**Chapter 1: Introduction**

* 1. **Overview of the System**

The “Student Information System” has been developed to overcome the problems existing in the manual system. This software is developed to eliminate and reduce the hardships faced by the existing system. Moreover this system is designed for the particular need of the administration to carry out operations in a smooth and effective manner.

This application is developed to speed up the operations as much as possible and to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this it proves that it is user-friendly. Student Information system, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

* 1. **Problem Statement**

The data of students is written on Microsoft excel sheet, taking name, ID, marks each column, attendance on a paper. Time consumption is very huge with lack of security. Any unauthorized person can get access to all the information. There is maximum chance of losing data in the manual system. There is no privacy of student data. We don’t know what action is being performed by whom.

Student’s data can be modified without any permission. There is no security for information and can easily be lost.

* 1. **Proposed System**

The **Student Information System** is developed to store student details like ID, name, marks obtained by the student during exams, attendance records and everything related to the payment section.

As our system is being developed at administrative end, it will help the administrator to reduce the workload that exists in the manual system and minimizing the probability of losing any data. This system will change the way of viewing the data and help them to do their work in a very less amount of time.

This system will provide user friendly approach and eliminate the hard work faced by the existing (manual) system.

* 1. **Objectives**

The main objective of the project **Student Information System** is to manage the details of the students. This project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the information of the students. It showcases all the details of the student, their courses, address, marks, payment details etc.

**Advantages of using Student Information System**:

* Provides smart management of student’s data.
* It manages the information of all the students.
* Provides the facility of searching student’s information based on their enrollment number.
* It provides editing, adding and updating of records which results in proper resource management of student data.
* More work in less time.
* Accuracy in works.
* Well designed reports.

**Chapter 2: Requirement Specification**

**2.1. Tools and Technology used**

**Hardware Specifications:**

* Processor : Intel Core i3
* Hard Disk: 1 TB
* RAM: 4 GB DDR3
* Processor Variant: 6006U

**Software to be used:**

* Operating System: Microsoft Windows 10
* Front End: Microsoft Visual Studio 2008
* Back End: MySQL Server 2005

**2.2. Feasibility Study**

The objective of feasibility is just not to solve the problem but also to acquire a sense of its scope. The three major areas to consider while determining the feasibility of a project are:

1. Behavioral Feasibility:The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system.
2. Economically Feasibility: The system being developed is economic with respect to University’s point of view. It is cost effective in the sense that has eliminated the paper work completely.

The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

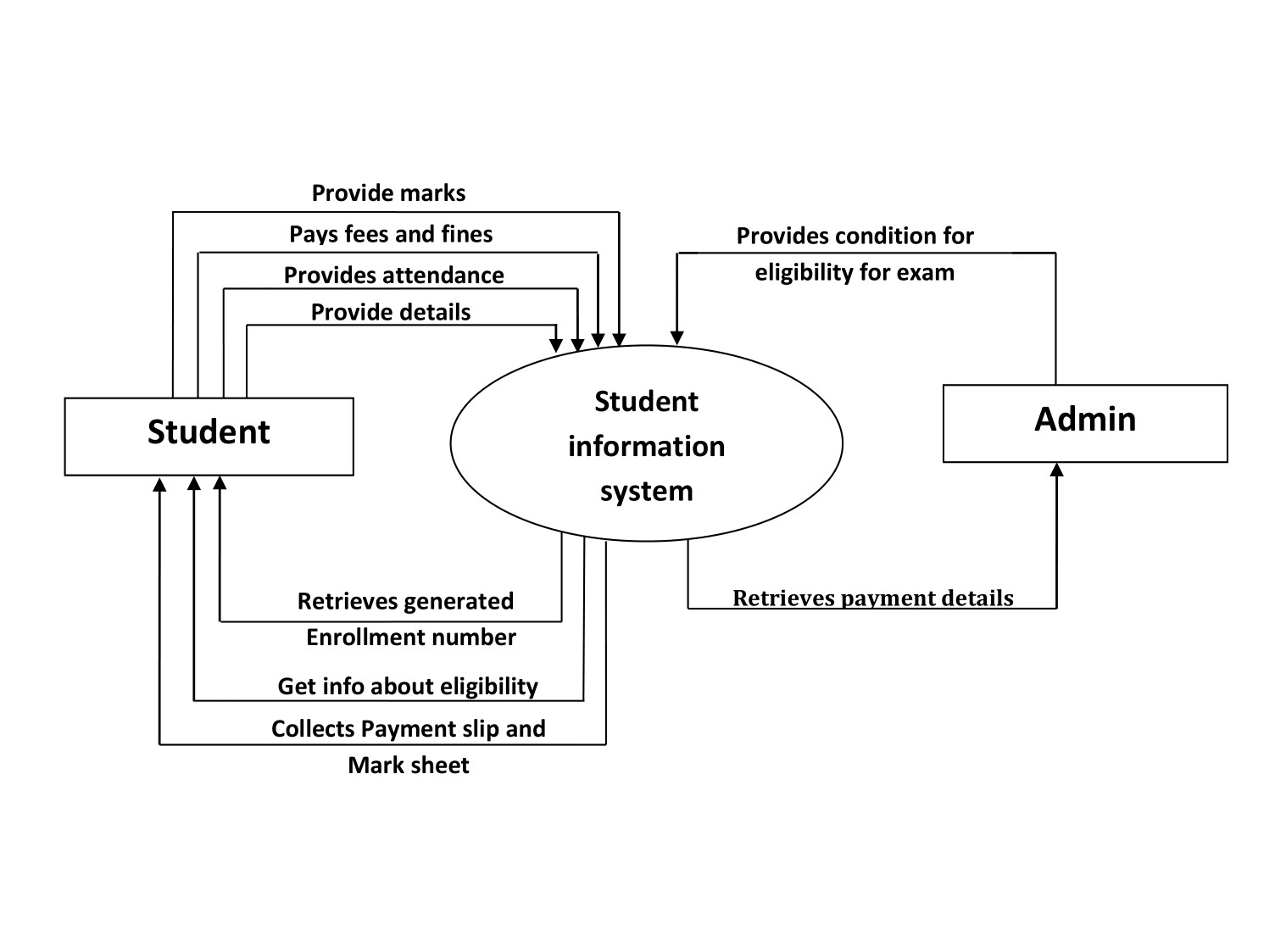
The result obtained contains minimum errors and are highly accurate as the data is required.

1. Technical Feasibility: The System working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system.

**Chapter 3: System Design**

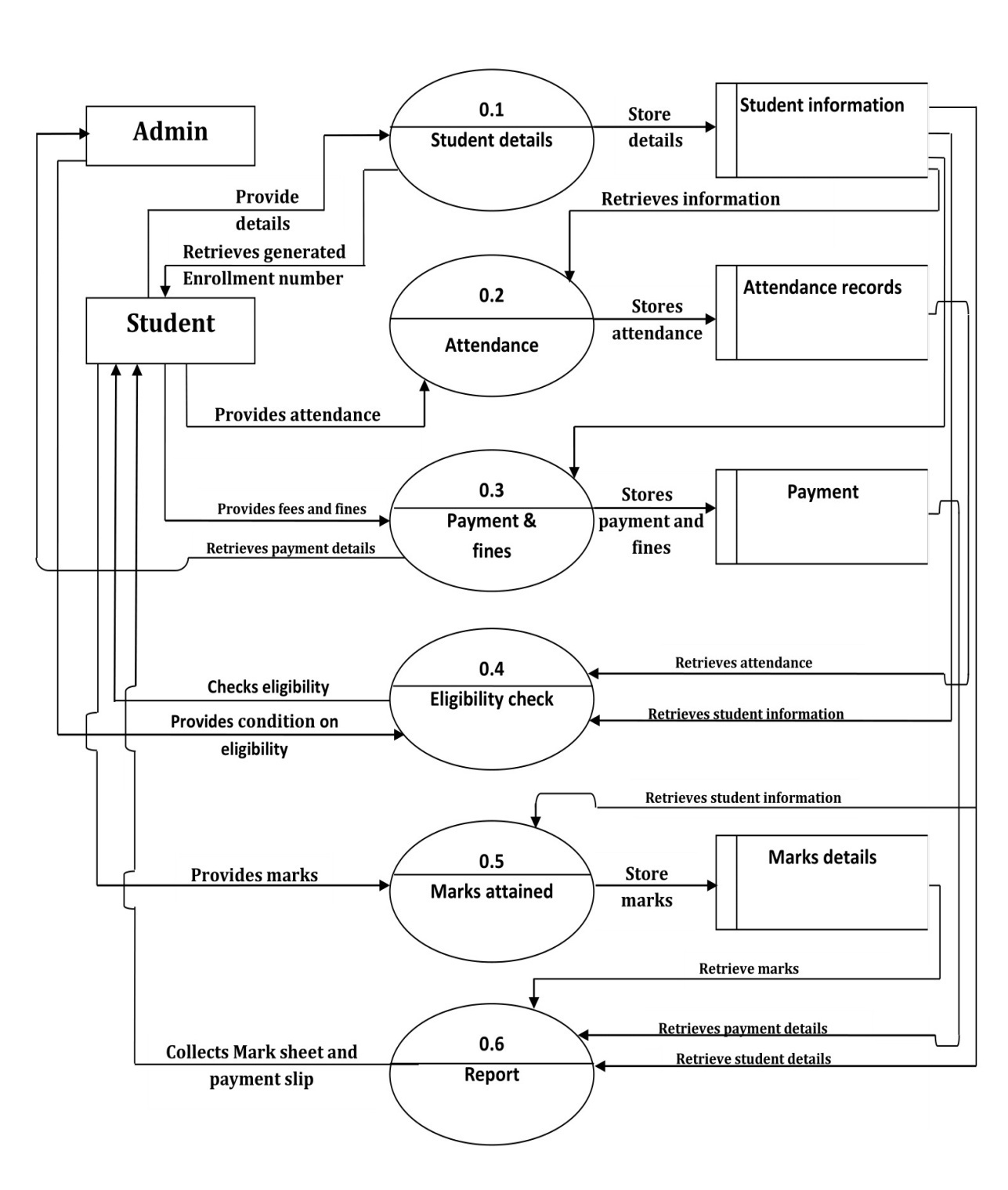
**3.1. Data Flow Diagrams**

**3.1.1. Zero level Data Flow (context) Diagram**

****

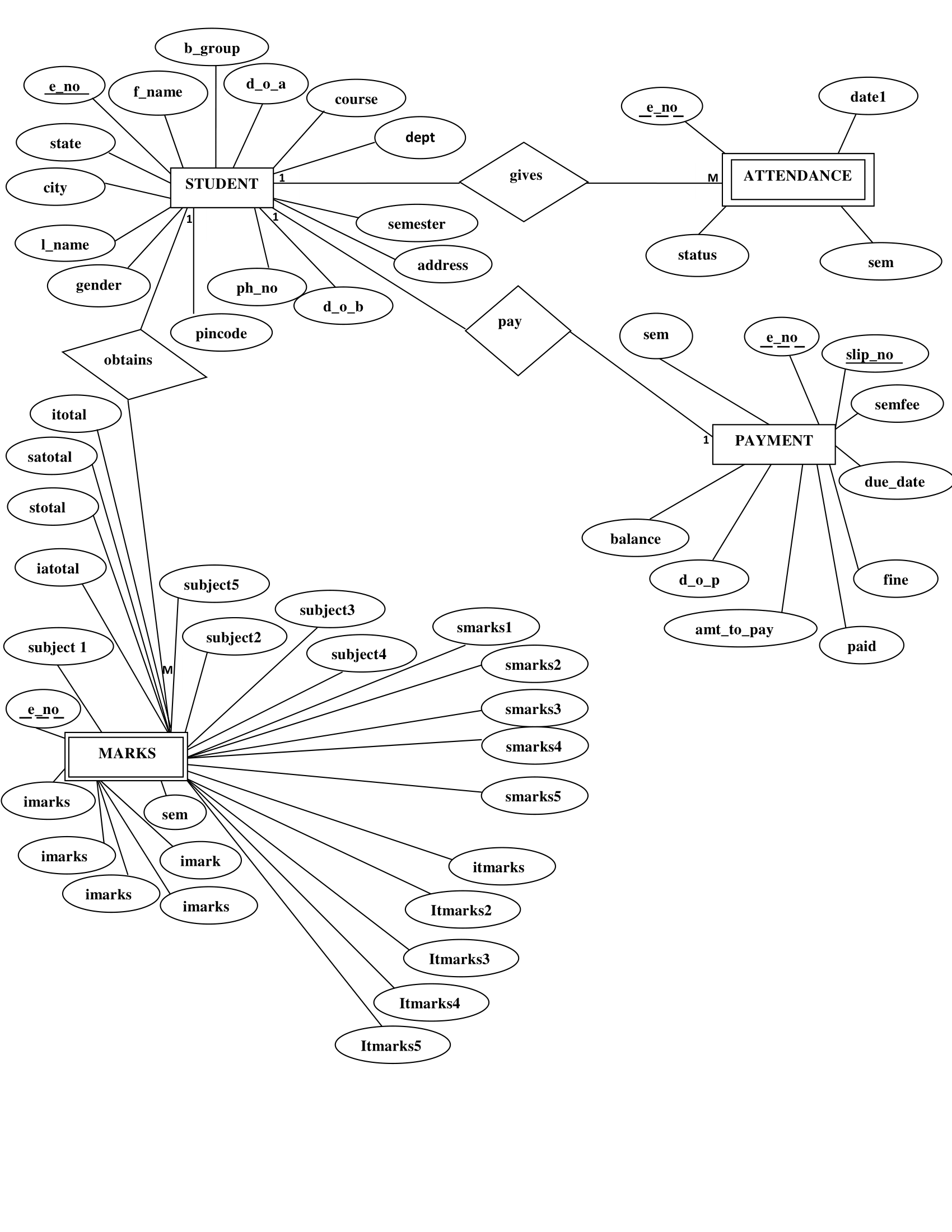
**Fig 3.1 Zero level Data Flow Diagram**

**3.1.2. First level Data Flow Diagram**

****

**Fig 3.2 First level data flow diagram**

**3.2. ER Diagram**

****

**Fig 3.3 ER Diagram**

**3.3. Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Source table** | **Constraint** |
| **address** | **Student's Address** | **Student** |  |
| **amt\_to\_pay** | **Amount to be paid** | **Payment** |  |
| **b\_group** | **Blood group** | **Student** |  |
| **balance** | **Payment balance** | **Payment** |  |
| **city** | **City where student lives** | **Student** |  |
| **course** | **Course taken by student** | **Student** |  |
| **d\_o\_a** | **Date of admission** | **Student** |  |
| **d\_o\_b** | **Date of birth** | **Student** |  |
| **d\_o\_p** | **Date of payment** | **Payment** |  |
| **date1** | **Date of Attendance** | **Attendance** |  |
| **dept** | **Department of the student** | **Student** |  |
| **due\_date** | **Due date** | **Payment** |  |
| **e\_no** | **Enrollment number** | **Student** | **primary key** |
| **f\_name** | **First Name of the student** | **Student** |  |
| **fine** | **Fines** | **Payment** |  |
| **gender** | **Gender of the student** | **Student** |  |
| **iatotal** | **Internal exam total marks achieved** | **Marks** |  |
| **imarks1** | **Internal exam subject 1 marks** | **Marks** |  |
| **imarks2** | **Internal exam subject 2 marks** | **Marks** |  |
| **imarks3** | **Internal exam subject 3 marks** | **Marks** |  |
| **imarks4** | **Internal exam subject 4 marks** | **Marks** |  |
| **imarks5** | **Internal exam subject 5 marks** | **Marks** |  |
| **itmarks1** | **Internal exam subject 1 total marks** | **Marks** |  |
| **itmarks2** | **Internal exam subject 2 total marks** | **Marks** |  |
| **itmarks3** | **Internal exam subject 3 total marks** | **Marks** |  |
| **itmarks4** | **Internal exam subject 4 total marks** | **Marks** |  |
| **itmarks5** | **Internal exam subject 5 total marks** | **Marks** |  |
| **itotal** | **Internal exam all subject total marks** | **Marks** |  |
| **l\_name** | **Last Name of the student** | **Student** |  |
| **paid** | **Payment paid** | **Payment** |  |
| **ph\_no** | **Phone number of the student** | **Student** |  |
| **pincode** | **Pin code** | **Student** |  |
| **satotal** | **Semester exam total marks achieved** | **Marks** |  |
| **sem** | **Semester** | **Attendance** |  |
| **sem\_fee** | **Semester fees** | **Payment** |  |
| **semester** | **Semester** | **Student** |  |
| **slip\_no** | **Slip number** | **Payment** | **primary key** |
| **smarks1** | **Semester exam subject 1 marks** | **Marks** |  |
| **smarks2** | **Semester exam subject 2 marks** | **Marks** |  |
| **smarks3** | **Semester exam subject 3 marks** | **Marks** |  |
| **smarks4** | **Semester exam subject 4 marks** | **Marks** |  |
| **smarks5** | **Semester exam subject 5 marks** | **Marks** |  |
| **state** | **State where student lives** | **Student** |  |
| **status** | **Student present or absent** | **Attendance** |  |
| **stmarks1** | **Semester exam subject 1 total marks** | **Marks** |  |
| **stmarks2** | **Semester exam subject 2 total marks** | **Marks** |  |
| **stmarks3** | **Semester exam subject 3 total marks** | **Marks** |  |
| **stmarks4** | **Semester exam subject 4 total marks** | **Marks** |  |
| **stmarks5** | **Semester exam subject 5 total marks** | **Marks** |  |
| **stotal** | **Semester exam all subject total marks** | **Marks** |  |
| **subject1** | **Name of subject 1** | **Marks** |  |
| **subject2** | **Name of subject 2** | **Marks** |  |
| **subject3** | **Name of subject 3** | **Marks** |  |
| **subject4** | **Name of subject 4** | **Marks** |  |
| **subject5** | **Name of subject 5** | **Marks** |  |

**Table 3.3.1 Data Dictionary**

**3.4. Table Structure**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Size** | **Constraint** |
| **e\_no** | **int** | **11** | **primary key** |
| **f\_name** | **varchar** | **30** |  |
| **l\_name** | **varchar** | **30** |  |
| **gender** | **varchar** | **7** |  |
| **d\_o\_a** | **date** | **8** |  |
| **course** | **varchar** | **70** |  |
| **dept** | **varchar** | **90** |  |
| **semester** | **int** | **11** |  |
| **d\_o\_b** | **date** | **8** |  |
| **address** | **varchar** | **100** |  |
| **ph\_no** | **bigint** | **25** |  |
| **b\_group** | **varchar** | **10** |  |
| **pincode** | **int** | **6** |  |
| **city** | **varchar** | **40** |  |
| **state** | **varchar** | **40** |  |

**Table 3.4.1 Student**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Size** | **Constraint** |
| **e\_no** | **int** | **11** | **foreign key** |
| **sem** | **int** | **11** |  |
| **date1** | **date** | **8** |  |
| **status** | **varchar** | **11** |  |

**Table 3.4.2 Attendance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Size** | **Constraint** |
| **e\_no** | **int** | **11** | **foreign key** |
| **sem** | **int** | **2** |  |
| **subject1** | **varchar** | **100** |  |
| **subject2** | **varchar** | **100** |  |
| **subject3** | **varchar** | **100** |  |
| **subject4** | **varchar** | **100** |  |
| **subject5** | **varchar** | **100** |  |
| **imarks1** | **int** | **4** |  |
| **imarks2** | **int** | **4** |  |
| **imarks3** | **int** | **4** |  |
| **imarks4** | **int** | **4** |  |
| **imarks5** | **int** | **4** |  |
| **itmarks1** | **int** | **4** |  |
| **itmarks2** | **int** | **4** |  |
| **itmarks3** | **int** | **4** |  |
| **itmarks4** | **int** | **4** |  |
| **itmarks5** | **int** | **4** |  |
| **itotal** | **int** | **4** |  |
| **iatotal** | **int** | **4** |  |
| **smarks1** | **int** | **4** |  |
| **smarks2** | **int** | **4** |  |
| **smarks3** | **int** | **4** |  |
| **smarks4** | **int** | **4** |  |
| **smarks5** | **int** | **4** |  |
| **stmarks1** | **int** | **4** |  |
| **stmarks2** | **int** | **4** |  |
| **stmarks3** | **int** | **4** |  |
| **stmarks4** | **int** | **4** |  |
| **stmarks5** | **int** | **4** |  |
| **stotal** | **int** | **4** |  |
| **satotal** | **int** | **4** |  |

**Table 3.4.3 Marks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Size** | **Constraint** |
| **slip\_no** | **int** | **11** | **primary key** |
| **e\_no** | **int** | **11** | **foreign key** |
| **sem** | **int** | **11** |  |
| **sem\_fee** | **decimal** | **8,2** |  |
| **due\_date** | **date** | **8** |  |
| **d\_o\_p** | **date** | **8** |  |
| **amt\_to\_pay** | **decimal** | **8,2** |  |
| **paid** | **decimal** | **8,2** |  |
| **fine** | **decimal** | **8,2** |  |
| **balance** | **decimal** | **8,2** |  |

**Table 3.4.4 Payment**

**Chapter 4: Testing and Implementation**

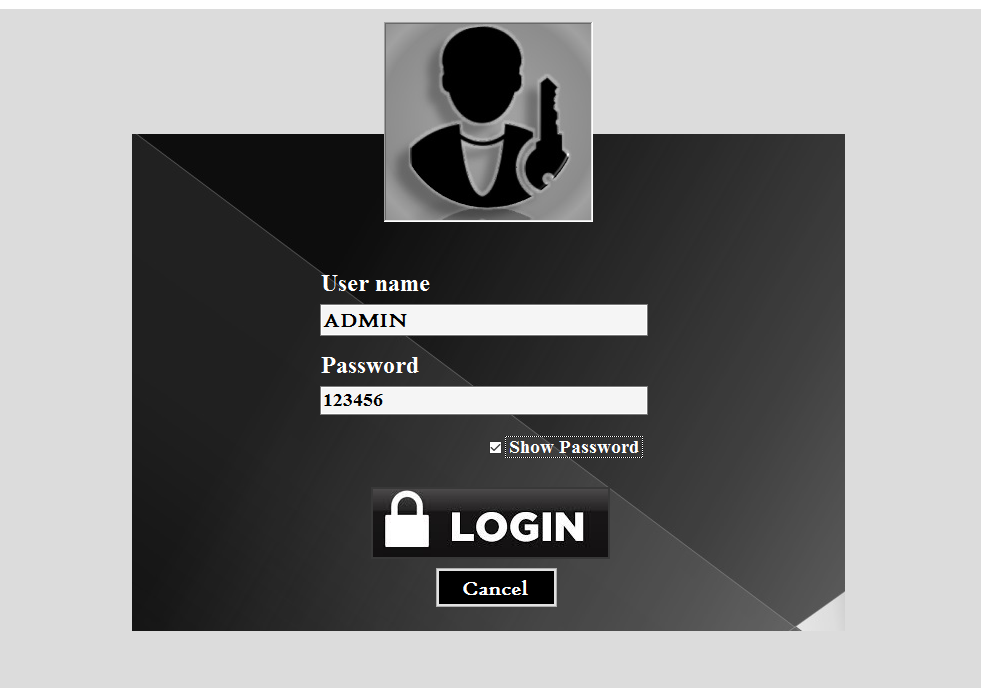
Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free.

Software testing has 4 different stages of the software development lifecycle where testing is conducted. The 4 different stages are

1. Unit testing: A level of the software testing process where individual units of software are tested. The purpose is to validate that each unit of the software performs as designed.
2. Integration Testing: A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.
3. System Testing: A level of the software testing process where a complete, integrated system is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements.
4. Acceptance Testing: (As we have not made this project based on any user’s ideology and just assumed the necessary requirements needed to make ‘**Student Information System**’, so we have skipped this testing)

**4.1. Unit Testing**

**4.1.1 Test for login screen**

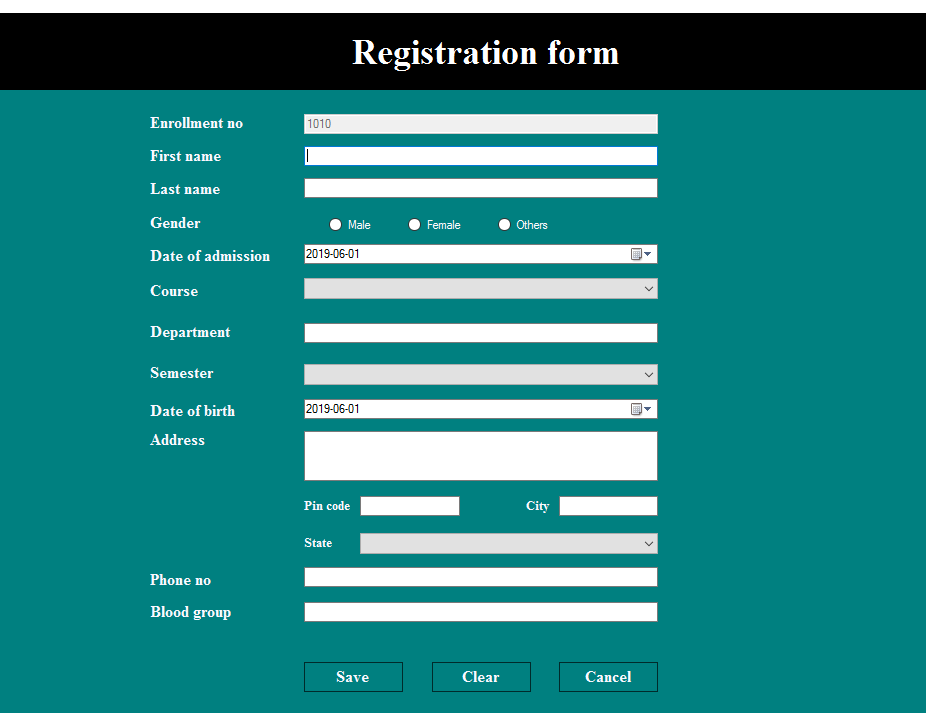
****

**Fig. 4.1.1 Login Form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test Value** | **Expected Results** | **Test Result** |
| 1. | To Check if the  ‘Login’ button works  properly | Click ‘Login’  Button | Username: Admin;  Password: 123456 | Incorrect  Username or  password | Pass |
| 2. | Username: admin;  Password: 123456 | Incorrect  Username or  password | Pass |
| 3. | Username: admin;  Password: 123 | Incorrect  Username or  password | Pass |
| 4. | Username: ADMIN;  Password: 123 | Incorrect  Username or  password | Pass |
| 5. | Username: ADMIN;  Password: 123456 | Enter homepage | Pass |
| 6. | To check if the checkbox  shows text on password  field upon checking it | Check ‘Show password’  while entering text in  password field | Password: 123456 | Password: ‘123456’  visible | Pass |
| 7. | To check if the checkbox  hides text on password  field upon unchecking it | Uncheck ‘Show password’  while entering text in  password field | Password: 123456 | Password  hidden | Pass |
| 8. | To Check if ‘Cancel’  Button works | Click ‘Cancel’ button | Click ‘Cancel’ | Software close | Pass |

**Table 4.1.1 Login form testing**

Login page has been successfully tested and has passed all the tests performed. The actual username and password was set to ‘ADMIN’ and ‘123456’ respectively. On putting these values in their respective fields we enter the home page which indicates that it is working perfectly.

**4.1.2 Test for Registration form**

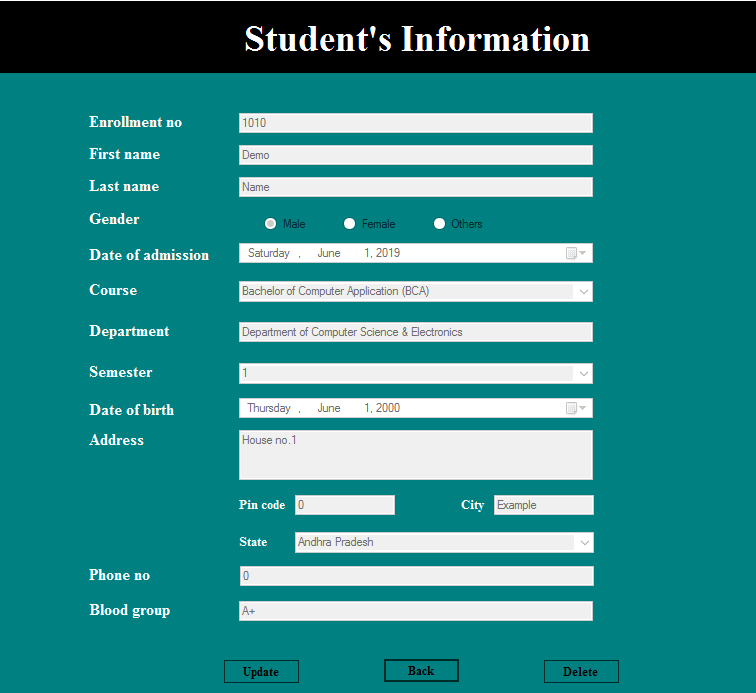
**Fig. 4.1.2 Registration form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test Value** | **Expected Results** | **Test Result** |
| 1. | To check if the  ‘Save’ button works | Enter relevant details  and click ‘Save’ button | Enter all relevant  details | Message box  “Save” | Pass |
| 2. | Leave few textbox  blank and click ‘Save’  button | Leave few  textbox blank | Message box  “Enter Records  Properly” | Pass |
| 3. | To check if the  ‘Clear’ button works | Fill up the textbox  and click ‘clear’ | Fill up textboxes  with random  values | All textboxes  cleared | Pass |
| 4. | To check if the  ‘Cancel’ button  works | Click ‘Cancel’ | Nil | Display home  page | Pass |

**Table 4.1.2 Registration form testing**

Registration form has been successfully tested and has passed all the tests performed. It gives a “Saved” message when a record is saved and stores only when every field is filled. If there’s any field that is not filled it gives an error message. The ‘Clear’ and ‘Back’ button also working as per the requirement

**4.1.3 Test for Search form**

****

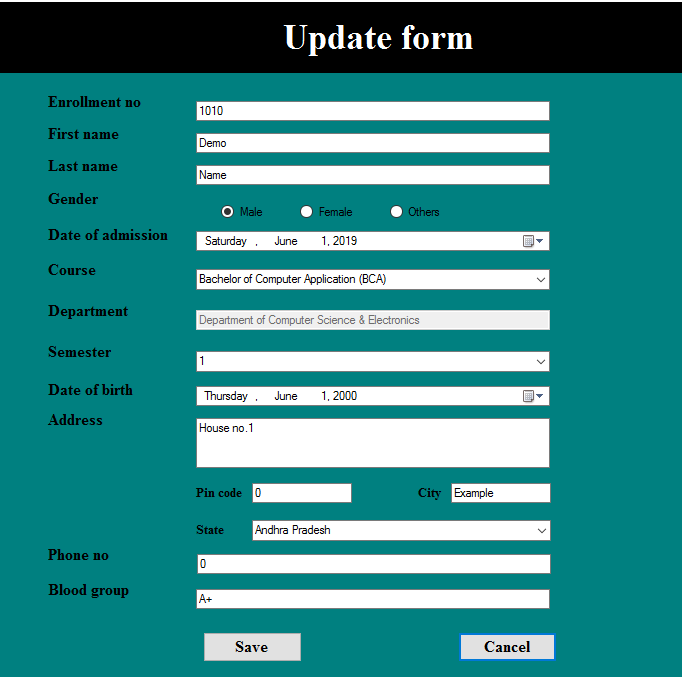
**Fig. 4.1.3 Search form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check if ‘Update’**  **button works** | **Click ‘Update’ button**  **and enter a value** | **1001** | **‘Update form’ is displayed**  **and Search form is hidden** | **Pass** |
| **2.** | **To check if ‘Back’**  **Button works** | **Click ‘Back’ button** | **Nil** | **Home page is**  **displayed** | **Pass** |
| **3.** | **To check**  **‘Delete’ button** | **Click ‘Delete’ button** | **Nil** | **Message “Record is**  **Deleted successfully”** | **Pass** |

**Table 4.1.3 Search form testing**

Search form has 3 buttons where it all three working as per requirement and hence has passed all the tests.

**4.1.4 Update form**



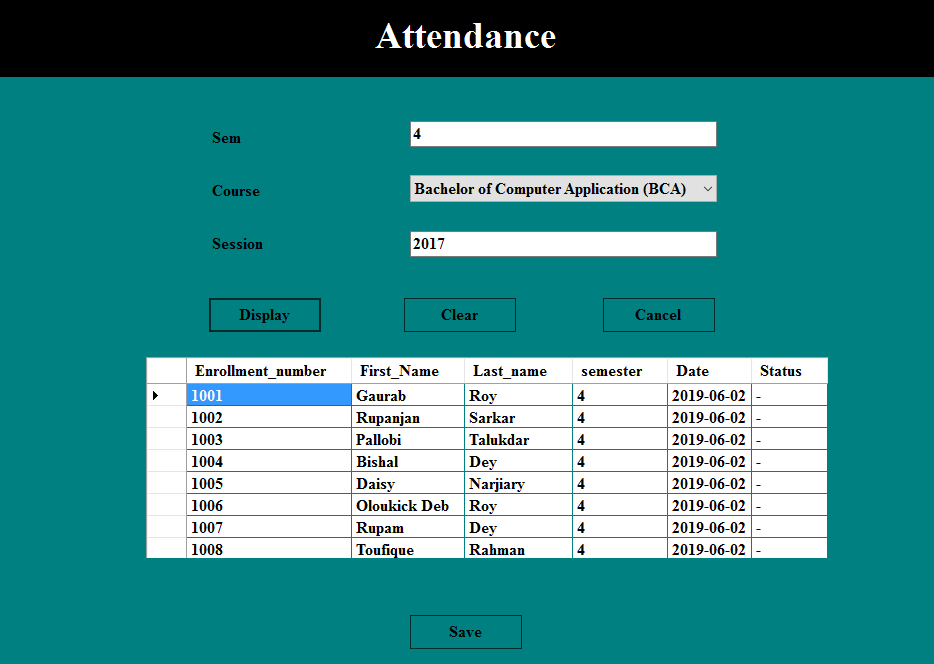
**Fig. 4.1.4 Update form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check ‘Save’ button** | **Change any values**  **and click ‘save’** | **Semester=2** | **Message box**  **‘Record updated**  **successfully’** | **Pass** |
| **2.** | **Leave any field empty** | **Nil** | **Message box ‘Field**  **cannot be empty’** | **Pass** |
| **3.** | **To check ‘Cancel’ button** | **Click ‘Cancel’** | **Nil** | **Display ‘Home’ page** | **Pass** |

**Table 4.1.4 Update form testing**

From the above tests performed we can say that the update form is working as per the requirements and has passed all the tests performed. On updating data it gives a message ‘Record updated successfully’ indicating record has been updated.

**4.1.5 Attendance form**

****

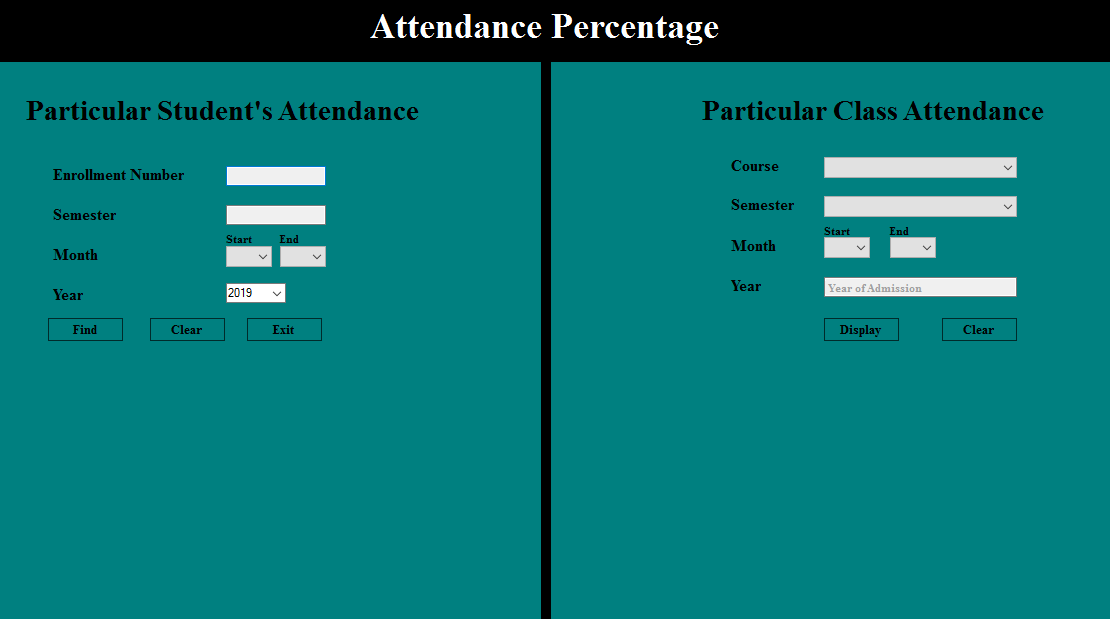
**Fig. 4.1.5 Attendance form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check ‘Display’**  **button** | **Enter appropriate**  **values** | **Refer fig. 4.3** | **Datagridview is displayed**  **with appropriate results** | **Pass** |
| **2.** | **To check ‘Clear’**  **button** | **Click ‘Clear’ button** | **Nil** | **All fields empty and**  **datagridview is hidden** | **Pass** |
| **3.** | **To check ‘Save’**  **button** | **Click ‘Save’ button** | **State ‘P’ for**  **present and ‘A’**  **for absent under**  **status** | **Datagridview is displayed**  **with appropriate results** | **pass** |
| **4.** | **To check ‘Cancel’**  **button** | **Click ‘Cancel’**  **button** | **Nil** | **Display ‘Home’ page** | **Pass** |

**Table 4.1.5 Attendance form testing**

Attendance form has passed all the tests performed and hence we can say that it is working as per the requirements.

**4.1.6 Attendance percentage form**

****

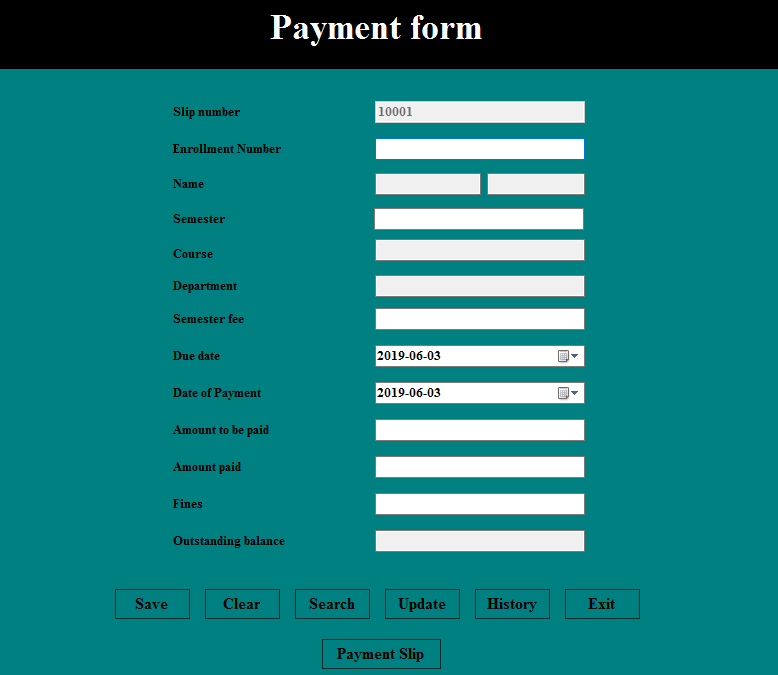
**Fig. 4.1.6 Attendance percentage form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check**  **‘Find’ button** | **Fill with appropriate**  **details and click**  **‘Find’** | **Fill values that**  **are present in**  **the database** | **Displays**  **1.No of days present**  **2.No of days absent**  **3.Total no of days**  **4.Attendance percentage** | **Pass** |
| **2.** | **Fill with details and**  **click ‘Find’** | **Fill values that**  **are not stored** | **Displays ‘0’ in every**  **field** | **Pass** |
| **3.** | **To check**  **‘Clear’ button** | **Click ‘Clear’ button** | **Nil** | **All textboxes field**  **cleared** | **Pass** |
| **4.** | **To check**  **‘Exit’ button** | **Click ‘Exit’ button** | **Nil** | **‘Home’ page is**  **displayed** | **Pass** |
| **5.** | **To check**  **‘Display’ button** | **Fill with appropriate**  **details and click**  **‘Display’** | **Fill values that**  **are present in**  **the database** | **Displays attendance**  **percentage of all the**  **students in the particular**  **class** | **Pass** |
| **6.** | **Fill with details and**  **click ‘Find’** | **Fill values that**  **are not stored** | **Datagridview value is**  **null** | **Pass** |
| **7.** | **To check**  **‘Clear’ button** | **Click ‘Clear’ button** | **Nil** | **All textboxes field**  **Cleared** | **Pass** |

**Table 4.1.6 Attendance percentage form testing**

All the tests performed in the attendance percentage form has been successfully executed and therefore working as per the requirement.

**4.1.7 Payment form**



**Fig. 4.1.7 Payment form**

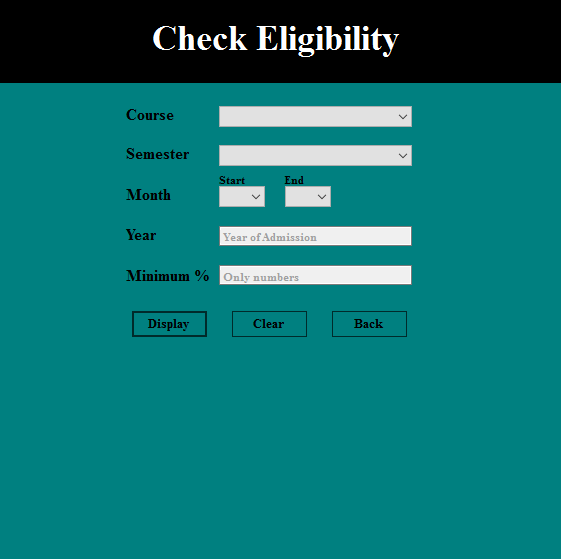
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check ‘Save’**  **button** | **Fill details and**  **click ‘Save’ button** | **Fill fields with**  **appropriate**  **values that are**  **stored** | **Saved in the**  **database with**  **‘Saved’ message** | **Pass** |
| **2.** | **Fill details and**  **click ‘Save’ button** | **Fill fields**  **with appropriate**  **values and skip a**  **field** | **‘Enter records**  **properly’ message**  **box appears** | **Pass** |
| **3.** | **To check**  **‘Clear’ button** | **Click ‘Clear’**  **button** | **Nil** | **All textboxes field**  **Cleared** | **Pass** |
| **4.** | **To check**  **‘Search’ button** | **Click ‘Search’**  **button** | **Enter ‘10001’ as**  **Slip number** | **Display all records**  **from slip no. 10001** | **Pass** |
| **5.** | **Click ‘Search’**  **button** | **Enter ‘101’ as**  **Slip number** | **Displays ‘No record**  **found’ message** | **Pass** |
| **6.** | **To check**  **‘Update’ button** | **Click ‘Update’**  **button** | **Change ‘Fines’**  **value for slip no**  **10001** | **‘Record updated**  **successfully’**  **message box appears** | **Pass** |
| **7.** | **To check**  **‘History’ button** | **Click ‘History’**  **button** | **Enter ‘1001’ as**  **Enrollment No.** | **Payment history**  **page is displayed** | **Pass** |
| **8.** | **To check**  **‘Exit’ button** | **Click ‘Exit’**  **button** | **Nil** | **Home page is**  **displayed** | **Pass** |
| **9.** | **To check**  **‘Payment slip’**  **button** | **Click ‘Payment**  **slip’ button** | **Enter ‘10001’ as**  **Slip number** | **Payment slip is**  **displayed** | **Pass** |

**Testing 4.1.7 Payment form testing**

Note: Values used here are stored in the database except for 101.

Payment form has passed all the tests performed above and hence it is working as per the requirement.

**4.1.8 Check Eligibility form**



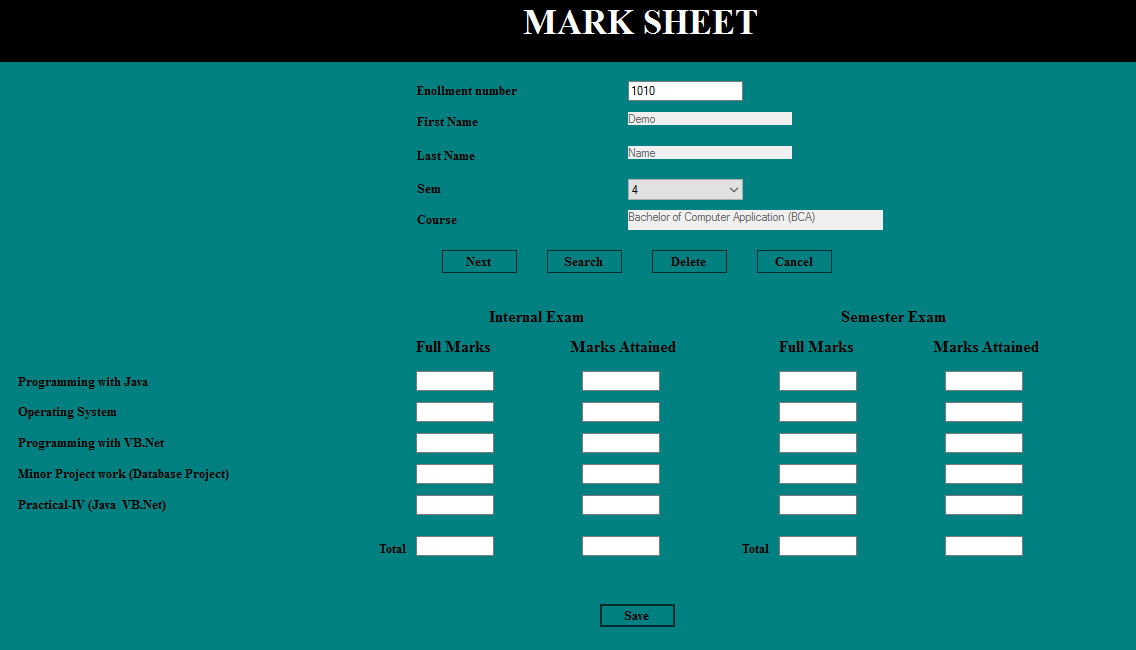
**Fig. 4.1.8 Check Eligibility form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check**  **‘Display’ button** | **Click ‘Display’**  **button** | **Fill with**  **appropriate values**  **that is present in the**  **database** | **List of students is**  **displayed that** | **Pass** |
| **2.** | **Click ‘Display’**  **button** | **Fill with**  **appropriate values**  **that is not present in**  **the database** | **‘No records found’**  **message box is**  **displayed** | **Pass** |
| **3.** | **Click ‘Display’**  **button** | **Nil** | **‘Insufficient**  **details’ message**  **box is displayed** | **Pass** |
| **4.** | **To check**  **‘Clear’ button** | **Click ‘Clear’**  **button** | **Nil** | **All textboxes field**  **Cleared** | **Pass** |
| **5.** | **To check**  **‘Back’ button** | **Click ‘Back’**  **button** | **Nil** | **Home page is**  **displayed** | **Pass** |

**Table 4.1.8 Check Eligibility form testing**

Eligibility check form works as per the requirement as it has cleared all the above tests.

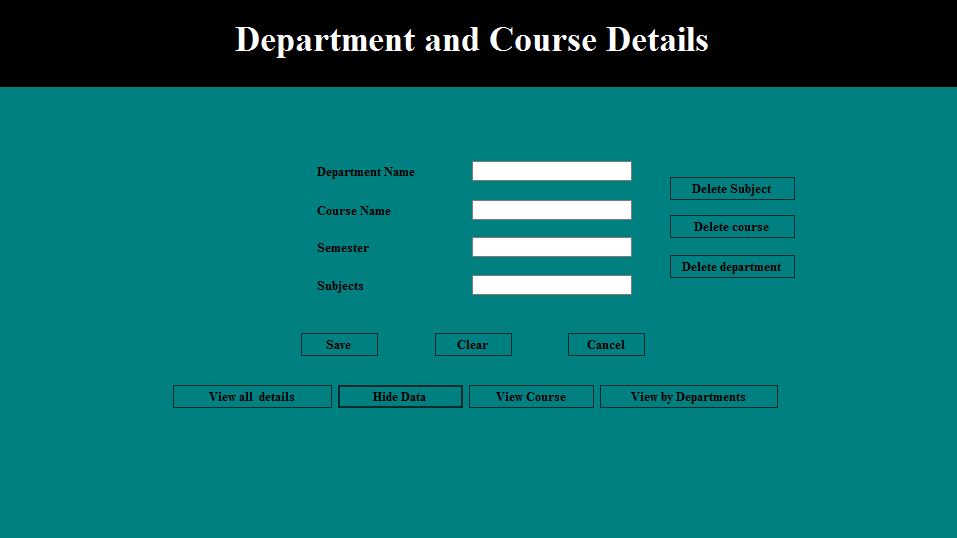
**4.1.9 Mark Sheet form**



**Fig. 4.1.9 Mark sheet form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check**  **‘Next’ button** | **Enter the enroll. No,**  **sem and click ‘next’** | **Enroll-1010,**  **sem-1** | **Subjects from the**  **particular semester**  **and course are**  **displayed** | **Pass** |
| **2.** | **To check**  **‘Save’ button** | **Enter values for all**  **the fields and click**  **‘Save’** | **Enter marks in**  **all the fields** | **‘Saved’ message box**  **is displayed** | **Pass** |
| **3.** | **Enter values and**  **click ‘Save’** | **Enter marks**  **leaving out few**  **fields** | **‘Enter records**  **properly’ is displayed** | **Pass** |
| **4.** | **To check**  **‘Search’ button** | **Click ‘Search’** | **Enroll-1010,**  **sem-1** | **Report is displayed** | **Pass** |
| **5.** | **To check**  **‘Delete’ button** | **Click ‘Delete’** | **Enroll-1010,**  **sem-1** | **‘Record deleted’**  **message box is**  **displayed** | **Pass** |
| **6.** | **To check**  **‘Cancel’ button** | **Click ‘Cancel’** | **Nil** | **Home page is**  **displayed** | **Pass** |

**Table 4.1.9 Mark sheet form testing**

Form **Mark Sheet** is working as per the requirement and has passed all the tests performed above. It displays the subject based on Course and semester and saves the marks in the database. It also searches the mark sheet result based on enrollment number and semester. 

**4.1.10 Department and Course Details form**

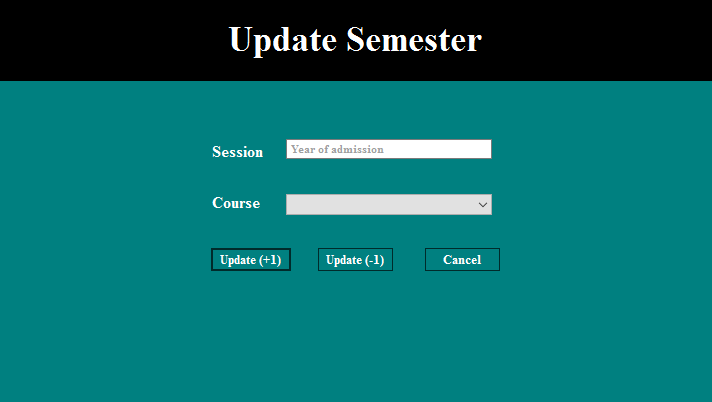
**Fig. 4.1.10 Department and Course form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check**  **‘Save’ button** | **Click ‘Save’** | **Nil** | **“Field empty!!! Try**  **again” message box** | **Pass** |
| **2.** | **Click ‘Save’** | **Enter appropriate**  **values** | **“Saved” message box** | **Pass** |
| **3.** | **To check**  **‘Clear’ button** | **Click ‘Clear’** | **Nil** | **All textboxes field**  **Cleared** | **Pass** |
| **4.** | **To check**  **‘Cancel’ button** | **Click ‘Cancel’** | **Nil** | **Home page is**  **displayed** | **Pass** |
| **5.** | **To check ‘View**  **all details’**  **button** | **Click ‘View**  **all details’** | **Nil** | **All details are**  **displayed in**  **datagridview** | **Pass** |
| **6.** | **To check ‘Hide**  **data’ button** | **Click ‘Hide data’** | **Nil** | **All details displayed**  **in datagridview are**  **hidden** | **Pass** |
| **7.** | **To check ‘View**  **course’ button** | **Click ‘View**  **course’ button** | **Nil** | **All courses are**  **displayed in**  **datagridview** | **Pass** |
| **8.** | **To check ‘View**  **departments’**  **button** | **Click ‘View**  **departments’**  **button** | **Nil** | **All departments are**  **displayed in**  **datagridview** | **Pass** |
| **9.** | **To check ‘Delete**  **Subject’ button** | **Click ‘Delete**  **subject’ button** | **Nil** | **‘Field empty’**  **message box appears** | **Pass** |
| **10.** | **Enter value that is**  **stored in subject**  **field** | **‘Deleted Successfully’**  **message box appears** | **Pass** |
| **11.** | **To check ‘Delete**  **Course’ button** | **Click ‘Delete**  **course’ button** | **Nil** | **‘Field empty’**  **message box appears** | **Pass** |
| **12.** | **Enter value that is**  **stored in course**  **field** | **‘Deleted Successfully’**  **message box appears** | **Pass** |
| **13.** | **To check ‘Delete**  **Department’**  **button** | **Click ‘Delete**  **Department’**  **button** | **Nil** | **‘Field empty’**  **message box appears** | **Pass** |
| **14.** | **Enter value that is**  **stored in dept.**  **field** | **‘Deleted Successfully’**  **message box appears** | **Pass** |

**Table 4.1.10 Department and Course form testing**

‘Department and Course’ form has passed all the tests performed. Results have been verified by reviewing the database and all the values that have been entered were stored in the database. With all the results coming out positive, we can say that this form is working as per the requirement.

**4.1.11 Update Semester form**



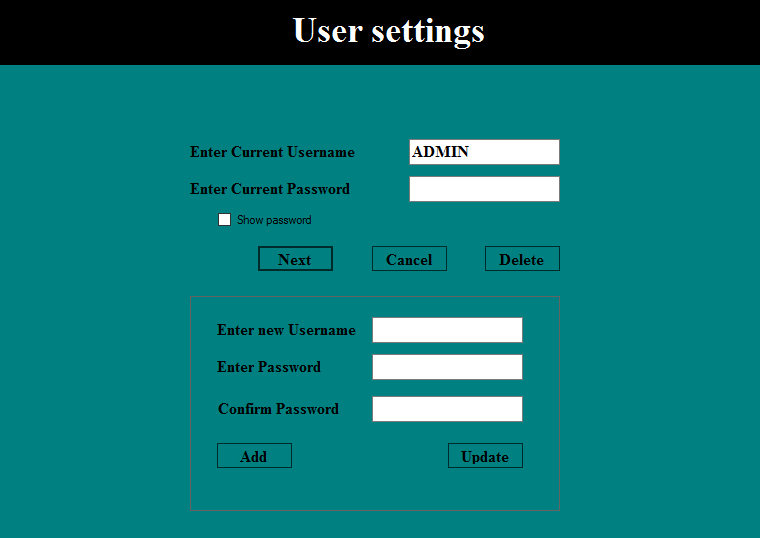
**Fig. 4.1.11 Update Semester form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test value** | **Expected Results** | **Test Result** |
| **1.** | **To check**  **‘Update(+1)’**  **button** | **Click**  **‘Update(+1)’**  **button** | **Nil** | **“Fields empty!!!”**  **message** | **Pass** |
| **2.** | **Session = 2017**  **Course = BCA** | **“Records updated**  **successfully”**  **message** | **Pass** |
| **3.** | **To check**  **‘Update(-1)’**  **button** | **Click**  **‘Update(-1)’**  **button** | **Nil** | **“Fields empty!!!”**  **message** | **Pass** |
| **4.** | **Session = 2017**  **Course = BCA** | **“Records updated**  **successfully”**  **message** | **Pass** |
| **5.** | **To check**  **‘Cancel’ button** | **Click ‘Cancel’** | **Nil** | **Home page is**  **displayed** | **Pass** |

**Table 4.1.11 Update Semester form testing**

Update semester has passed all the tests performed and is working as per the requirement. After filling up the conditions and clicking update button will help user to increment their semester by one with ease.

**4.1.12 Mark Sheet form**



**Fig. 4.1.12 User settings form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test Value** | **Expected Results** | **Test Result** |
| 1. | To Check if the  ‘Next’ button works  properly | Click ‘Next’ button | Username: Admin;  Password: 123456 | Incorrect  Username or  password | Pass |
| 2. | Username: admin;  Password: 123456 | Incorrect  Username or  password | Pass |
| 3. | Username: admin;  Password: 123 | Incorrect  Username or  password | Pass |
| 4. | Username: ADMIN;  Password: 123 | Incorrect  Username or  password | Pass |
| 5. | Username: ADMIN;  Password: 123456 | Display Panel1  and ‘Delete’ button | Pass |
| 6. | To check if the checkbox  shows text on password  field upon checking it | Check ‘Show password’  while entering text in  password field | Password: 123456 | Password:  ‘123456’  visible | Pass |
| 7. | To Check if ‘Cancel’  Button works | Click ‘Cancel’ button | Nil | ‘Home’ page  is displayed | Pass |
| 8. | To check if ‘Delete’  button works | Click ‘Delete’ button | Username: admin | ‘Incorrect  Username or  password’ | Pass |
| 9. | Username: ADMIN | ‘User deleted’ | Pass |
| 10. | To check if ‘Add’  button works | Click ‘Add’ button | Enter new username  & different password | ‘Password do not  match’ | Pass |
| 11. | Enter new username  & same password | ‘Saved as new user’ | Pass |
| 12. | To check if ‘Update  button works | Click ‘Update’ button | Enter new username  & different password | ‘Password do not  match’ | Pass |
| 13. | Enter new username  & same password | ‘Updated current  user’ | Pass |

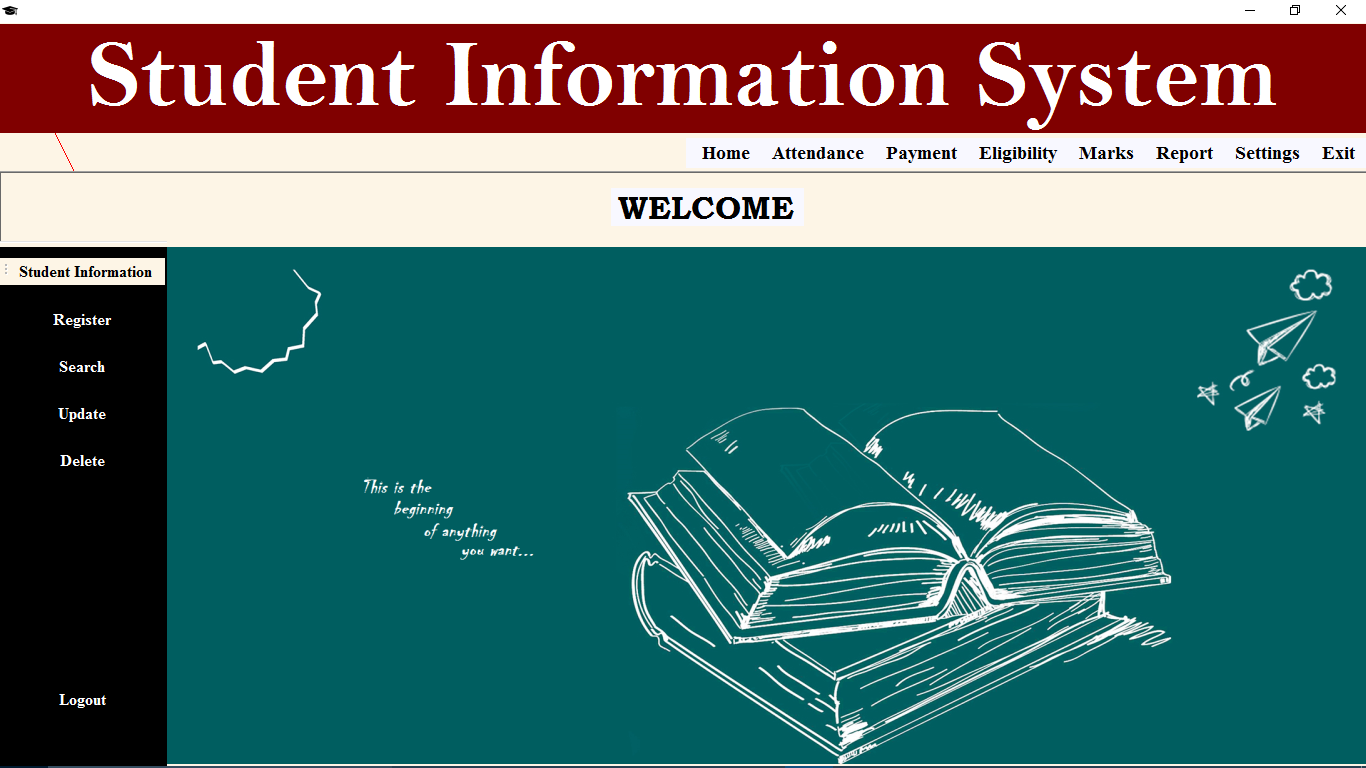
**Table 4.1.12 User settings form testing**

User settings form has passed all the tests performed above. All the buttons are working perfectly meeting all the requirements. It can add, update and delete user. Also checks if the passwords are same.

With this, we have completed our unit testing and every form has been tested individually and is ready for the integration testing.

**4.2. Integration Testing**

**4.2.1. Test for Home page**



**Fig. 4.2.1 Home page**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl.no** | **Objective** | **Steps to Perform** | **Test Value** | **Expected Results** | **Test Result** |
| **1.** | **To check for ‘Register’** | **Click ‘Register’** | **Nil** | **Registration form opens** | **Pass** |
| **2.** | **To check for ‘Search’** | **Click ‘Search’** | **Enter the enrollment number** | **Search form opens** | **Pass** |
| **3.** | **To check for ‘Update’** | **Click ‘Update’** | **Enter the enrollment number** | **Update form opens** | **Pass** |
| **4.** | **To check for ‘Delete’** | **Click ‘Delete’** | **Enter the enrollment number** | **‘Deleted successfully’** | **Pass** |
| **5.** | **To check for ‘Attendance’** | **Click ‘Daily attendance’** | **Nil** | **Daily attendance form opens** | **Pass** |
| **6.** | **Click ‘Attendance percentage’** | **Nil** | **Attendance percentage form**  **opens** | **pass** |
| **7.** | **To check for ‘Payment’** | **Click ‘Payment’** | **Nil** | **Payment form opens** | **Pass** |
| **8.** | **To check for ‘Eligibility’** | **Click ‘Eligibility** | **Nil** | **Eligibility form opens** | **Pass** |
| **9.** | **To check for ‘Marks’** | **Click ‘Marks’** | **Nil** | **Mark sheet form opens** | **Pass** |
| **10.** | **To check for ‘Report’** | **Click ‘Report’** | **Enter the Enrollment number & semester** | **Report is displayed** | **Pass** |
| **11.** | **To check for ‘Settings’** | **Click ‘Course details’** | **Nil** | **Course details form opens** | **Pass** |
| **12.** | **Click ‘Update semester’** | **Nil** | **Update semester form opens** | **Pass** |
| **13.** | **Click ‘Admin/user settings** | **Nil** | **User settings form opens** | **Pass** |
| **14.** | **To check for ‘Exit’** | **Click ‘Exit’** | **Nil** | **Software closes** | **Pass** |
| **15.** | **To check for ‘Logout’** | **Click ‘Logout’** | **Nil** | **Login form opens** | **Pass** |

**Table 4.2.1 Home page testing**

The tests performed above have been successful and thus indicates that the software has been integrated properly and therefore is ready for System testing.

**4.3. System Testing**

Here interface between two sub-systems are tested because errors exists in the interface between two sub-systems may not be visible until they are combined tested as a complete unit.

After combining the individual sub-system, the overall program has been tested for determining whether the correct individual sub-system fit together properly into a correct whole, the system.

System testing is of two types: i. White box testing ii Black box testing

Here we have performed the white box testing.

The testing has been done keeping in mind if it meets the expected requirements. While testing we found no errors and therefore completes the System testing.

**Chapter 5: Documentation**

1. Login form

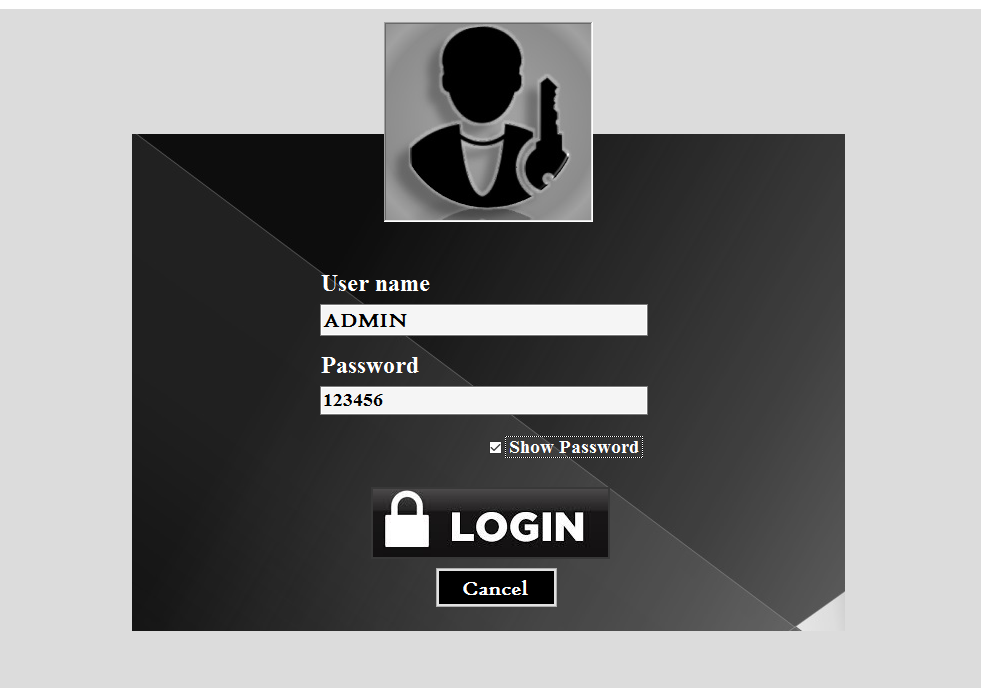
****

Fig 5.1 Login form

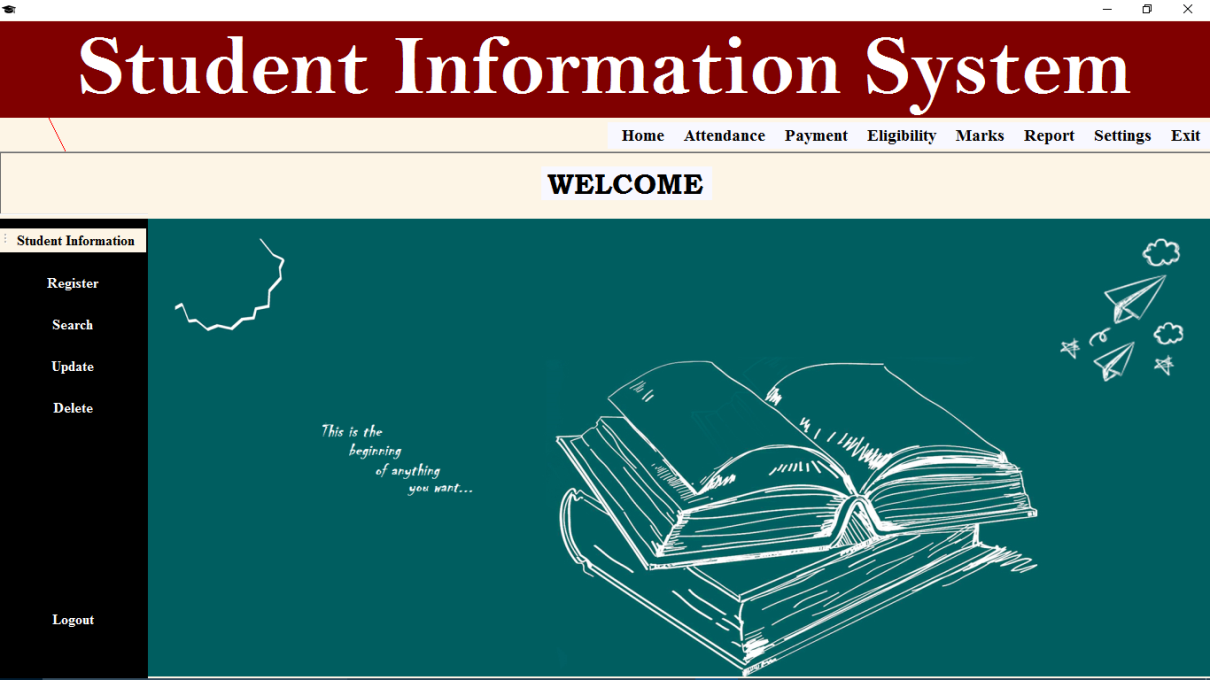
1. Home page

Fig 5.2 Home Page

1. Registration form

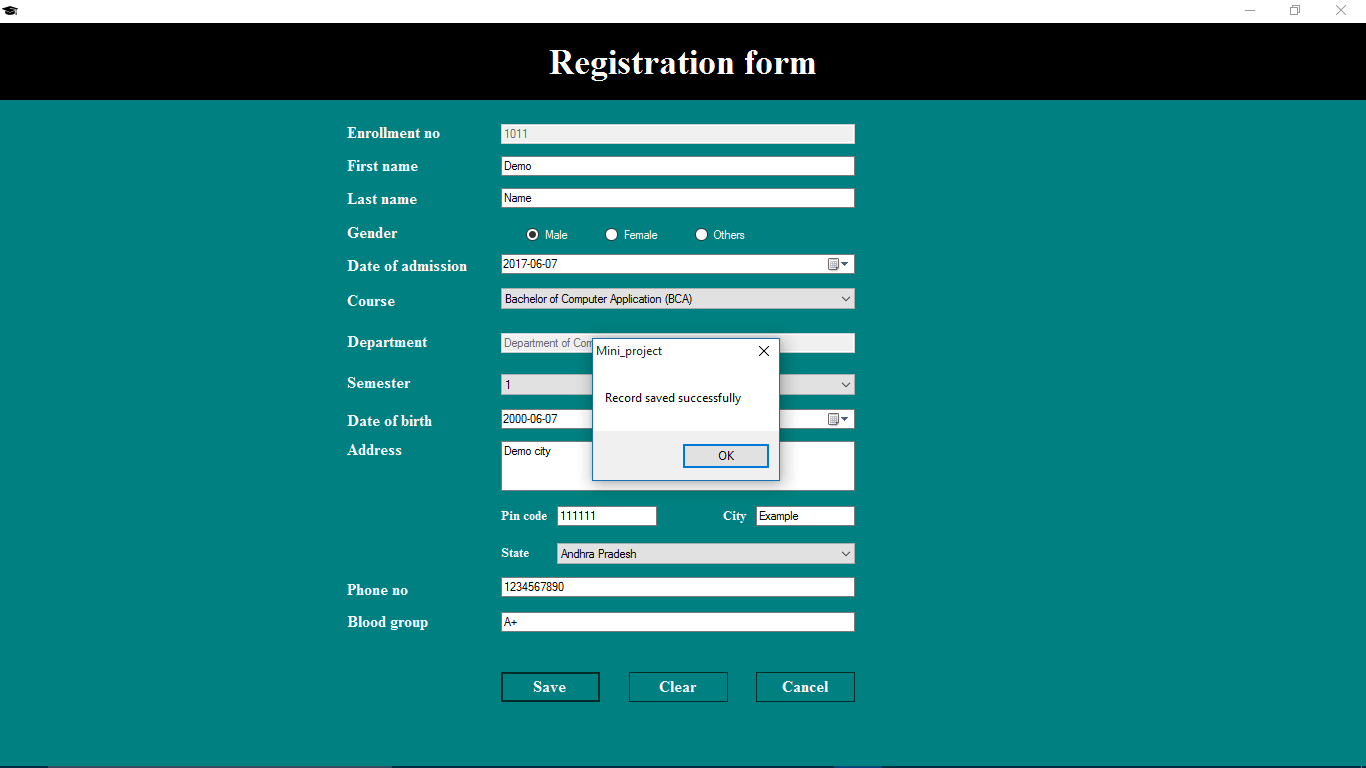


Fig 5.3 Registration form

1. Search form

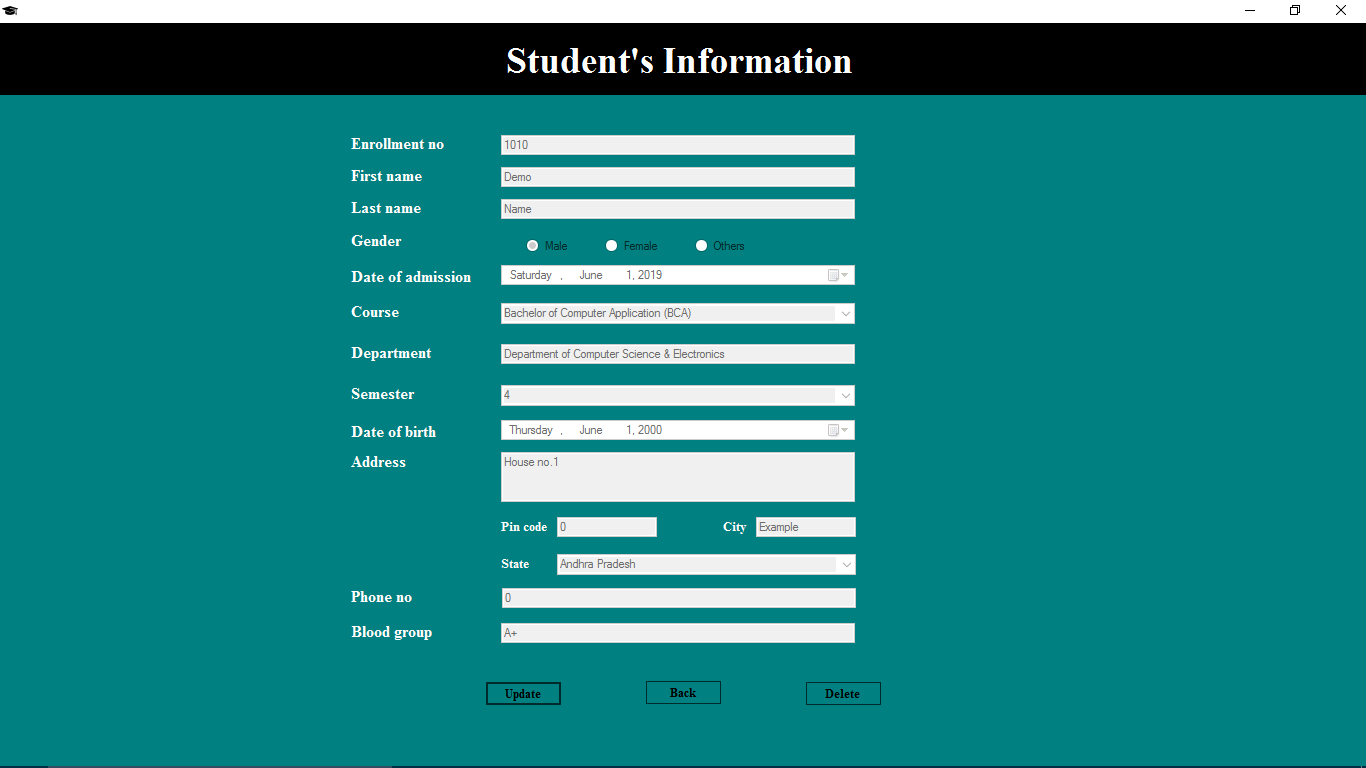


Fig 5.4 Student’s Information/search form

1. Update form

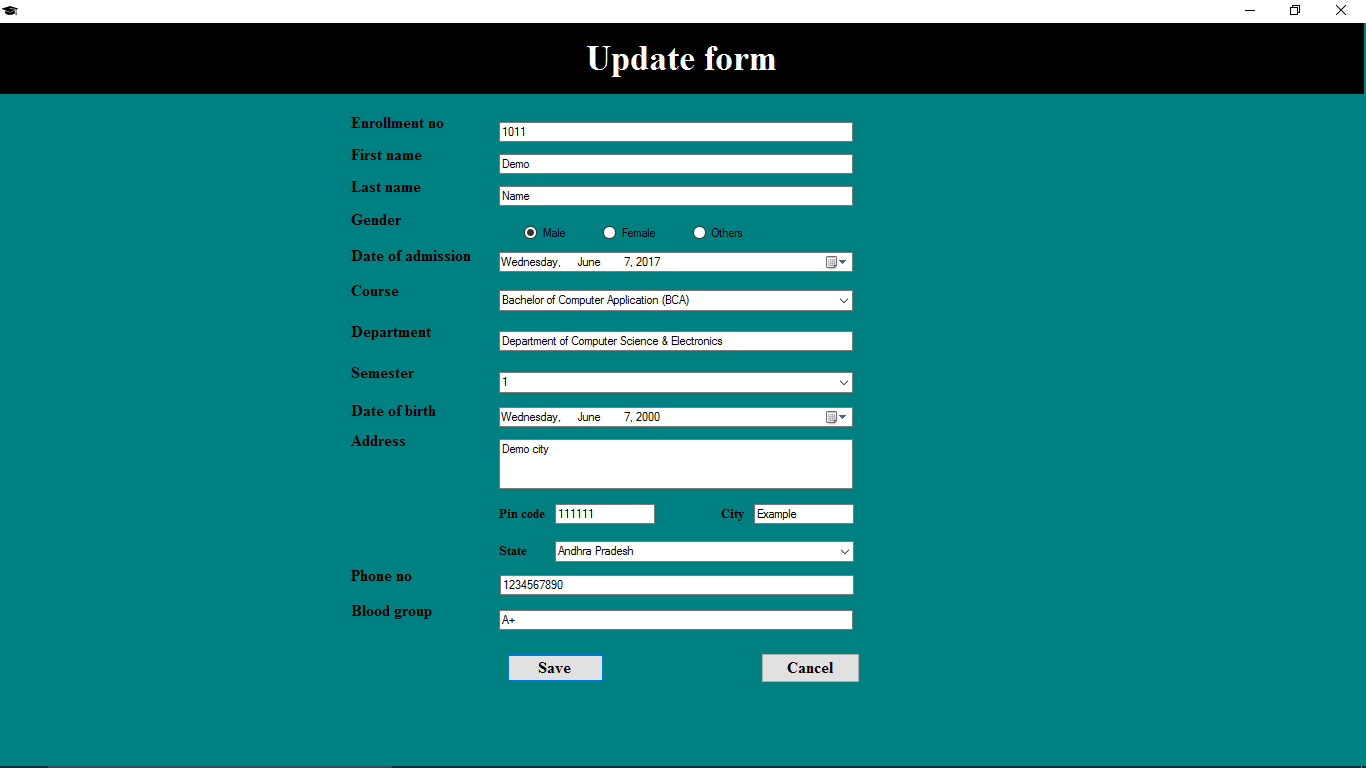


Fig 5.5 Student’s information update form

1. Attendance form

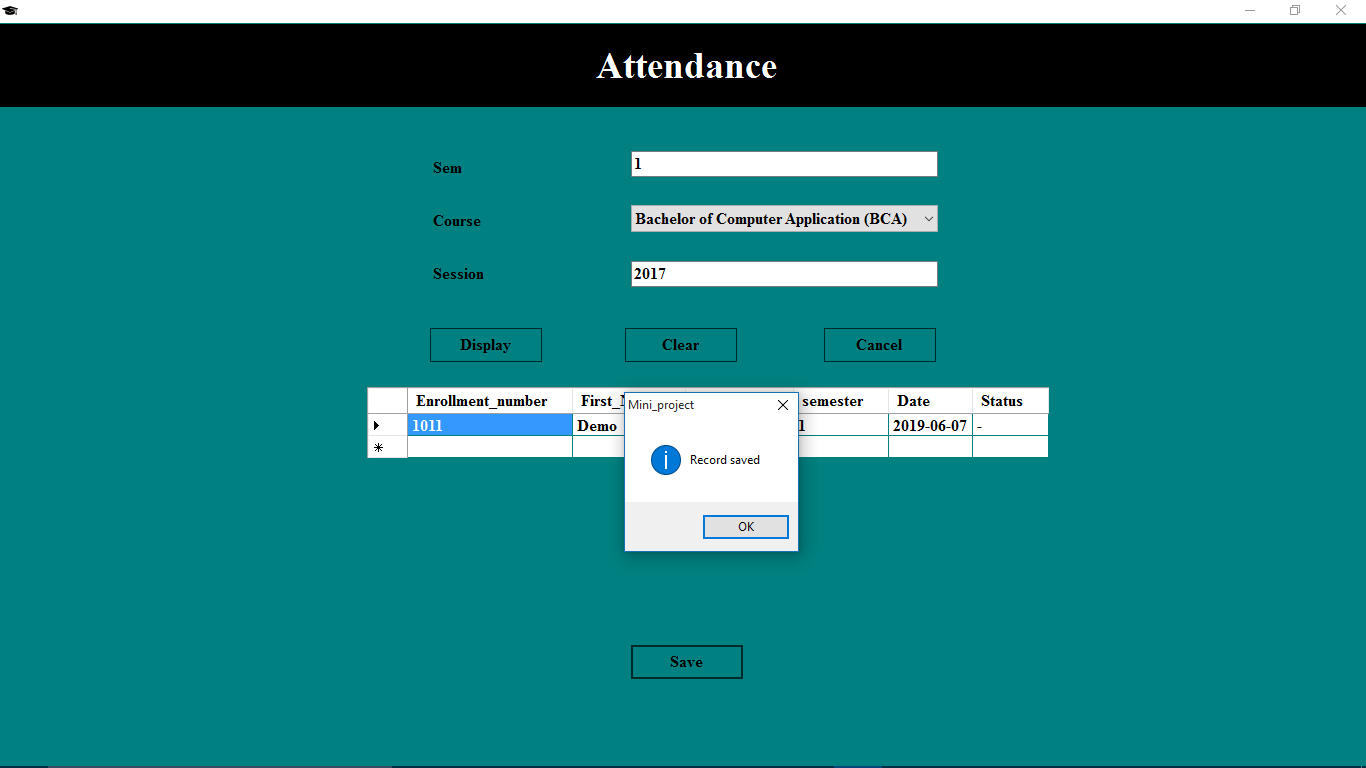


Fig 5.6 Attendance form

1. Attendance Percentage form

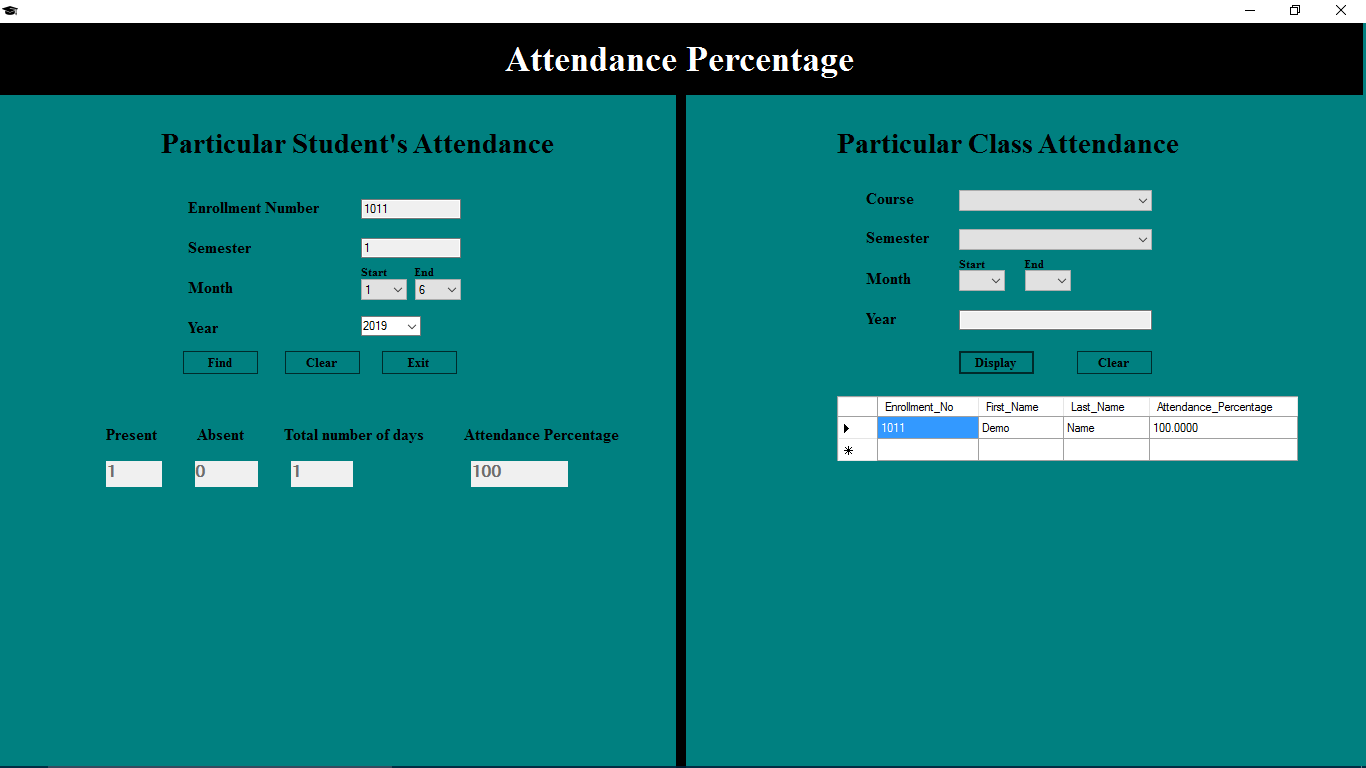


Fig 5.7 Attendance percentage form

1. Payment form

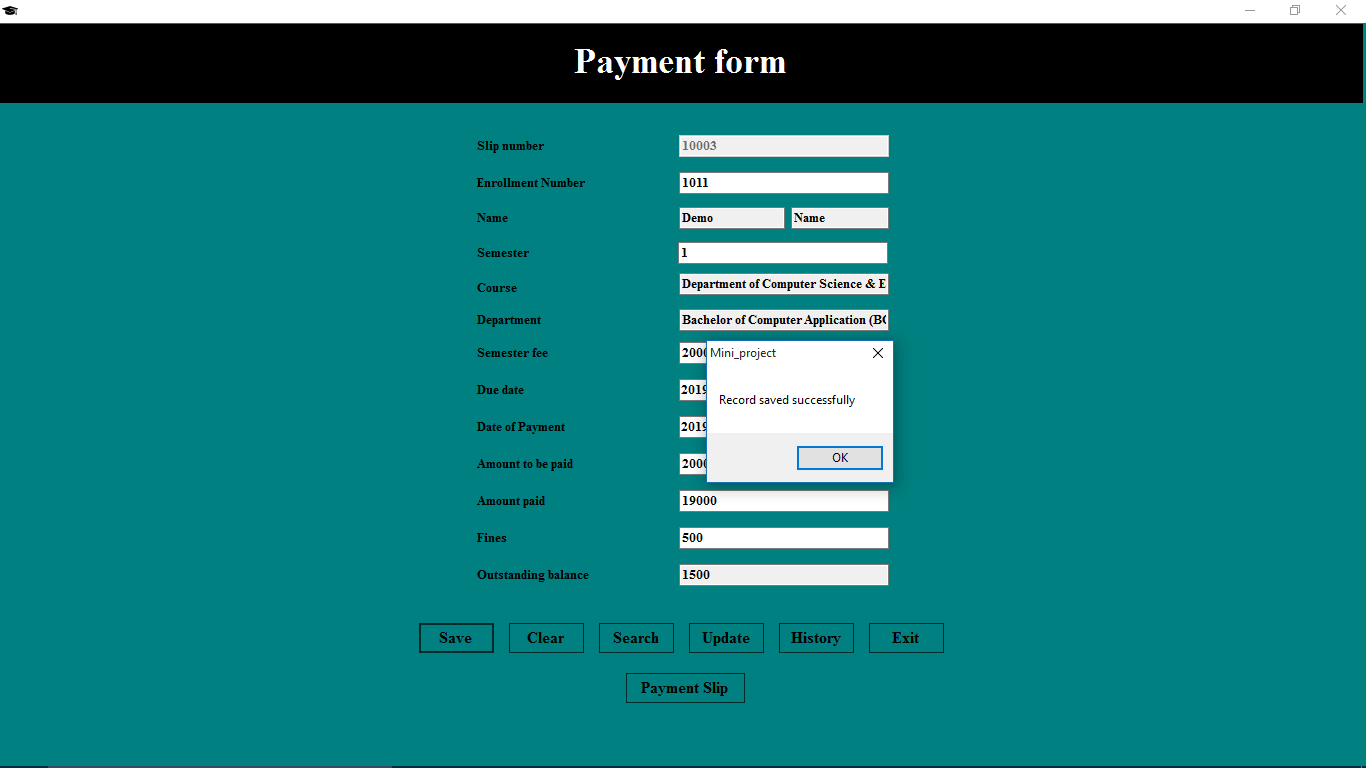


Fig 5.8 Payment form

1. Payment History form

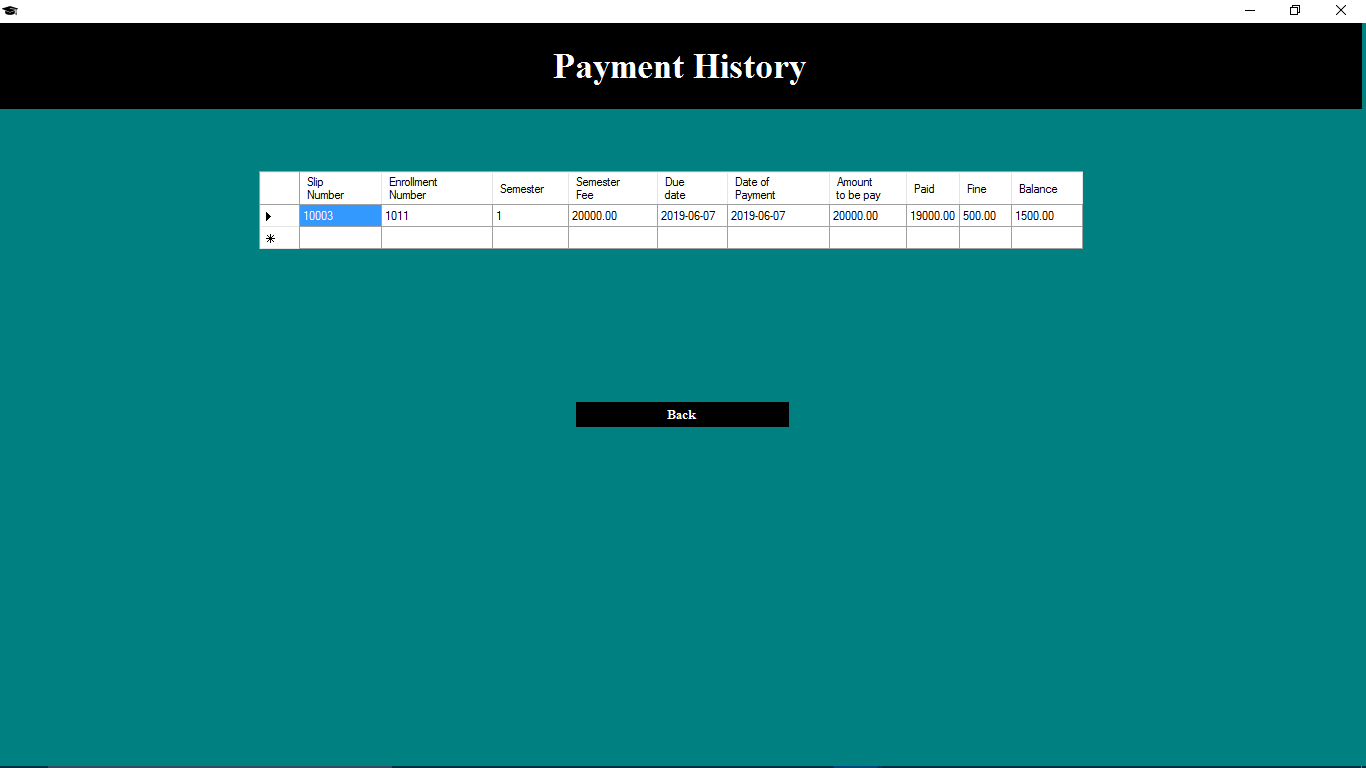


Fig 5.9 Payment History

1. Payment Slip

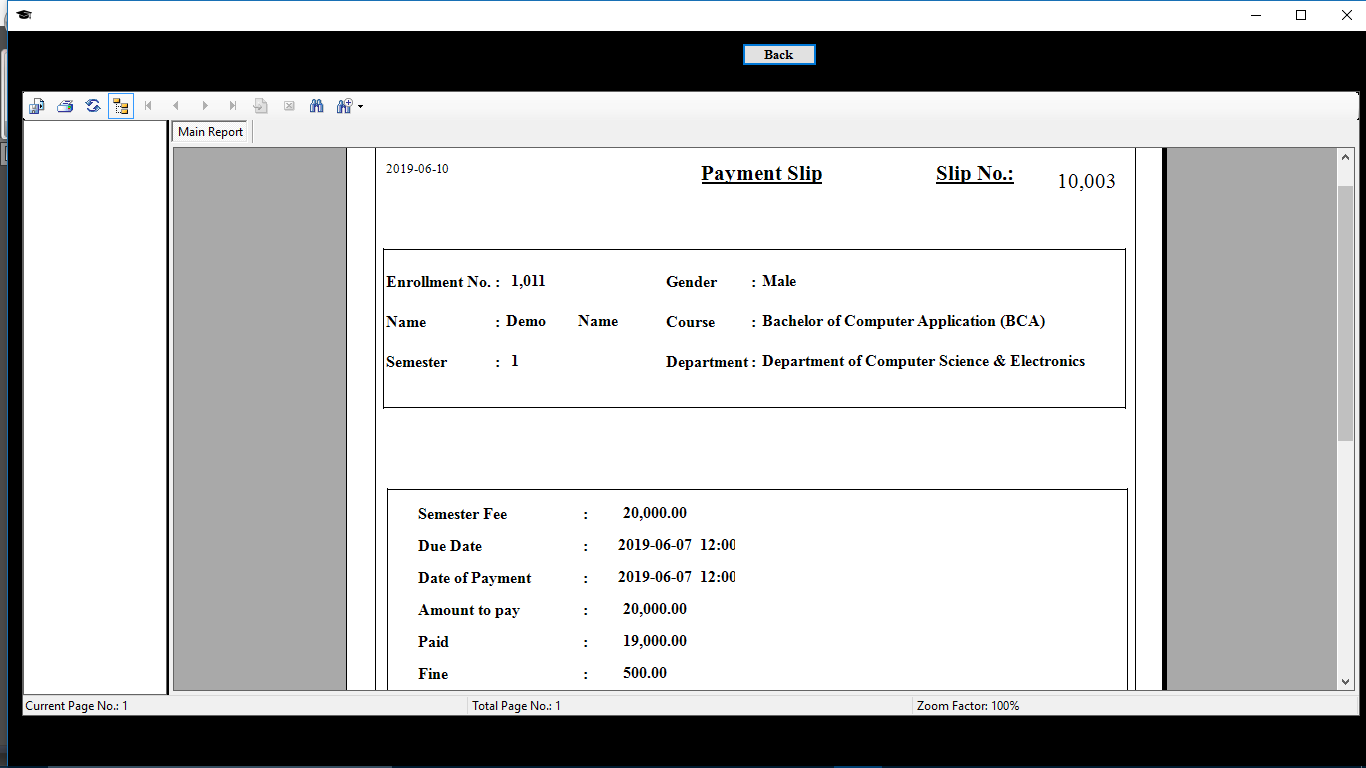


Fig 5.10 Payment Slip

1. Eligibility form

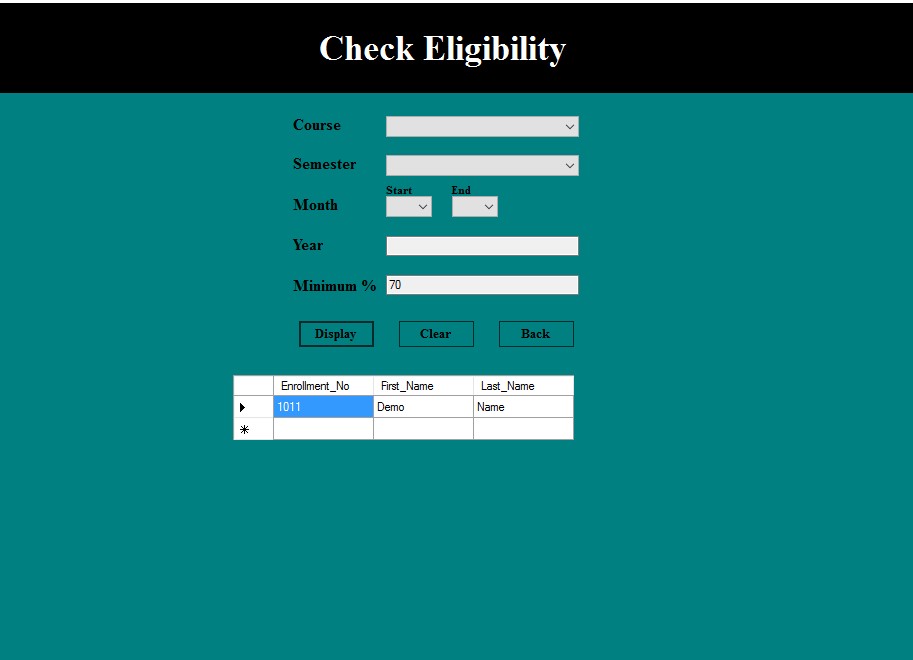


Fig 5.11 Check Eligibility form

1. Mark sheet form

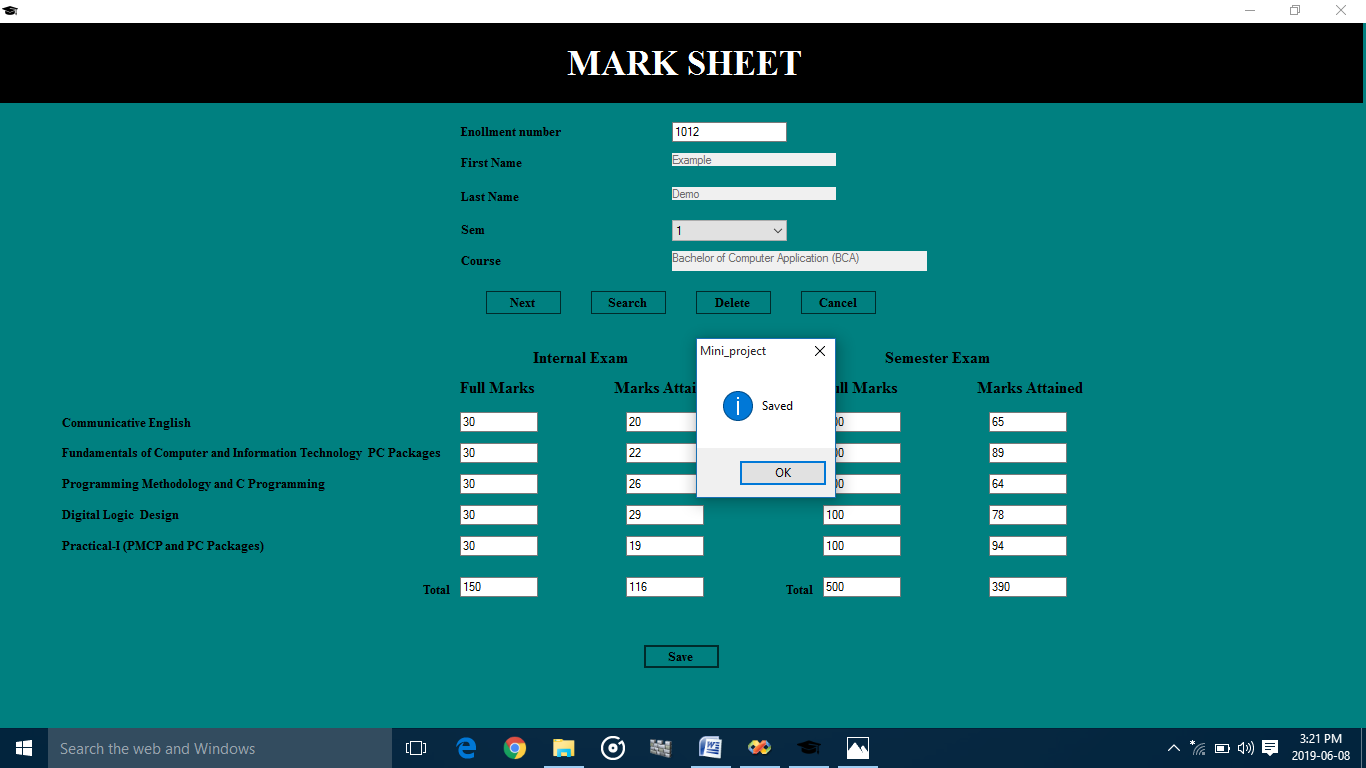


Fig 5.12 Mark sheet form

1. Mark sheet

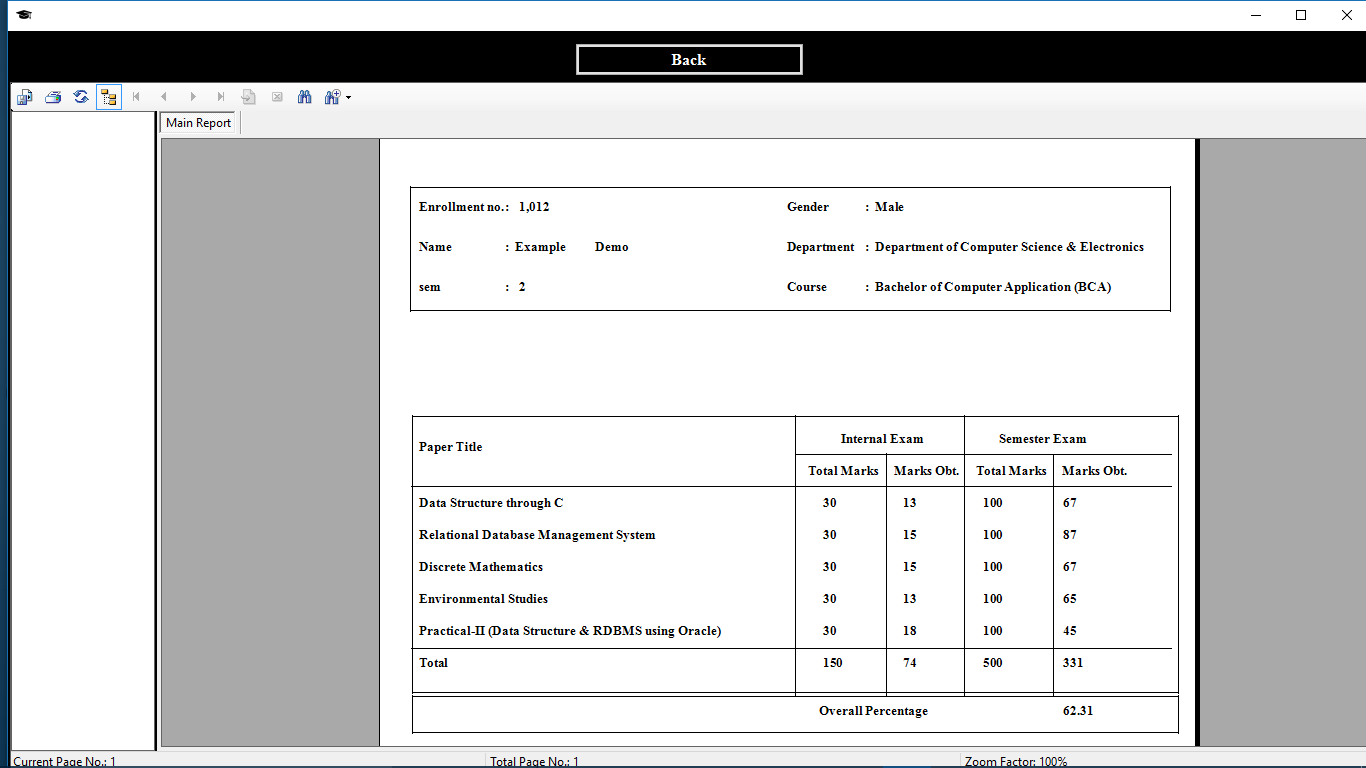


Fig 5.13 Mark sheet

1. Department & Course details

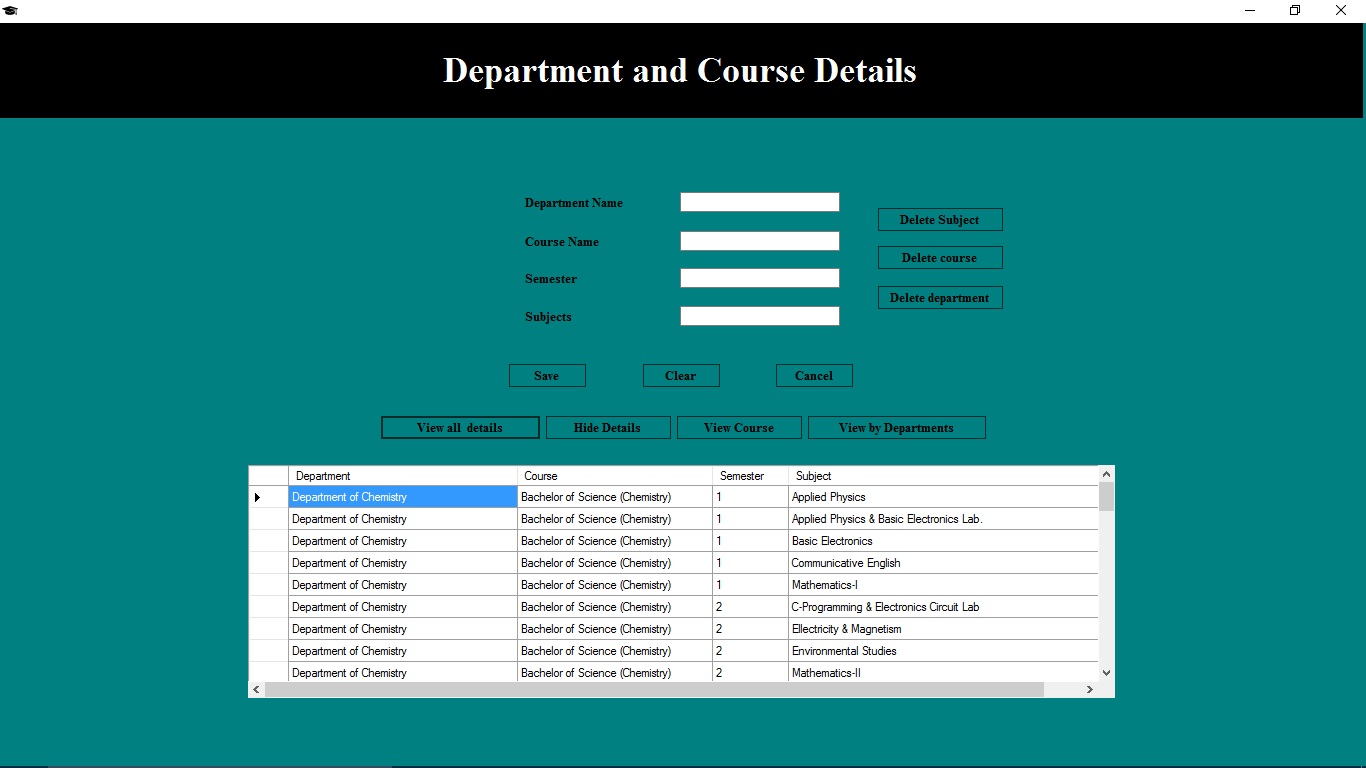


Fig 5.14 Department & Course details form

1. Department & Course details

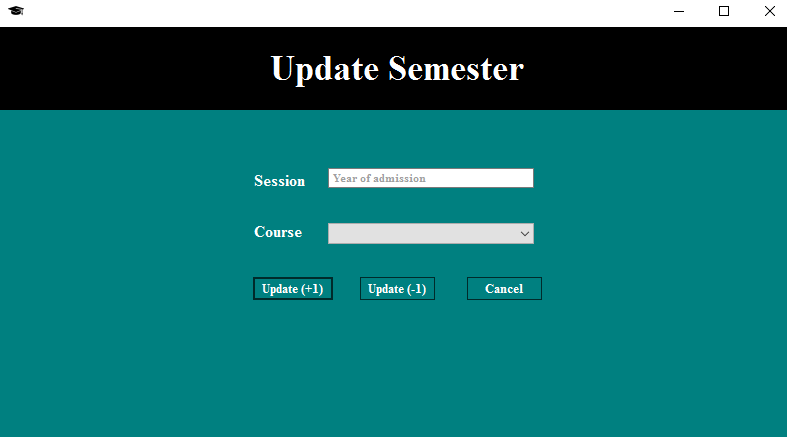


Fig 5.15 Update Semester form

1. Mark sheet form

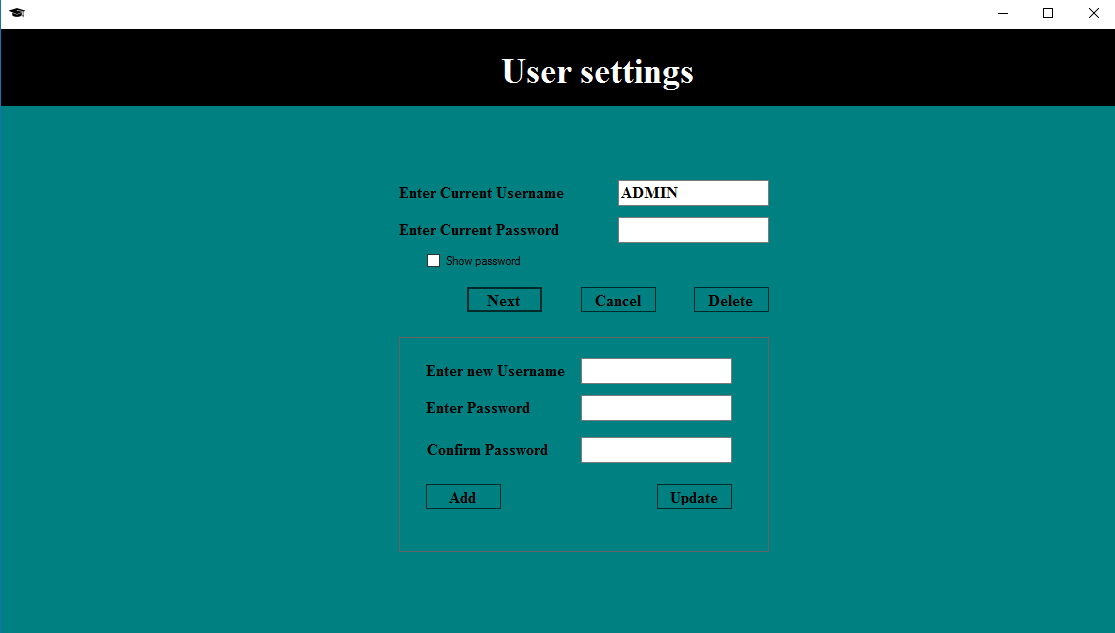


Fig 5.16 User Settings form

**Chapter 6: Future Scope and Conclusion**

**6.1 Future Scope**

* Our future work will be to provide access to few features of this software to students as well as teachers.
* Teachers can directly provide the attendance.
* Will add new features when required.

**6.2 Conclusion**

The system we have developed will be able to store student details like personal details, marks obtained by the student during exams, attendance records and everything related to the payment section.

As our system is built at administrative end, it will help the administrators to reduce the workload that exist in the manual system and minimizing the probability of losing any data. This system will change the way of viewing the data and help them to do their work in a very less amount of time.

Few of the important characteristics that this system will provide are as follows:

* Search, modify and delete information of any student based on their enrollment number.
* Check if the student is eligible or not to give the exam.
* Well designed reports.

This system will provide user friendly approach and eliminate the hard work faced by the existing (manual) system.

**Chapter 7: References/Bibliography**

MySQL: The Complete References – Vaswani Vikram

Information Practices – Sumit Arora

The definitive guide to Mysql – Kofler Michael

Web References

<https://www.scribd.com>

<https://meeraacademy.com>

<https://slideshare.com>

<https://tableplus.io>

<https://www.mysqltutorial.com>

<https://stackoverflow.com>

<https://dev.mysql.com>