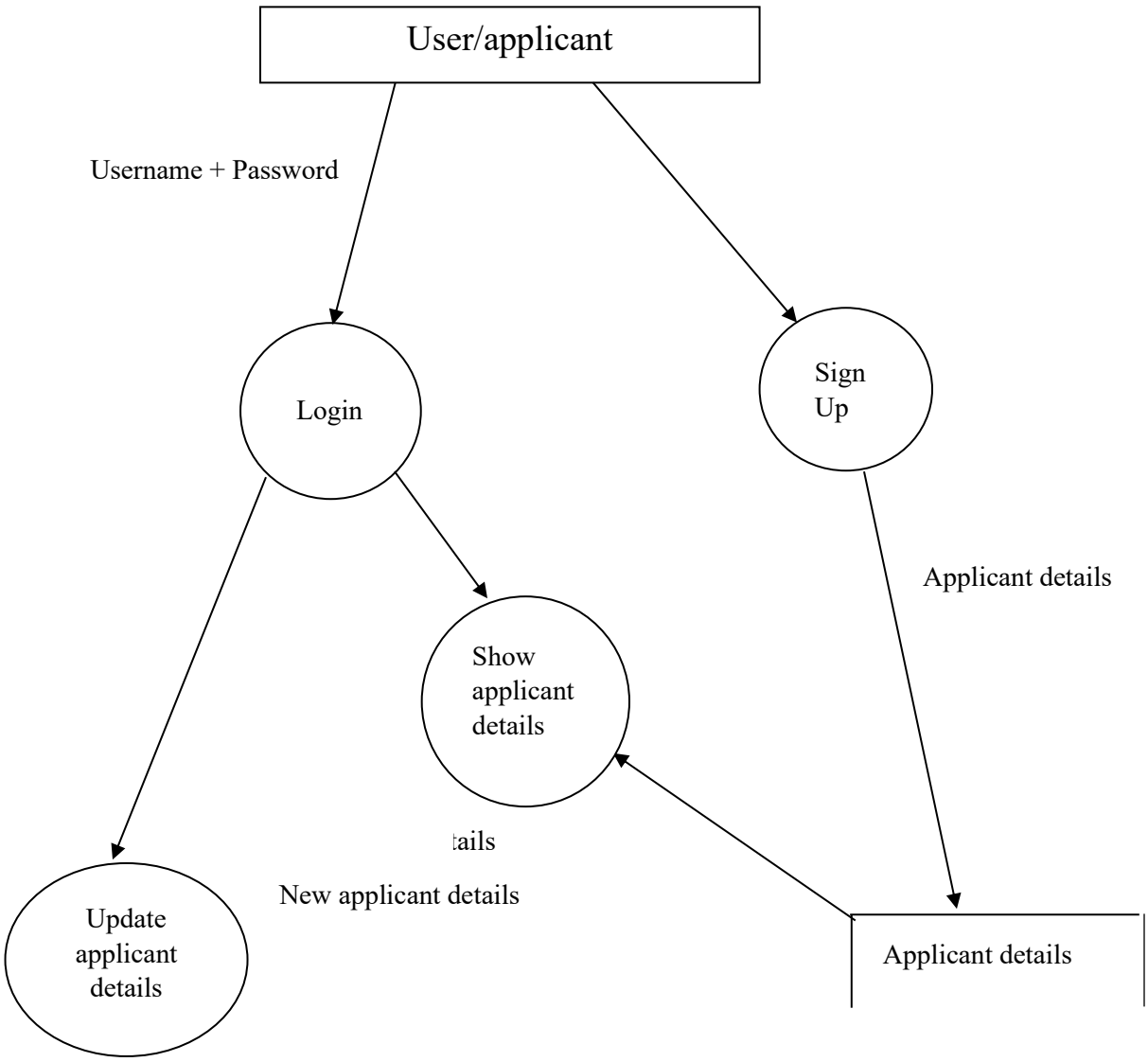
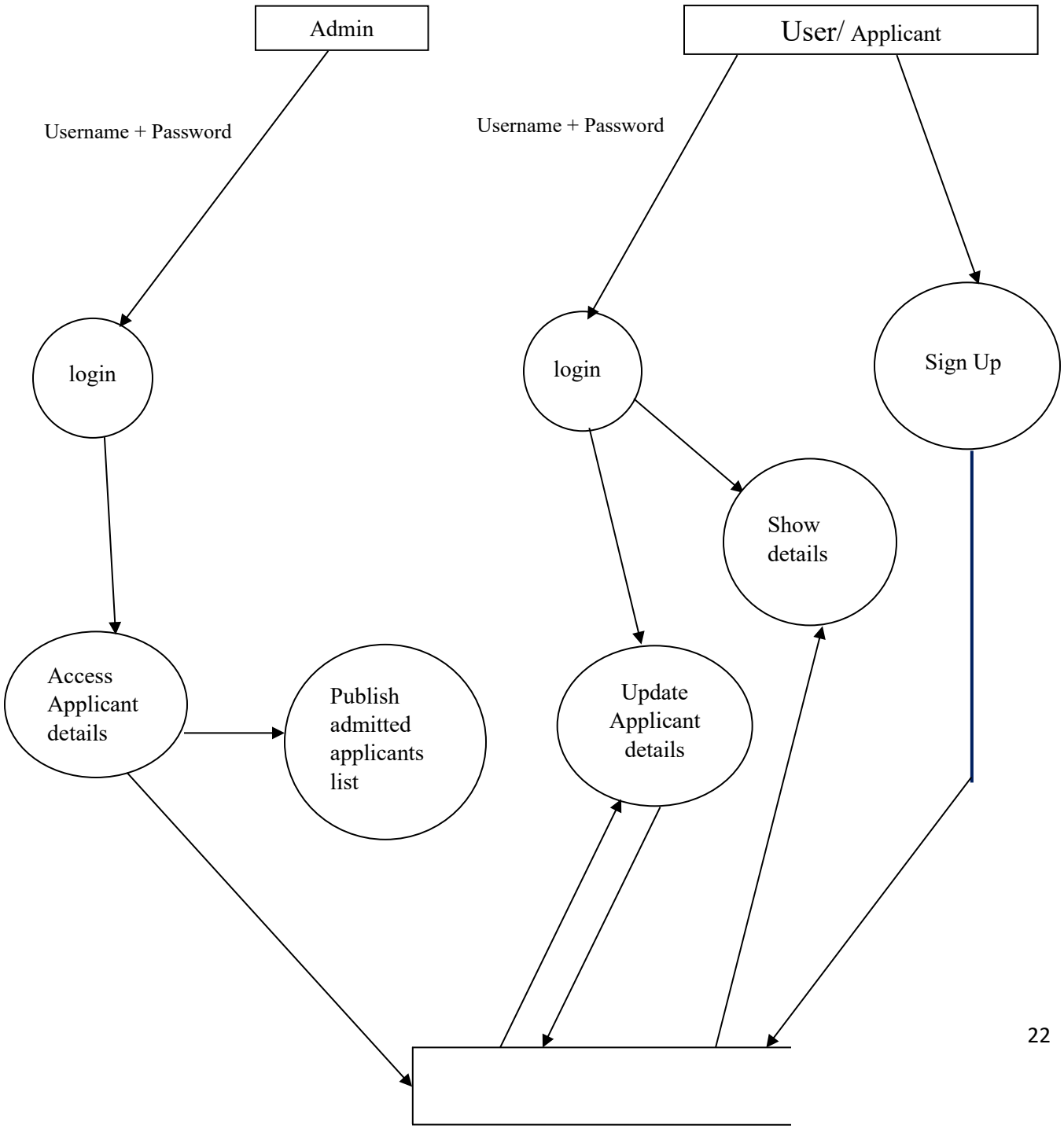


Figure 3: Data flow diagram(Level 2)

Level 3:



Applicant details

Figure 4: Data flow diagram(Level 3)

3.6 E-R Diagram

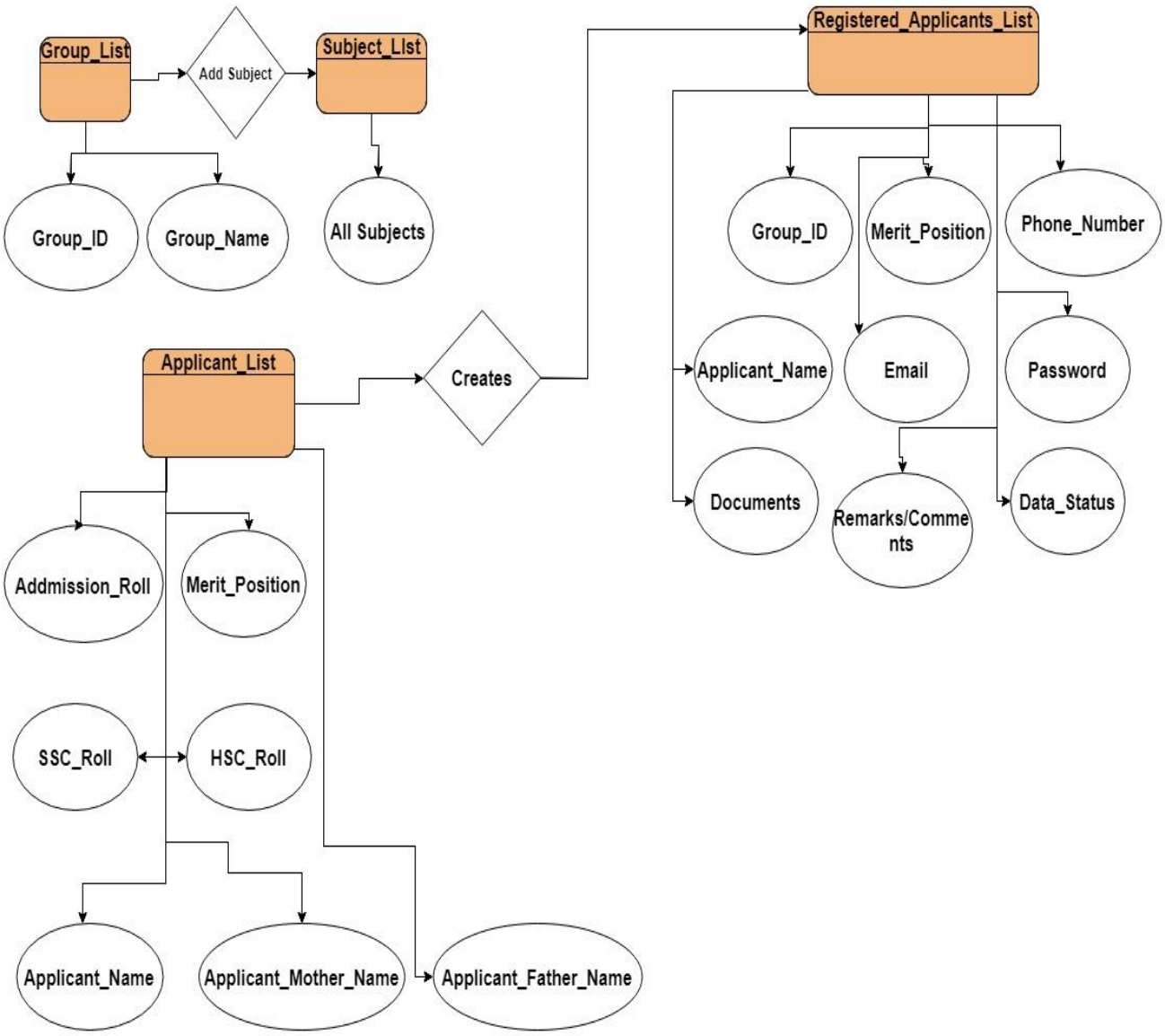


Figure 5: E-R Diagram(Part-1)

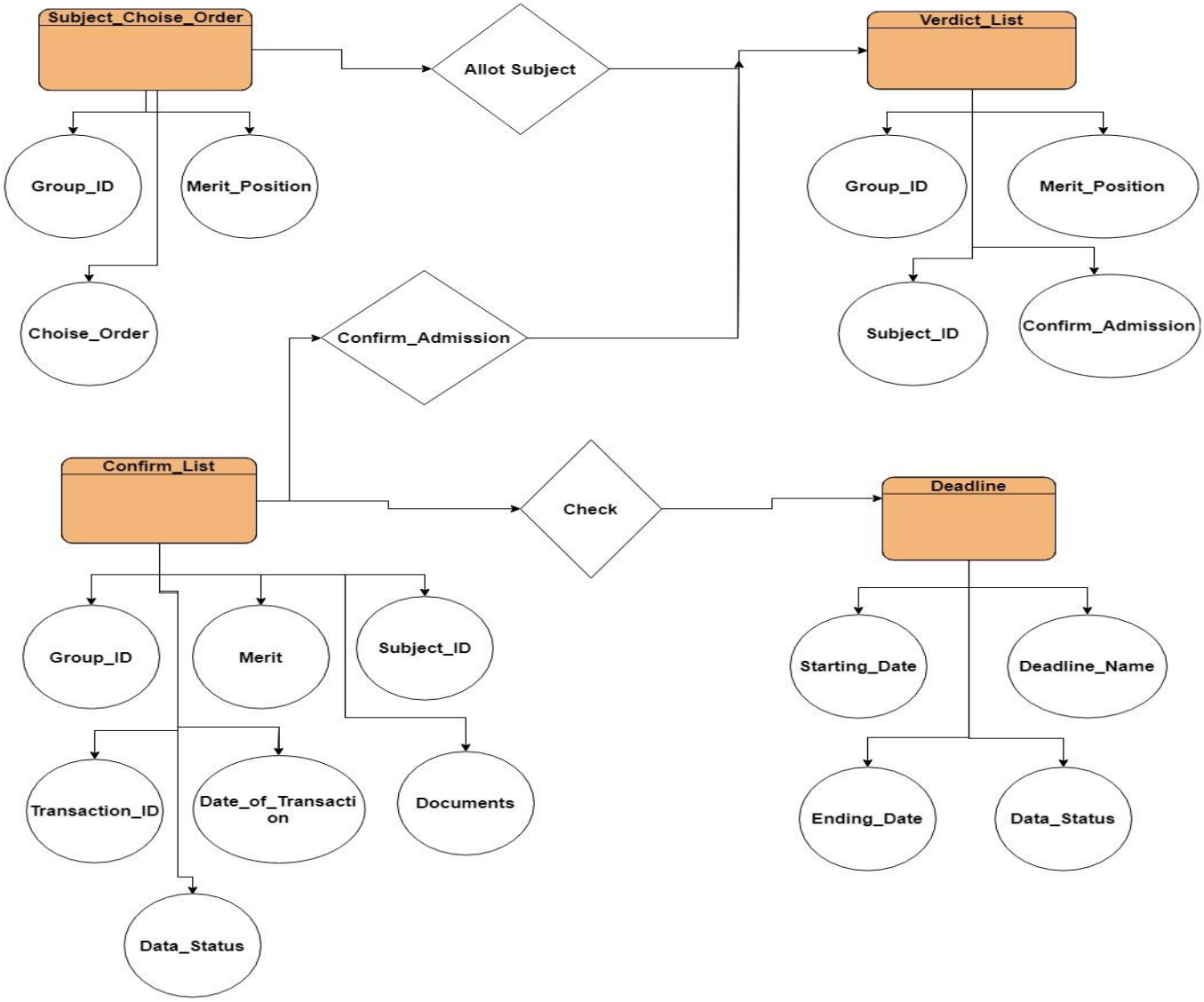


Figure 6: E-R Diagram(Part-2)

3.7 IMPLEMENTATION PLANS

The major activities in implementation plan are cost estimation, schedule and milestone determination, project staffing, quality control plans, and controlling and monitoring plans. The implementation plan involves the following:

- Testing to confirm effectiveness.
- Detection and correction of errors.

The system has two modules. They are

- User
- Administrator

Students logging is to apply for the course by filling an application form provided by online. College principal/administrator logging in may also access/search information put up by the students.

3.6.1 TESTING

Software testing is a critical element of the software development cycle. The testing is essential for ensuring the Quality of the software developed and represents the ultimate view of specification, design and code generation. Software testing is defined as the process by which one detects the defects in the software. Testing is a set of activities that work towards the integration of entire computer based system.

A good test case is one that has a high probability of finding an as-yet undiscovered error. A successful test is one such uncovers or finds such errors. If testing is conducted successfully, it will uncover errors in the software. It also demonstrates that software functions are being performed according to specifications and also behavioural and performance requirements are satisfied. For this, test plans have to be prepared. The implementation of a computer system requires that test data has to be prepared and that all the elements in the system are tested in a planned and efficient manner. Nothing is complete without testing, as it is vital success of the system.

Testing objectives

There are several rules that can serve as testing objectives. They are:

- Testing is process of executing a program and finding a bug.

A good test case is one that has a high probability of finding an undiscovered.

- A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specification, that performance requirements appear to have been met.

Unit Testing

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

This is the first level of testing. In this, codes are written such that from one module, we can move on to the next module according to the choice we enter.

System Testing

In this, the entire system was tested as a whole with all forms, code, modules and class modules. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. It is a series of different tests that verifies that all system elements have been properly integrated and perform allocated functions. System testing makes logical assumptions that if all parts of the system are correct, the goal will be successfully achieved. Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show that software errors are present.

Integration Testing

This testing strategies combines all the modules involved in the system. After the independent modules are tested, dependent modules that use the independent modules are tested. This sequence of testing layers of dependent modules continues until the entire system is constructed.

Though each module individually, they should work after linking them together. Data may be lost across interface and one module can have adverse effect on another. Subroutines, after linking, may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time, conducting test to uncover errors associated with the interface. In the testing the programs are constructed and tested in the small segments.

3.8 Screenshots

Some screenshots describing the project are given below:

Admin Login:

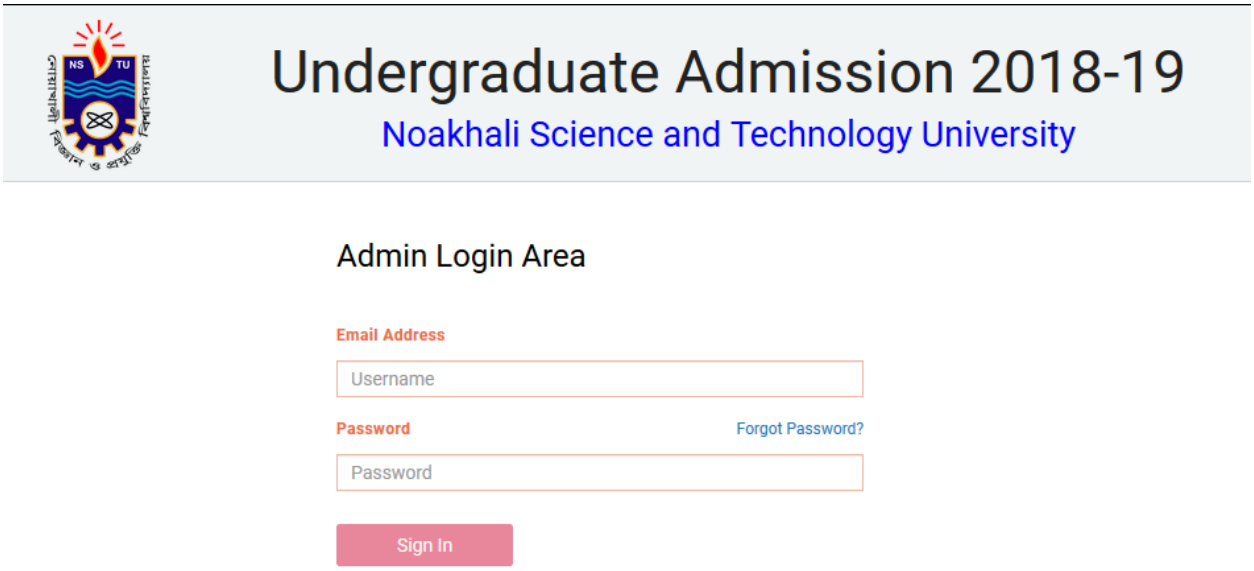


Figure 7: Admin Login

Subject Allot(Admin):

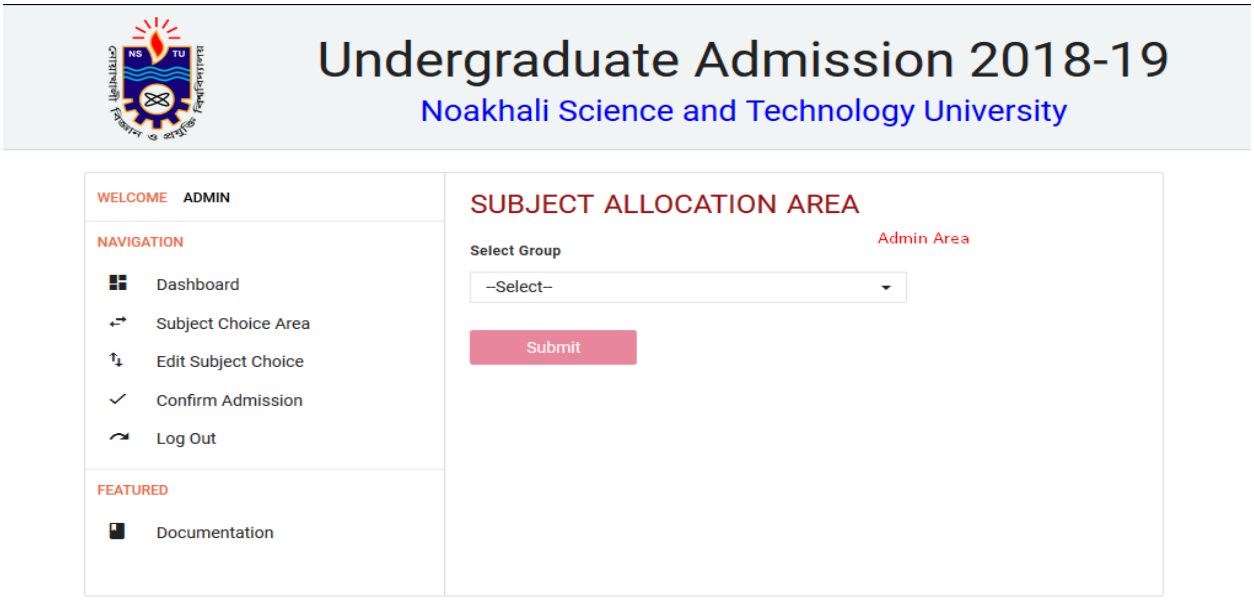


Figure 8: Subject Allotment

Applicant Subject Choise:

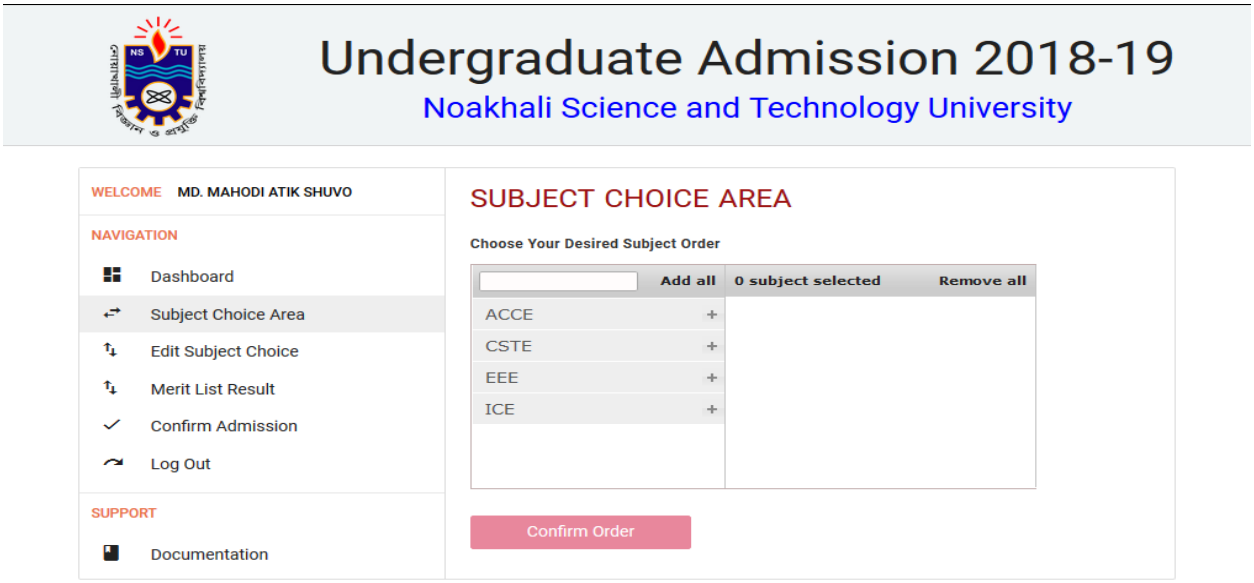


Figure 9: Applicant Subject Choise

Confirm Admission:

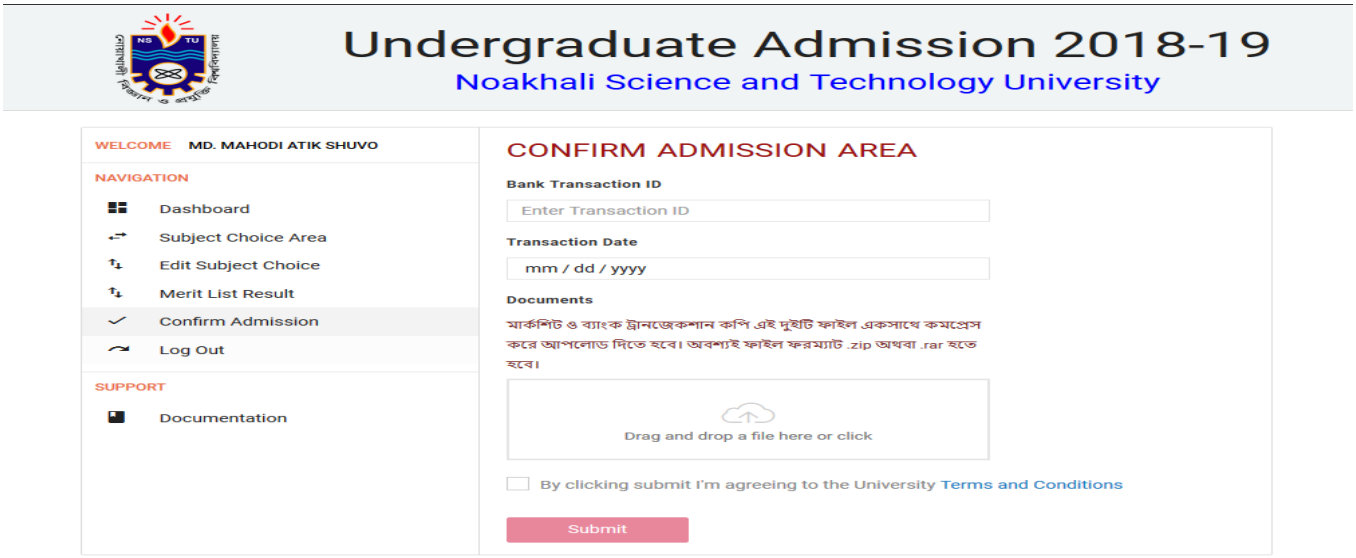


Figure 10: Confirm Admission

Dashbord:

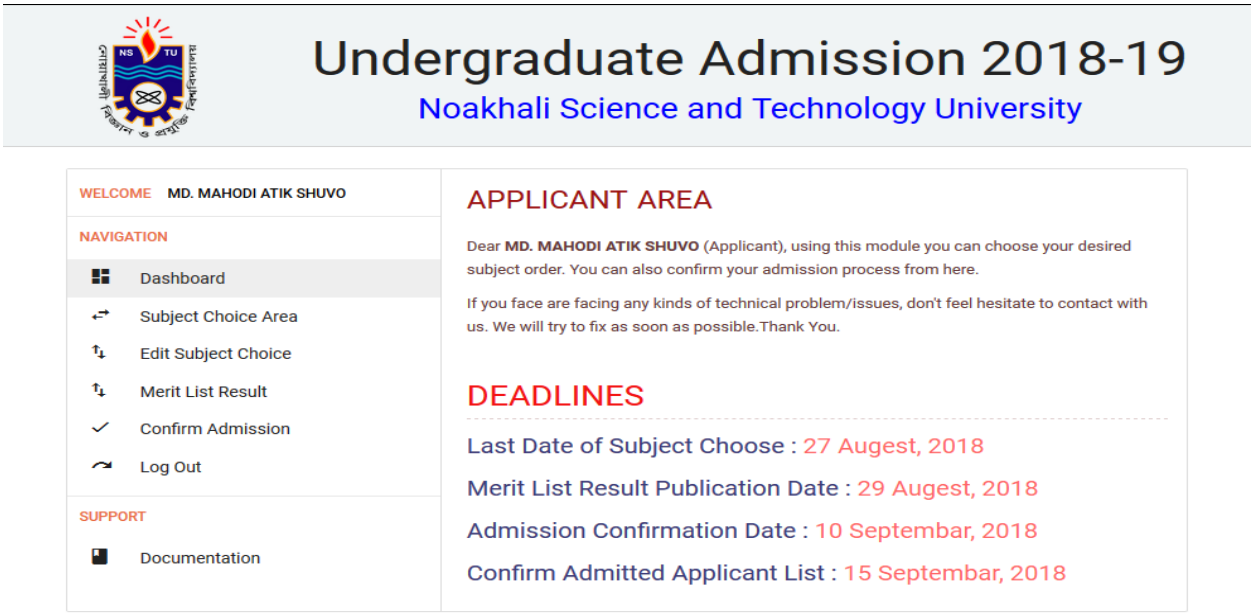


Figure 11: Dashbord

Documentation:

Figure 12: Documentation

Applicant Sign Up:

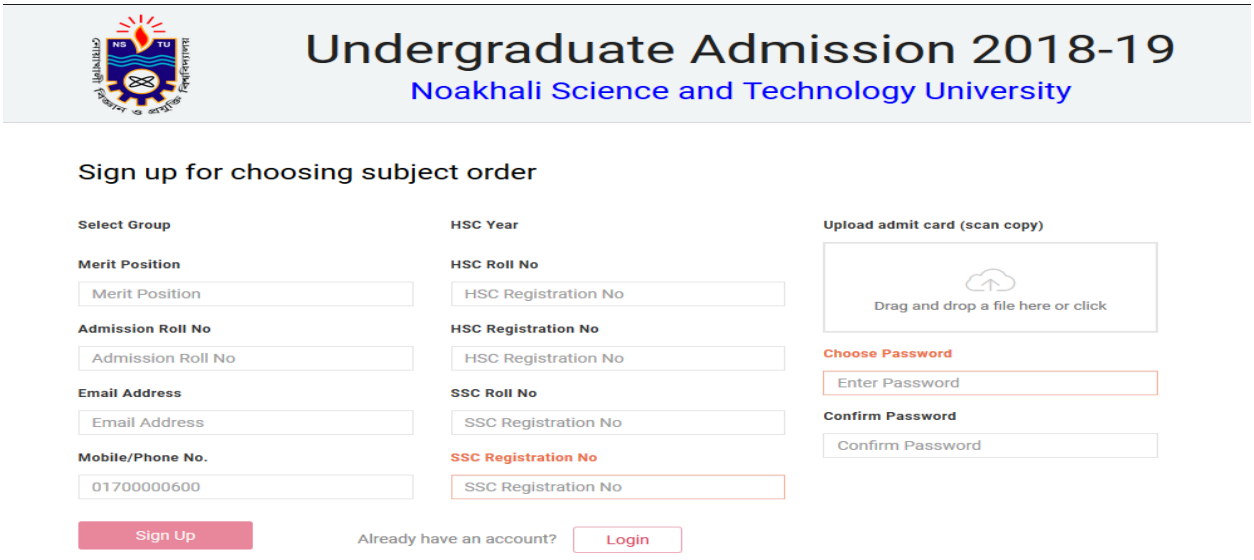


Figure: Applicant Sign Up

Merit List Result:

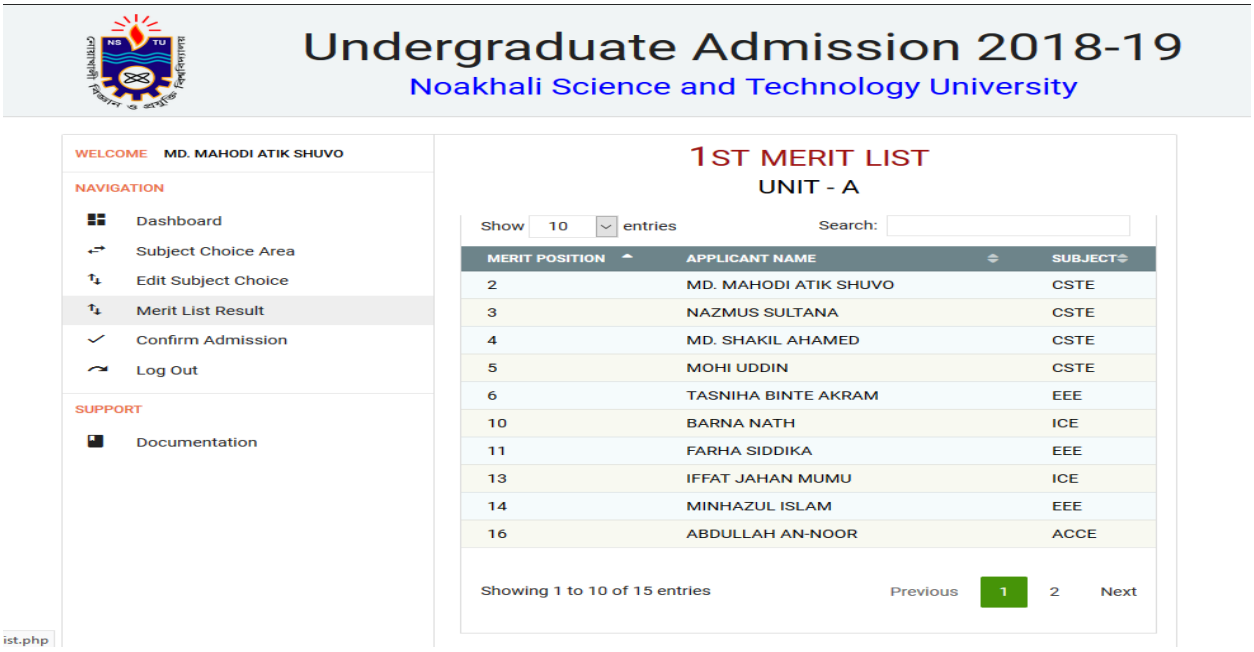


Figure: Merit List Result

CHAPTER 4

OUTCOMES

The Outcomes of this project is very broad. Few of them are:

- This can be implemented in less time for proper admission process
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system.

CHAPTER 5

Conclusion

After completing our project ***ONLINE BASED UNIVERSITY ADMISSION SYSTEM FOR UNDERGRADUATES*** we are sure that all the problems are being solved. The online admission system is the use of a central database. This database is the basis for all actions in the system and can be trivially updated and used to aid in all of the system's processes, meaning all of the required information is stored in one central location and thus is easily accessible. This is a far more reasonable storage method than a paper-based file system, where the time of traveling to and physically searching the records for the required information could be a burden. So, system reduces the consumption of time and work load for humans. System can maintain the detail of all students and can retrieve it at the time of need. Operational efficiency is improved by improving the quality of the process.

Also obtaining the basic aim of the system that is to communicate with the students scattered geographically. And students can have quick access to system.

This system has solved almost all the problems relating with the manual system. Thus helping the users to have quick access and get admission. With the computerization of the system it not only have improved the quality of the processing but also have make this process easy and quick.

CHAPTER 6

REFERENCES

1. ROGER S. PRESSMAN. 2010. *Software Engineering*; *A Practitioner's Approach*. 7th Edition. McGraw-Hill, New York, NY.
2. www.php.net：The PHP Manual, PHP PDO, PHP OPP, File Operation Visited Date: 15/06/2018
3. GILMORE & W.JASON. 2010. *Beginning PHP and MySQL* From Novice to Professional, 4th Edition .Apress, New York, NY.
4. JASON LENGSTORF, THOMAS BLOM HANSEN. 2014. *PHP for Absolute Beginners.* 2nd Edition. Apress, New York, NY.
5. http://www.w3schools.com/css/ HTML5 Tutorial; Visited date: 05/06/2018
6. http://www.tutorialspoint.com/javascript/javascript_overview.htm What is JavaScript; Visited date: 09/06/2018

CHAPTER 7

APPENDIX

Project Code:

Some important part of project codes are given below:

Login Class:

```
<?php
include('dbconnection.php');

class RegisterLogin extends Connection{
      public function
applicantSignUp($GroupID,$Merit,$Name,$Phone,$Email,$Password,$Documents){

            // Entry process
            $statement = $this->db->prepare("insert into
registeredapplicants (GroupID,Merit,Name,Phone,Email,Password,Documents)
values (?,?,?,?,?,?,?)");
            $statement-
>execute(array($GroupID,$Merit,$Name,$Phone,$Email,$Password,$Documents));
            //
      }
      public function CheckDuplicateApplicant($GroupID,$Merit){
            $statement = $this->db->prepare("select GroupID,Merit from
registeredapplicants where GroupID=? and Merit=?");
            $statement->execute(array($GroupID,$Merit));
            $num = $statement->rowCount();
            return $num;
      }
      public function CheckDuplicateUserEmail($email){
            $statement = $this->db->prepare("select Email from
registeredapplicants where Email=?");
            $statement->execute(array($email));
            $num = $statement->rowCount();
            return $num;
      }

      public function getApplicantName($Merit,$table_name){
            $statement = $this->db->prepare("select Merit,Name from
$table_name where Merit=?");
            $statement->execute(array($Merit));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);
            foreach($result as $row){}
            return $row['Name'];
      }
```

```
	public function
CheckApplicantData($AdmissionRoll,$Merit,$HSCRoll,$HSCYear,$HSCReg,$SSCRol
l,$SSCReg,$table_name){
		$statement = $this->db->prepare("select
AdmissionRoll,Merit,HSCRoll,HSCYear,HSCReg,SSCRoll,SSCReg from $table_name
where AdmissionRoll=? and Merit =? and HSCRoll=? and HSCYear=? and
HSCReg=? and SSCRoll=? and SSCReg=?");
		$statement-
>execute(array($AdmissionRoll,$Merit,$HSCRoll,$HSCYear,$HSCReg,$SSCRoll,$S
SCReg));
		$num = $statement->rowCount();
		return $num;
	}


	public function rowCount($query,$data){
		$statement = $this->db->prepare("{$query}");
		$statement->execute(array($data));
		$num = $statement->rowCount();
		return $num;
	}

	public function UserLogin($Email,$Password){
		$statement = $this->db->prepare("select
Email,Password,Merit,GroupID,Name from registeredapplicants where Email=?
and Password=?");
		$statement->execute(array($Email,$Password));
		$result = $statement->fetchAll(PDO::FETCH_ASSOC);
		$num = $statement->rowCount();

		if($num > 0){
			foreach($result as $row){
				session_start();
				$_SESSION['UserLogin'] = "HelloUserLogin";
				$_SESSION['Merit'] = $row['Merit'];
				$_SESSION['GroupID'] = $row['GroupID'];
				$_SESSION['Name'] = $row['Name'];
				return true;
			}
		}
		else{
			return false;
		}
	}
	// Check post data security check
	public function test_input($data) {
	  $data = trim($data);
	  $data = stripslashes($data);
	  $data = htmlspecialchars($data);
	  return $data;
	}
	//
```

```
}

?>
```

Class Project Function:

```
<?php
include('dbconnection.php');

class ProjectFunctions extends Connection{

      // Get subject list
      public function SubjectList($GroupID){
            $statement = $this->db->prepare("SELECT * FROM subjectlist
where GroupID=? order by SubjectID ASC");
            $statement->execute(array($GroupID));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);

            foreach($result as $row){
                  echo "<option
value='".$row['SubjectID']."'>".$row['SubjectName']."</option>";
            }
      }
      //

      // Get Group list
      public function GroupList(){
            $statement = $this->db->prepare("SELECT * FROM grouplist order
by GroupName ASC");
            $statement->execute();
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);

            foreach($result as $row){
                  echo "<option
value='".$row['GroupID']."'>".$row['GroupName']."</option>";
            }
      }
      //

      // Get subject list
      public function EditSubjectOrder($GroupID,$subject_order){
            $statement = $this->db->prepare("SELECT * FROM subjectlist
where GroupID=? order by SubjectID ASC");
            $statement->execute(array($GroupID));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);

            foreach($result as $row){
                  if($subject_order !== null){

                        $old_post_cat_arr = explode(",",$subject_order);
                        $old_post_cat_arr_size = count($old_post_cat_arr);
                        $k=0;
```

```
                        for($i=0;$i < $old_post_cat_arr_size;$i++){
                              if($old_post_cat_arr[$i] ==
$row['SubjectID']){
                                    $k=1;
                                    break;
                              }
                        }

                        if($k == 1){
                              echo "<option
value='".$row['SubjectID']."'selected>".$row['SubjectName']."</option>";
                        }
                        else{
                              echo "<option
value='".$row['SubjectID']."'>".$row['SubjectName']."</option>";
                        }

                  }
                  else{
                        echo "<option
value='".$row['SubjectID']."'>".$row['SubjectName']."</option>";
                  }
            }
      }
      //


      // Get subject choise order of a specific applicant
      public function getChooseOrder($GroupID,$Merit){
            $statement = $this->db->prepare("select * from
applicantchoiseorder where GroupID=? and Merit=?");
            $statement->execute(array($GroupID,$Merit));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);
            foreach($result as $row){}
            return $row['ChoiseOrder'];
      }

      // Get subject Name by ID
      public function getSubjectInfo($SubjectID){
            $statement = $this->db->prepare("select * from subjectlist
where SubjectID=?");
            $statement->execute(array($SubjectID));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);
            return $result;
      }

      // Get Unit Name by ID
      public function getGroupName($GroupID){
            $statement = $this->db->prepare("select * from grouplist where
GroupID=?");
            $statement->execute(array($GroupID));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);
            foreach($result as $row){}
            return $row['GroupName'];
```

```
	}
	//

	// Get Subject Name by ID
	public function getSubjectNameByID($SubjectID){
		$statement = $this->db->prepare("select SubjectName from
subjectlist where SubjectID=?");
		$statement->execute(array($SubjectID));
		$result = $statement->fetchAll(PDO::FETCH_ASSOC);
		foreach($result as $row){}
		return $row['SubjectName'];
	}
	//

	// Insert subject order
	public function addSubjectChooseOrder($GroupID,$Merit,$ChoiseOrder){
		// Entry process
		$statement = $this->db->prepare("insert into
applicantchoiseorder (GroupID,Merit,ChoiseOrder) values (?,?,?)");
		$statement->execute(array($GroupID,$Merit,$ChoiseOrder));
		//
	}

	// Edit subject choice order
	public function
EditSubjectChooseOrder($GroupID,$Merit,$ChoiseOrder){
		// update applicantchoiseorder table
		$statement = $this->db->prepare("update applicantchoiseorder
set ChoiseOrder=? where GroupID=? and Merit=?");
		$statement->execute(array($ChoiseOrder,$GroupID,$Merit,));
		//
	}
	// Check Duplicate applicant choose
	public function CheckDuplicateChoice($GroupID,$Merit){
		$statement = $this->db->prepare("select GroupID,Merit from
applicantchoiseorder where GroupID=? and Merit=?");
		$statement->execute(array($GroupID,$Merit));
		$num = $statement->rowCount();
		return $num;
	}
	//

	// Allocate subject to applicants
	public function AllocateSubject($GroupID){
		$statement = $this->db->prepare("select
GroupID,Merit,ChoiseOrder from applicantchoiseorder where GroupID=? order
by Merit ASC");
		$statement->execute(array($GroupID));
		$result = $statement->fetchAll(PDO::FETCH_ASSOC);
		foreach($result as $row){
			$ChoiseOrder = explode(",",$row['ChoiseOrder']);


			$arr_size = count($ChoiseOrder);
```

```
                for($i=0;$i < $arr_size;$i++){

                        // Calculate the number of completed seat
                        $statement1 = $this->db->prepare("select
SubjectID,GroupID from verdictlist where GroupID=? and SubjectID=?");
                        $statement1-
>execute(array($GroupID,$ChoiseOrder[$i]));
                        $num_of_completed = $statement1->rowCount();
                        //

                        // Get the number of seat of a centain subject
                        $statement2 = $this->db->prepare("select
GeneralSeat from subjectlist where SubjectID=?");
                        $statement2->execute(array($ChoiseOrder[$i]));
                        $result2 = $statement2->fetchAll(PDO::FETCH_ASSOC);
                        foreach($result2 as $row2){}
                        $number_of_seat = $row2['GeneralSeat'];
                        //

                        $available_seat = $number_of_seat -
$num_of_completed;

                        if($available_seat > 0){
                              $statement3 = $this->db->prepare("insert into
verdictlist (GroupID,Merit,SubjectID) values (?,?,?)");
                              $statement3-
>execute(array($GroupID,$row['Merit'],$ChoiseOrder[$i]));
                              break;
                        }


                }
          }
     }
     //

     // Get group based verdict list
     public function getGroupBasedVerdict($GroupID){
          $statement = $this->db->prepare("select * from verdictlist
where GroupID=?");
          $statement->execute(array($GroupID));
          $result = $statement->fetchAll(PDO::FETCH_ASSOC);
          return $result;
     }
     //

     // Get applicant Name by merit & group ID
     public function getApplicantNameByInfo($Merit,$GroupID){

          if($GroupID = 1){
               $table_name = "applicant_list_a";
          }
          else if($GroupID = 2){
```

```
                $table_name = "applicant_list_b";
            }
            else if($GroupID = 3){
                $table_name = "applicant_list_c";
            }
            else if($GroupID = 4){
                $table_name = "applicant_list_d";
            }
            else{
                $table_name = "";
            }

            $statement = $this->db->prepare("select Merit,Name from
$table_name where Merit=?");
            $statement->execute(array($Merit));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);
            foreach($result as $row){}
            return $row['Name'];
      }

      // Add applicant to confirm list
      public function
addApplicantToConfirm($GroupID,$Merit,$SubjectID,$TransactionID,$Transacti
onDate,$Documents){
            // Entry process
            $statement = $this->db->prepare("insert into confirmlist
(GroupID,Merit,SubjectID,TransactionID,TransactionDate,Documents) values
(?,?,?,?,?,?)");
            $statement-
>execute(array($GroupID,$Merit,$SubjectID,$TransactionID,$TransactionDate,
$Documents));
            //
      }

      // Check an applicant is valid for confirm admission
      public function isApplicantIsValidForAd($GroupID,$Merit){
            $statement = $this->db->prepare("select SubjectID from
verdictlist where GroupID=? and Merit=?");
            $statement->execute(array($GroupID,$Merit));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);
            return $result;
      }
      //


      // Check diplicate admission entry
      public function checkDuplicateAdmission($GroupID,$Merit){
            $statement = $this->db->prepare("select GroupID,Merit from
confirmlist where GroupID=? and Merit=?");
            $statement->execute(array($GroupID,$Merit));
            $number_of_entry = $statement->rowCount();
            return $number_of_entry;
      }
      //
```

```
      // Finding accoount approved status || return ture, false
      public function AccountApporvedStatus($profile_id){
            $statement = $this->db->prepare("SELECT profile_id,IsApproved
FROM u_users where profile_id=?");
            $statement->execute(array($profile_id));
            $result = $statement->fetchAll(PDO::FETCH_ASSOC);
            foreach($result as $row){}
            //return $row['IsApproved'];

            if($row['IsApproved'] == 1){
                  return true;
            }
            else{
                  return false;
            }
      }
      //


}

?>
```