A

Mini-Project Report On

ADMISSION MANAGEMENT SYSTEM

In the subject of Mini project

Submitted To

KIT College Of Engineering (Autonomous), Kolhapur

In Partial Fulfillment of the Requirement for the Degree

Of

Bachelor of Engineering

S.Y. B.Tech. CSE

Submitted by

Roll No	Name
A03	Amit Khadke
A10	Saad Attar
A13	Pranav Salokhe
A16	Sudhnwa Ghorpade

(COMPUTER SCIENCE AND ENGINEERING)

Under The Guidance Of

Ms. Ranjeeta Pandhare

Department Of Computer Science& Engineering
KIT'S College of Engineering, Kolhapur.

2021-2022

CERTIFICATE

This is to certify that Amit Khadke, Saad Attar, Pranav Salokhe, Sudhnwa Ghorpade have successfully completed the Mini-Project entitled **"ADMISSION MANAGEMENT SYSTEM",** in partial fulfilment of the requirement for the Bachelor of Engineering (Computer Science & Engineering) Of KIT's College of Engineering, Kolhapur in the S.Y. B.Tech. CSE of the academic year 2021-22.

Project Guide	H.O.D. (CSE)
Prof. Ranjeeta Pandhare	Dr. M. S. Kalas

Kolhapur Institute of
Technology's College of
Engineering, Kolhapur. Year
2021-2022

ACKNOWLEDGEMENT

We would like to express our deep gratitude to Prof. Ranjeeta Pandhare & Dr. M. S. Kalas Head of CSE for their constant encouragement and belief in us. Their guidance and attention throughout the project work has been of immense help to us. We express our sincere thanks to all the teaching and non-teaching staff and all those who have directly and indirectly helped in marketing project a success.

Sincerely by

Roll no.	Name
A03	Amit Khadke
A10	Saad Attar
A13	Pranav Salokhe
A16	Sudhnwa Ghorpade

CERTIFICATE

This is to certify that the Mini Project Report entitled **"ADMISSION MANAGEMENT SYSTEM"** submitted by Mr. Amit Khadke(A03), Mr. Saad Attar(A10), Mr. Pranav Salokhe(A13), Mr. Sudhnwa Ghorpade(A16) to KIT'S College of Engineering (Autonomous), Kolhapur, Maharashtra is a record of Bonafide project work carried out by them under my supervision and guidance.

Supervisor **Supervisor**

Date:

ABSTRACT

ADMISSION MANAGEMENT SYSTEM is a software application to manage all the activities concern to the admission. This project will help to store all admission related information. This application is useful store all the information related to the student's admission.

Today 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 Computerized Automated Student Admission System, to speed up and make it easy to use system. A poor admissions system can mean fewer students being admitted into a college because of mistakes or an overly slow response time. This project's aim is to automate the system, prechecking the inclusion of all required material and automatically ranking each student's application based on a number of criteria. It supports the current process but centralizes it and makes it possible for decisions to be made earlier and easier way.

CONTENT

CHAPTER 1

INTRODUCTION

As the number of students appearing for the counseling is increasing rapidly 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 serve 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 tries to avoid the mistakes that are usually happen during the Admission Process.

This project's aim is to automate the system, prechecking the inclusion of all required material and automatically ranking each student's application based on a number of criteria. It supports the current process but centralizes it and makes it possible for decisions to be made earlier and easier way.

Scope

- 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 with all the colleges, courses offering by them and availability of seats.
- For each event database should be updated and should be ready to provide information for next candidate in no time.

Objective

- To make better system which will make the process easier by computerized student admission management system
- Reduce the work load of universities with expected quality
- Eliminate the wastage of paper

CHAPTER 2

SYSTEM FEATURES

FUNCTIONAL REQUIRENMENT:

- **Login:**

1. Input: Enter the user id and password
2. Output: User will be able to use the features of the software

- **Applicant List**: The list of all students applied to their respective branches will be available
- **Student Addition/Updating/Deletion**: Here the task regarding the updating or deletion of aconcerned student will done.
- **Fee's details**: Here the fee of all branches will be displayed according to cast wise [OPEN/OBC/NT/ST/TFWS]
- **Important dates:** Here the important dates will be flashed regarding the admission process.

NON-FUNCTIONAL REQUIRENMENT:

Performance Requirements

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the student as soon as the request is submitted. The data of all students should me in sorted in seven branches. The intake for each branch is of 10 students.

Reliability Requirements:

The system has to be 100% reliable due to the importance of the data and damages that can be caused by incorrect/incomplete data.

Availability Requirements:

The system is available 100% for the user and it used 24 hours a day & 365 days a year. The system shall be operational 24 hours a day & 7 days a week.

EXISTING SYSTEM

- This system totally works manually
- All details of students are maintained in a single record
- This system require lot of man power
- Hard to operate and maintain
- This system paper based

Limitations:

- Administration cannot edit or modify scores after the deadline.
- Minor technical glitches and issues.

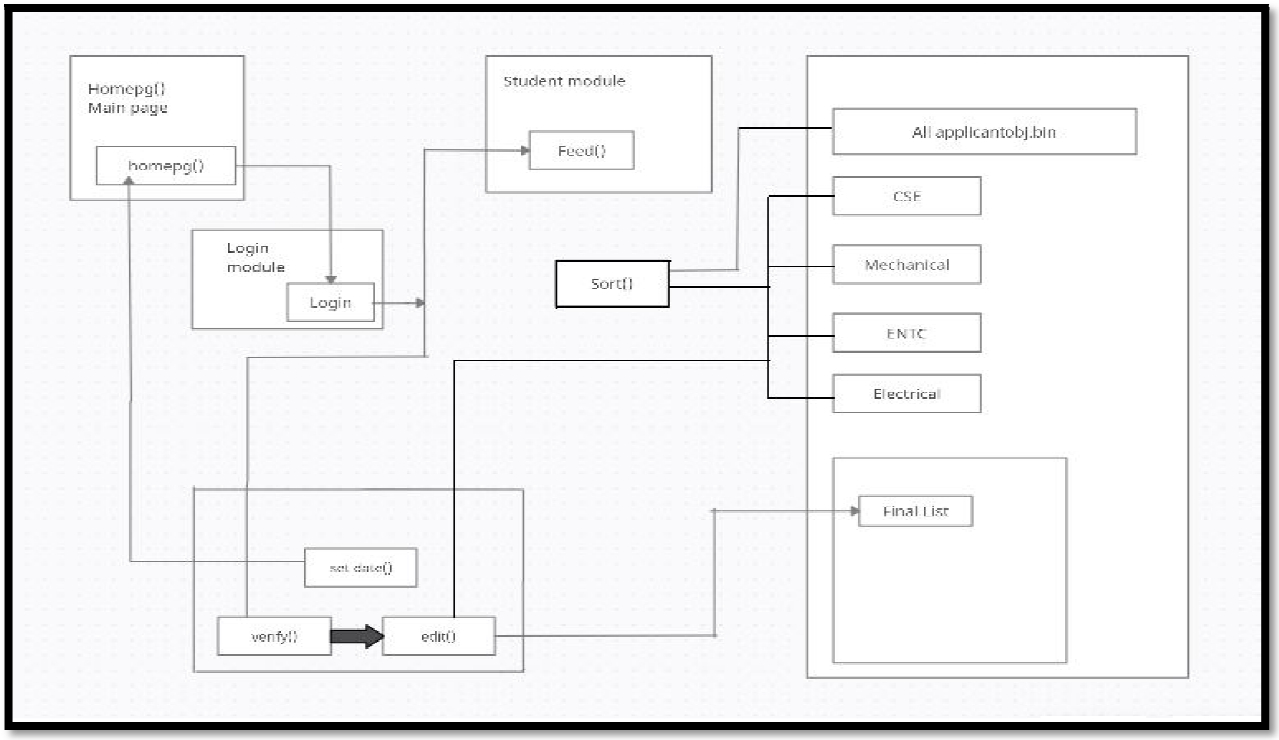
PROPOSED SYSTEM

- Our system enables the storage of the data after proper updation
- Our proposed system includes proper display of the student admission related information
- Our system helps with sorting process too
- Storage issue is also solved

Advantages of Proposed System:

- One of the greatest advantages of the online application system is that applicants can choose to submit their applications at their convenience.
- It will save pen and paper work
- Also, it will save time for admission process
- It works faster using that admission system
- Efficient reports

PROPOSED SYSTEM ARCHITECTURE



CHAPTER 3

Features:

Administrator:

o Login/Logout:

In that administrator login the interface using some particular username name and user id password. If the username and password will map then administrator will login the interface.

o View student information:

Administrator access to seeing the information related to student.

Example: Student Name, student address, 10^{th},12^{th}, JEE and CET marks whatever marks will see.

o Edit Student Information:

Administrator also access to edit the student information after student request Or grievance will accept.

o Enable/disable student accounts:

Also, Administrator allow the access of enable or disable student account.

o Search students: Using bubble sort technique search student name and their information.

3.1 STUDENT:

o Login/Logout:

similar like administrator student also allow login or logout module.

o View profile:

In that student allow to see or check their profile after submit their admission form.

o Edit profile:

After accepting the student grievance, student will change or update their profile in that admission form.

o Change password:

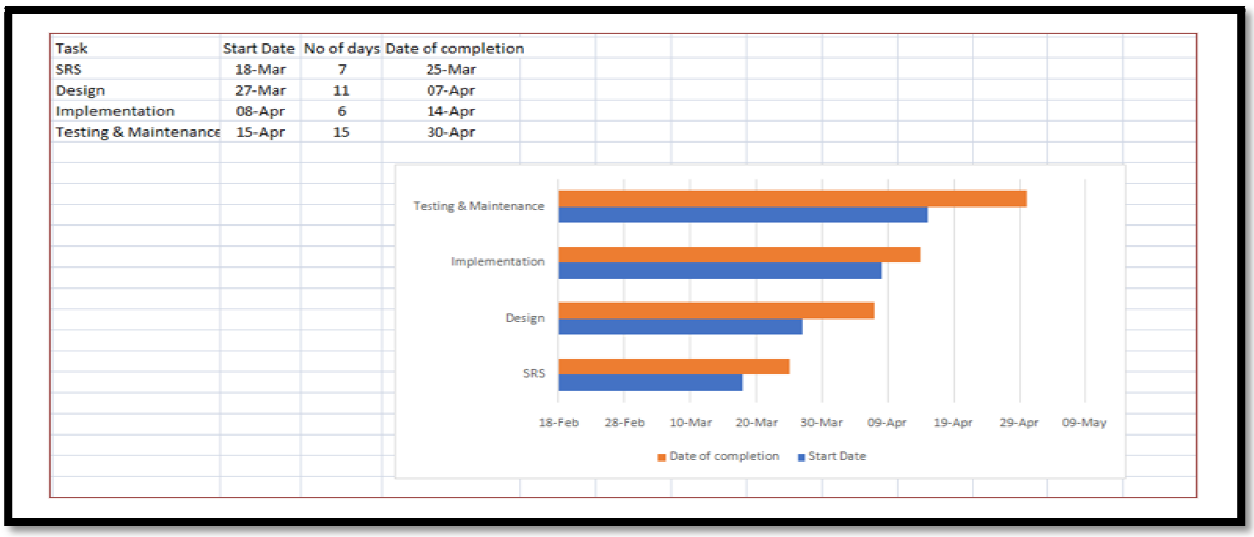
Also, student allow the access of change their login password for their security.

o Register new profile

CHAPTER 4

PROJECT SCHEDULE

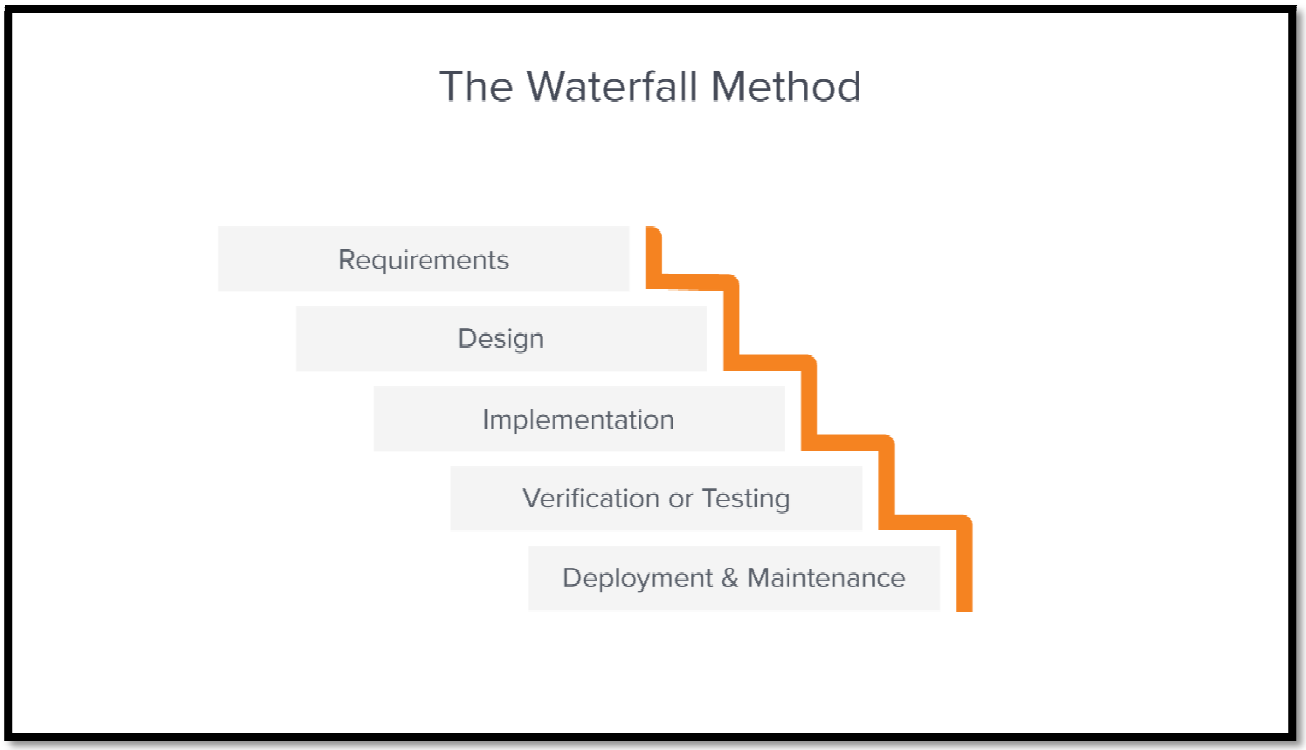
Gantt Chart



Software Hardware Requirement

- Windows 8 and above
- RAM 4GB
- Processor Intel i3 (dual core) and above

SDLC Model


The Waterfall Method
Requirements
Design
Implementation
Verification or Testing
Deployment & Maintenance

CHAPTER 5

PROJECT DESCRIPTION

Product Perspective

- Automate manual paper work on eat the time of student's admission in the college.
- As the main aim of the System is to reduce the cost needed for Admission Process, so it automatically reduces the manual power needed to perform the entire task and improve the quality of the work.
- Eliminate the paper work

Product Functions

- Applicant's list
- Student addition/ updating /deletion
- Fee details
- Important dates for admission process.

User Classes and Characteristics

There are mainly two kinds of users for the product.The users include:

- Administrator
- Student

Operating Environment

- Windows 8 and above
- RAM 4GB
- Processor Intel i3 (dual core) and above

CHAPTER 6

MODULES

Sorting module:

In this module students will get sorted:

1) According to caste
2) Students' CET marks.

Administrator module:

In this module when the administrator will enter his/her username and password, then he/she will enter into the administrator page and this page consists of following sub modules:

1) Applicant's list
2) Student addition/updating /deletion
3) Fee details
4) Important dates for admission process.

Student module:

In this module students are able to:

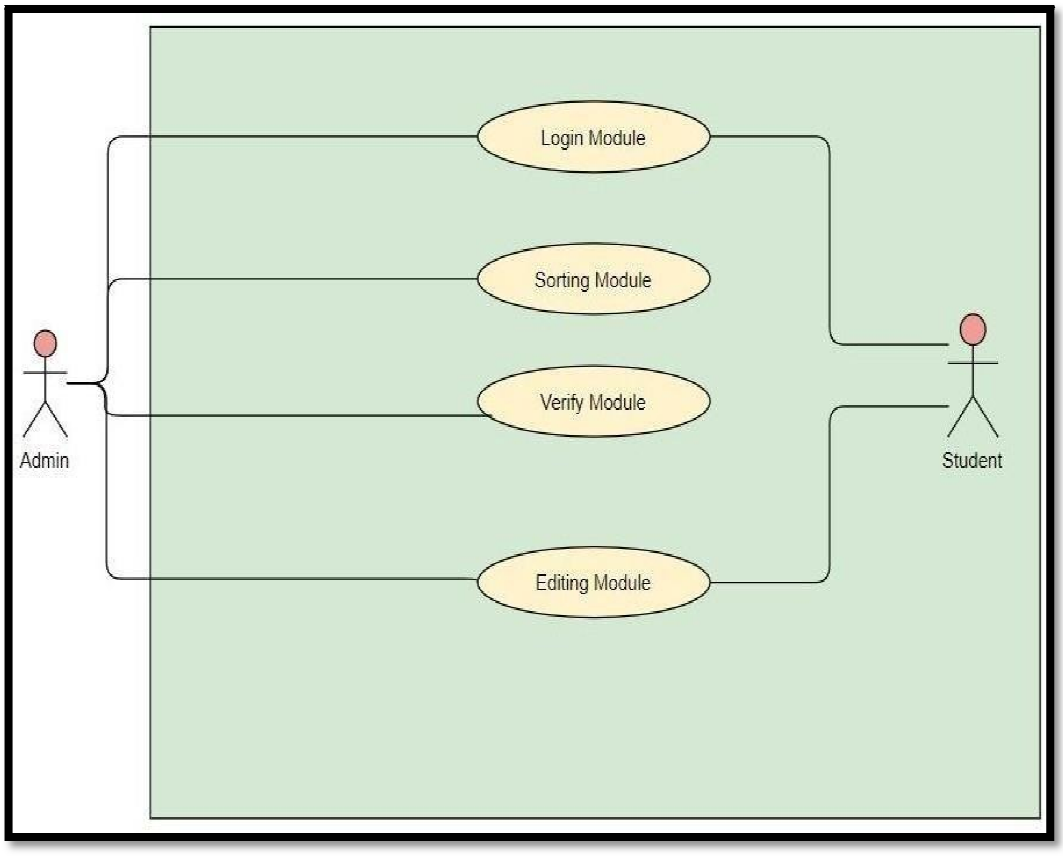
1) View their admission status
2) View their fee status
3) Cancel their admission.

CHAPTER 7

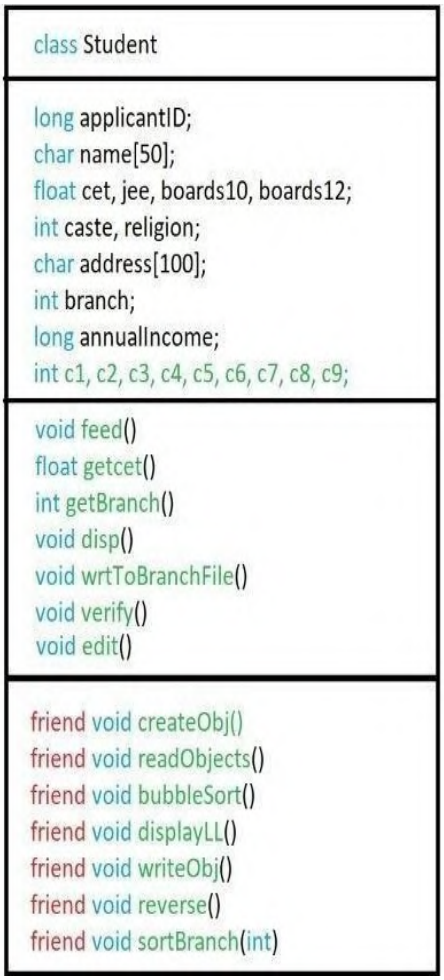
SYSTEM DESIGN

A software is designed in such a way that it should minimize the admission related work load using the concepts of C++ and minimizes the time of processing the data.

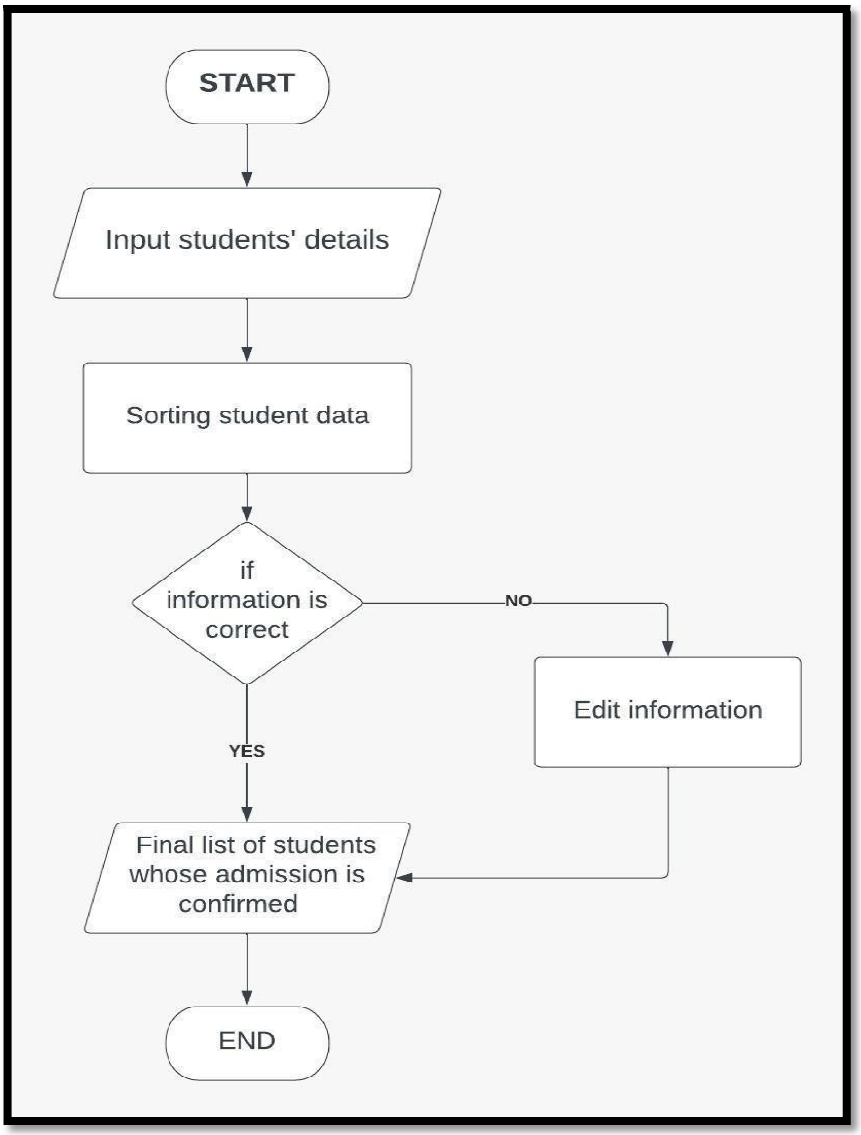
USE CASE DIAGRAM:



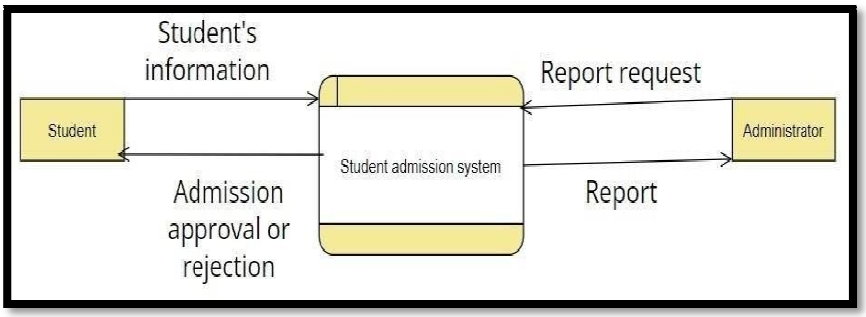
CLASS DIAGRAM:



Flow chart:



DFD:



CHAPTER 8
SYSTEM TESTING

1. Unit Testing

It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.

2. Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.

3. System Testing

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.

Test case	Objective	Input data	Expected result	Actual result	Status
TC1	Integer input validation	Any string	Module should give an error and ask for input again	If integer input is accepted if string asks for input again	Pass
TC2	Integer input validation	Any integer	Module should accept an integer if its within specified range	Accepted if integer input is within the range else asks for input again	Pass
TC3	File validation	File name	If file name exists it should open the file and perform desired operation, if file doesn't exist it should give an error	If file exists it accepts the input else gives an error	Pass
TC4	Search	Applicant ID of data type long	The function should accept long integer input and give error for string input	Function accepts long integer input and gives error for string input	Pass

Test case	Objective	Input data	Expected result	Actual result	Status
TC6	Username validation	Username of type string	Module should accept username if it belongs to file else should give an error	If username belongs to the file it is accepted else error is given	Pass
TC7	Password validation	Password of type string	Module should accept the password if it matches with its object password having same username	Accepted if password matches with the object password for same username	Pass

CHAPTER 9

SYSTEM IMPLEMENTATION

Login Implementation:

```
class Login{
  char userName[10];
  char password[10];

  public:
    Login(){}
    void feedAcc(){
      cout<<"\nThe username and password should be of maximum 10 characters";
      cout<<"\nEnter Username: ";
      cin.get(userName, 10);
      cin.ignore();
      cout<<"\nEnter Password: ";
      cin.get(password, 10);
      cin.ignore();
      return;
    }
    friend int login(const char*);
    friend int signup(const char*);
};
```

File handling for opening the file:

```
void wrtToBranchFile(const char *filename)
{
      ofstream fout1;
    fout1.open(filename, ios::app);
      if(!fout1.is_open()){
          cout<<"ERROR";
      }
      fout1<<applicantID<<"\t\t\t\t"<<cet<<"\t\t\t\t"<<name<<endl;
      fout1.close();
}
```

Write or fill the data of students and count the students:

```
void createAndWriteObj(){
  Student *ptr=new Student;
  ptr->feed();
  ofstream fout1;
  fout1.open("allApplicantsObj.bin", ios::binary | ios::app);
  if(!fout1.is_open()){
    cout<<"ERROR: line 222";
  }
  fout1.write((char*)ptr, sizeof(Student));
  fout1.close();
  count++;
  ofstream fout;
  fout.open("count.txt", ios::out);
  if(!fout.is_open()){
    cout<<"\nERROR: can not write count to the file";
  }
  fout<<count;
  fout.close();
}
```

For updating the applicant file:

```
void updateApplicantFile()
{
  temp=start;
  ofstream fout;
  fout.open("allApplicantsObj.bin", ios::trunc);
  fout.close();

  fout.open("allApplicantsObj.bin", ios::binary | ios::out);
  if(!fout.is_open()){
    cout<<"\nERROR : writeLL()";
  }
  while(temp!=NULL){
    fout.write((char*)temp, sizeof(Student));
    temp=temp->nxt;
  }
  fout.close();
}
```

Sorting :

```
void sortBranch(int branchID)
{
  Student *ptr;
  ptr=start;
  if(branchID==1){
    while(ptr!=NULL){
      if(branchID==ptr->getbranch()){
        ptr->wrtToBranchFile("CSE.txt");
      }
      ptr=ptr->nxt;
    }
  }
  else if(branchID==2){
    while(ptr!=NULL){
      if(branchID==ptr->getbranch()){
        ptr->wrtToBranchFile("MECH.txt");
      }
      ptr=ptr->nxt;
    }
  }
  else if(branchID==3){
    while(ptr!=NULL){
      if(branchID==ptr->getbranch()){
        ptr->wrtToBranchFile("ENTC.txt");
      }
```

```
      ptr=ptr->nxt;
    }
  }
  else if(branchID==4){
    while(ptr!=NULL){
      if(branchID==ptr->getbranch()){
        ptr->wrtToBranchFile("CIVIL.txt");
      }
      ptr=ptr->nxt;
    }
  }
  else if(branchID==5){
    while(ptr!=NULL){
      if(branchID==ptr->getbranch()){
        ptr->wrtToBranchFile("ELE.txt");
      }
      ptr=ptr->nxt;
    }
  }
  else if(branchID==6){
    while(ptr!=NULL){
      if(branchID==ptr->getbranch()){
        ptr->wrtToBranchFile("BIOTECH.txt");
      }
```

```
      ptr=ptr->nxt;
    }
  }
  else if(branchID==7){
    while(ptr!=NULL){
      if(branchID==ptr->getbranch()){
        ptr->wrtToBranchFile("ENV.txt");
      }
      ptr=ptr->nxt;
    }
  }
}
```

CHAPTER 10

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

Student Management System can be used by educational institutions to maintain their student records easily. Achieving this objective is difficult using the manual system as the information is scattered, can be redundant, and collecting relevant information may be very time-consuming. All these problems are solved by this project.

This system helps in maintaining the information of pupils of the organization. It can be easily accessed by the manager and kept safe for a long period of time without any changes.

Scope for future development

The admission system prototype which was developed is effective in a way that it will automate and make online admission instead of manual. Yet some areas in this study not explored in detail due to time constraint.

1. The payment system can be increased by using the original merchant account for credit card holders.
2. The admission offer letter can be sent online when student is admitted to the university.
3. Student can be assigned academic coordinator to advise them after admission to avoid problems of students facing for coming university and choosing subjects.

CHAPTER 11

REFERENCES

References:

1. https://www.slidscarnival.com/category/free-templates/elegant-presentations

2. 102427562-45883498-Project-Report-on-Student-Information-Management-System-Php-Mysql.pdf (123seminarsonly.com)

3. www.javapoint.com

4. www.geeksforgeeks.com