# **Acknowledgement**

We would like to acknowledge and special thanks to our project supervisor **Er. Ghan Bahadur Thapa** sir, who took interest in our project and guided us throughout the project by providing all the necessary ideas, information and knowledge for the project. His proper guidance and advice carried us through all the stages of writing this project documentation. We would like to thank our project members for letting this project have more enjoyable moments and for proper comments and suggestions.

We would like to thank to Aadikavi Bhanubhakta Campus for constant guidance and supervision, as well as for providing all the necessary study materials for the successful completion of this project.

Finally, we would like to thank our classmate for letting us through all the difficulties and for providing many valuable suggestions to complete this project successfully.

**Manoj Shrestha (76288013)**

**Nisha Rana (76288016)**

**Salma Khatun (76288020)**

# **Abstracts**

We proposed a web-based desktop application ***Result Management System*** which will be used as a platform for the interaction between the facilitators, students and admins in order to prepare the result fast and efficiently. The main objective of this project is to computerize the paperwork of the result management process. The work will be automated due to this software. This will reduce the time and effort of the colleges and data can be easily accessed by the students with their handy devices. Similarly, it helps to detect the mistakes while entering the marks data and attendances.

This system uses **JAVA** as a programming language with its framework **Swing and JDBC** and **MySQL** as a database for the system. Thus, the main purpose of this system is to reduce time and cost. This system also reduces the amount of paper and time to get the results.

**Keywords used: JAVA, Swing, JDBC, MySQL, computerize**

[Acknowledgement 1](#_Toc23213)

[Abstracts 2](#_Toc14815)

[LIST OF FIGURES 5](#_Toc1064)

[LIST OF ABBREVIATIONS 6](#_Toc22049)

[CHAPTER 1 : INTRODUCTION 7](#_Toc21616)

[1.1 Problem Statement 7](#_Toc10435)

[1.2 Motivation 7](#_Toc31286)

[1.3 Objectives 7](#_Toc9737)

[1.4 Project Scope 7](#_Toc24509)

[1.5 Limitations 8](#_Toc25098)

[1.4 Report Organization 8](#_Toc22550)

[CHAPTER 2 : REQUIREMENT ANALYSIS 9](#_Toc7556)

[2.1 Literature Review 9](#_Toc15159)

[2.2 Problem Definition 9](#_Toc10778)

[2.3 Requirement Analysis 9](#_Toc4011)

[2.3.1 Functional Requirement (Use Case) 10](#_Toc27066)

[2.3.2 Non-functional Requirement 12](#_Toc5877)

[2.4 Feasibility Study 12](#_Toc2559)

[2.4.1 Economic feasibility 12](#_Toc31196)

[2.4.1.1 Payback Analysis 12](#_Toc25543)

[2.4.1.2 Return on Investment 12](#_Toc31796)

[2.4.1.3 Net Present Value 12](#_Toc26709)

[2.4.2 Technical feasibility 13](#_Toc7143)

[2.4.3 Legal feasibility 13](#_Toc4286)

[2.4.5 Operational feasibility 13](#_Toc23880)

[2.4.6 Schedule feasibility 14](#_Toc10601)

[2.4.7 Risk feasibility 14](#_Toc22263)

[2.5 Structuring System Requirements 15](#_Toc25325)

[2.5.1 ER Diagram 15](#_Toc25049)

[2.5.2 Process Modeling(DFD Level-0) 16](#_Toc6811)

[2.5.3 Process Modeling(DFD Level-1) 17](#_Toc13816)

[CHAPTER 3 : SYSTEM DESIGN 18](#_Toc8291)

[3.1 System Architecture and Overview 18](#_Toc17227)

[3.2 System Design 19](#_Toc11527)

[3.2.1 Database Schema 19](#_Toc19882)

[3.2.1 Data Dictionary 19](#_Toc4784)

[3.2.2 UML Class Diagram 19](#_Toc19838)

[3.3 Interface Design 21](#_Toc17252)

[CHAPTER 4 : IMPLEMENTATION AND TESTING 32](#_Toc25332)

[4.1 Implementation and Overview 32](#_Toc25580)

[4.2 Technology used in my system 32](#_Toc22456)

[4.2.1 JAVA 32](#_Toc5931)

[4.2.2 Swing 32](#_Toc19324)

[4.2.3 JDBC 32](#_Toc19707)

[4.2.4 MySQL 33](#_Toc18911)

[4.3 Testing 33](#_Toc30972)

[4.3.1 Integration Testing 33](#_Toc15476)

[4.3.2 Unit Testing 33](#_Toc30990)

[4.3.3 System Testing 33](#_Toc492)

[CHAPTER 5: CONCLUSION AND RECOMMENDATION 34](#_Toc28735)

[5.1 Conclusion 34](#_Toc31671)

[5.2 Recommendation 34](#_Toc17005)

# **LIST OF FIGURES**

**Fig 1 : Use Case - Admin**

**Fig 2 : Use Case - Facilitator**

**Fig 3 : Use Case - Student**

**Fig 4 : ER Diagram**

**Fig 5 : Level - 0 DFD**

**Fig 6 : Level -1 DFD**

# **LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATION** | **DESCRIPTION** |
| RMS | School Management System |
| MYSQL | Structured Query Language (MySQL in database engine) |
| DFD | Data Flow Diagram |
| ERD | Entity Relationship Diagram |
| SDLC | Software Development Life Cycle |
| RDBMS | Relational Database Management System |
| JDBC | JAVA Database Connectivity |

# **CHAPTER 1 : INTRODUCTION**

## **1.1 Problem Statement**

Recently different colleges of Nepal are facing a lot of problems due to COVID. Students are not able to get the results in proper time as the facilitators cannot meet regularly with the college result section employee/staff. Marks cannot be given in time which delays the result. Similarly, extra staff are required for the result making process and result declaring process which creates a financial hit on the college economy. Also, the students who are far away on vacation have to come to colleges in the middle of vacation in order to get the results. This has a bad reflection on the educational system about not being able to apply the technology in this sector.

## **1.2 Motivation**

The importance of the result is great in today's world. The growth of an individual highly depends upon the quality of result and present college education system. And in a good college the chief objective is to stimulate the interest and curiosity in the students and provide facilities to the students, facilitators and parents to achieve the desired goal in a better and easier way. So the motivation of our project is to make tasks related to result management easy for the students, facilitators and parents.

## **1.3 Objectives**

* To manage all the information about the facilitators and students.
* To update the information easily.
* To provide the results to the students easily.
* To reduce the time for the result making process.
* To build a software for result management.
* To facilitate the attendance entry process of students by facilitators.
* To computerize the paperwork in the system and automate the work.

## **1.4 Project Scope**

In this era of technology colleges are based on paperwork. This system is the model for the user-friendly and effective management of result processing tasks. The ***Result Management System*** manages the result management process of the administration and provides a popper communication between the students, facilitators, admins and parents.

## **1.5 Limitations**

* Result info doesn’t contain the remarks of the students.
* Students and facilitators cannot create their own user id.
* Requires skill and knowledge of computers.
* It can only mark one at a time.
* Cost of this system will be expensive.
* Cannot enter the data without the internet.

## **1.4 Report Organization**

**Chapter 1:** This chapter explains about the overview, introduction,problem statement, motivations, project scopes and limitations of the system.

**Chapter 2:** This chapter covers all the history, methods, requirement specification and feasibility analysis and structured system requirements.

**Chapter 3:** Design of the result management system is explained in detail with all the necessary diagrams and brief functionality.

**Chapter 4:** Process of implementation and testing is described along with all the tools used for the development.

**Chapter 5:** Conclusion and future scope of the application are explained.

# **CHAPTER 2 : REQUIREMENT ANALYSIS**

## **2.1 Literature Review**

In ancient result management system were done by paper based, which are very time consuming and often leads to human error. Right now, there are numbers of schools and colleges that have implemented computer base result management system, which are very essential for human life. In general, ***Result Management System*** has managed to ease the task of admins, facilitators, students and others through a multi-function platform.

It is a paperless task that aids in automating current manual procedures and may be remotely monitored and managed on a server-based network. As there is no computerized system to add each record paper will be needed which will increase the cost the management of library.

## **2.2 Problem Definition**

The problem occurred before having computerized system includes:

* **File lost**: when computerized system is not implemented file is always lost because of human environment due to some human error there may be a lost of records.
* **File damaged**: when computerized system is not there file is always lost due to some accident like of water by some member on file accidentally. Besides some natural disaster like floods or fire may also damage the files.
* **Difficult to search resul**t: when there is no computerized system there is always difficulty in searching of records if the records are large in number.
* **Space consuming**: After the number of records become large the space for physical storage of file and records also if no computerized system is not implemented.

## **2.3 Requirement Analysis**

In our project, we have collected list of documents with sufficient and necessary requirements for the project development.

To derive the requirements, we have done better understanding of the products under development which we achieved through detailed and continuous communications with the project team throughout the software development process.

### **2.3.1 Functional Requirement (Use Case)**

**1. Admin**:- Admin can update facilitator and student info in the system.

**2. Student**:- Students can view facilitator and student info provided by the JAVA.

**3.** **View Instructor Info:-** Admins, facilitators and students can view facilitator info provided by the admin the instructor section.

**4. Auto management**:- System verifies login-id and provides all information automatically.

**5. Show management system**: - If the system is validated it then the system displays homepage.

**6. Update Instructor Info: -** Admins and facilitators can update facilitator info in the system.

**7. Update Student Info: -** Admins and facilitators can update student info in the system.

**8. View Student Info: -** Admin, facilitators and students can view student info in the system.

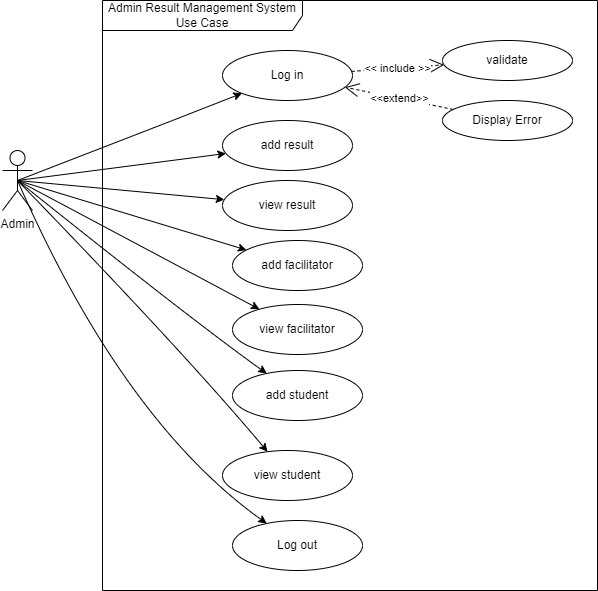


Fig 1: Use Case - Admin

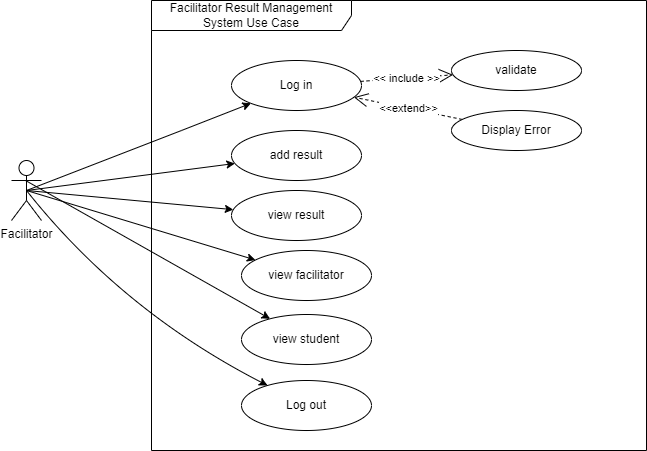


Fig 2: Use Case - Facilitator

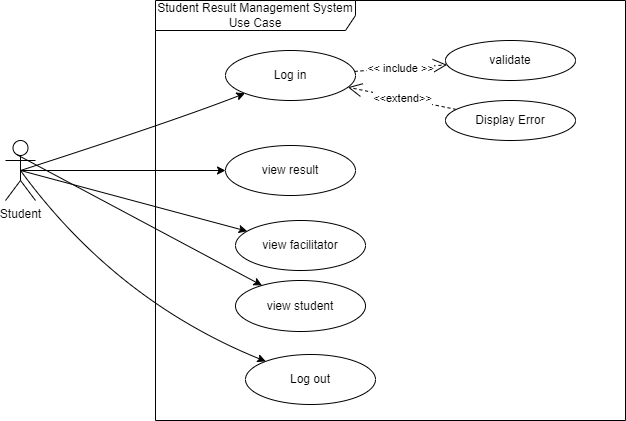


Fig 3: Use Case - Student

| Use Case Id: | 1.0 |
| --- | --- |
| Use Case Name: | Log In |
| Created By: | Manoj,Nisha,Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Student, Facilitator, Admin |
| Description: | This use case allows users to login into the system and perform different functions according to the user roles. All the users have to enter their unique username and password. |
| Precondition: | Users have to validate their account. |
| Post-condition: | System displays the relevant homepage. |
| Normal Courses: | Users enter the username and password..  System validates the username and password.  System verifies the username and password.  System detects the user type.  System displays the relevant homepage.  The use case ends. |
| Alternative Courses: | 1a. Upon missing username and password:  System prompts for empty username and password.  The use case resumes at step 1. |
| Exceptions: | Nil |
| Includes: |  |
| Priority: | High |
| Frequency of use: | 24 hours a day |
| Business Rule: | User must login through a registered and verified account. |
| Special Requirement: |  |
| Assumptions: | Nil |

|  |  |
| --- | --- |
| Use Case Id: | 2.0 |
| Use Case Name: | Add Facilitator |
| Created By: | Manoj, Nisha, Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Admin |
| Description: | Admin can add facilitator info in the add facilitator section. |
| Precondition: | Admin login into the system.  Admin opens the add facilitator section. |
| Post-condition: | Admin can add the facilitator info. |
| Normal Courses: | Admin enter the username, password, class, subject, id, phone number, email of the facilitator.  System validates the details.  System stores the details in database.  The use case ends. |
| Alternative Courses: | 1a. Upon not entering facilitator info:  Prompts an message with empty field. |
| Exceptions: | User’s don’t add the facilitator info. |
| Includes: |  |
| Priority: | Low |
| Frequency of use: | 24 hours a day |
| Business Rule: |  |
| Special Requirement: |  |
| Assumptions: | Nil |
| Notes and Issues: |  |
| Use Case graphics: |  |

|  |  |
| --- | --- |
| Use Case Id: | 2.1 |
| Use Case Name: | View Facilitator |
| Created By: | Manoj, Nisha, Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Admin,Student,Facilitator |
| Description: | Users can view facilitator info provided by the admin in the view facilitator section. |
| Precondition: | 1.Users login into the system.  2.Users opens the view facilitator section. |
| Post-condition: | 1.Users can view the facilitator info. |
| Normal Courses: | 1.Users get the facilitator details.  2.The use case ends. |
| Alternative Courses: | 1a. Upon not getting facilitator info:  1. Shows an empty page with no info message. |
| Exceptions: | 1.User’s don’t view the facilitator info. |
| Includes: |  |
| Priority: | Low |
| Frequency of use: | 24 hours a day |
| Business Rule: |  |
| Special Requirement: |  |
| Assumptions: | Nil |
| Notes and Issues: |  |
| Use Case graphics: |  |

|  |  |
| --- | --- |
| Use Case Id: | 3.0 |
| Use Case Name: | Add Student |
| Created By: | Manoj, Nisha, Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Admin |
| Description: | Admin can add student info in the add student section. |
| Precondition: | 1.Admin login into the system.  2.Admin opens the add student section. |
| Post-condition: | 1.Admin can add the student info. |
| Normal Courses: | 1.Admin enter the username, password, class, id, phone number, email of the student.  2.System validates the details.  3.System stores the details in database.  4.The use case ends. |
| Alternative Courses: | 2a.Upon not entering student info:  1. Prompts an message with empty field. |
| Exceptions: | 1.User’s don’t add the facilitator info. |
| Includes: |  |
| Priority: | Low |
| Frequency of use: | 24 hours a day |
| Business Rule: |  |
| Special Requirement: |  |
| Assumptions: | Nil |
| Notes and Issues: |  |
| Use Case graphics: |  |

|  |  |
| --- | --- |
| Use Case Id: | 3.1 |
| Use Case Name: | View Student |
| Created By: | Manoj, Nisha, Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Admin,Student,Facilitator |
| Description: | Users can view student info provided by the admin in the view student section. |
| Precondition: | Users login into the system.  Users opens the view student section. |
| Post-condition: | Users can view the student info. |
| Normal Courses: | Users get the student details.  The use case ends. |
| Alternative Courses: | 1a. Upon not getting student info:  Shows an empty page with no info message. |
| Exceptions: | Users don’t view the student info. |
| Includes: |  |
| Priority: | Low |
| Frequency of use: | 24 hours a day |
| Business Rule: |  |
| Special Requirement: |  |
| Assumptions: | Nil |
| Notes and Issues: |  |
| Use Case graphics: |  |

|  |  |
| --- | --- |
| Use Case Id: | 4.0 |
| Use Case Name: | Add Result |
| Created By: | Manoj, Nisha, Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Admin, Facilitator |
| Description: | Admin and facilitator can add marks info in the add marks section but facilitator can add marks according to the subject he has been granted. |
| Precondition: | 1.Admin and facilitator login into the system.  2.Admin and facilitator opens the add marks section. |
| Post-condition: | 1.Admin and facilitator can add the marks info. |
| Normal Courses: | 1.Admin enter the respective subject marks.  2.System calculates the total and percentage of the marks.  3.System stores the details in database.  4.The use case ends. |
| Alternative Courses: | 1a.Upon not entering marks info:  1. Prompts an message with empty field. |
| Exceptions: | 1.Admin or facilitator’s don’t add the marks info. |
| Includes: |  |
| Priority: | High |
| Frequency of use: | 24 hours a day |
| Business Rule: |  |
| Special Requirement: | Students must have their username and password. |
| Assumptions: | Nil |
| Notes and Issues: |  |
| Use Case graphics: |  |

|  |  |
| --- | --- |
| Use Case Id: | 4.1 |
| Use Case Name: | View Result |
| Created By: | Manoj, Nisha, Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Admin,Student,Facilitator |
| Description: | Users can view student info provided by the admin in the view marks section. |
| Precondition: | 1.Users login into the system.  2.Users opens the view marks section. |
| Post-condition: | 1.Users can view the marks info. |
| Normal Courses: | 1.Users get the marks details.  2.The use case ends. |
| Alternative Courses: | 1a. Upon not getting marks info:  1. Shows an empty page with no info message. |
| Exceptions: | 1.Users don’t view the marks info. |
| Includes: |  |
| Priority: | High |
| Frequency of use: | 24 hours a day |
| Business Rule: |  |
| Special Requirement: | Students must have their username and password. |
| Assumptions: | Nil |
| Notes and Issues: |  |
| Use Case graphics: |  |

| Use Case Id: | 1.0 |
| --- | --- |
| Use Case Name: | Log out |
| Created By: | Manoj,Nisha,Salma Last updated by: |
| Date Created: | December 25, 2022 Date last updated: |
| Actor(s): | Student, Facilitator, Admin |
| Description: | This use case allows users to log out from the system to return back to the login page. |
| Precondition: | 1.Users have to login into their account. |
| Post-condition: | 1.System displays the login page erasing all the previous sessions.. |
| Normal Courses: | 1.Users click on the log out button.  2.System erases the session of the users.  3.System redirect the user to login page.  4.The use case ends. |
| Alternative Courses: | 1a. When user doesn't click on the log out button:   * User remain logged in. |
| Exceptions: | Nil |
| Includes: |  |
| Priority: | Medium |
| Frequency of use: | 24 hours a day |
| Business Rule: | User must login through a registered and verified account. |
| Special Requirement: |  |
| Assumptions: | Nil |

### **2.3.2 Non-functional Requirement**

1. **Security**: - Only authorize users can access the system with username and password
2. **Performance**: - Easy tracking of records and update can be done.
3. **User Friendly**: - The system is very interactive.
4. **Availability**: The system is available all the time, no time constraint

## **2.4 Feasibility Study**

Feasibility study is an analysis of how successfully a project can be completed, accounting for factors that effect it such as economic, technical, legal and other factors, Similarly, we use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it..

### **2.4.1 Economic feasibility**

Economic feasibility is done in order to know whether the project is benefit-able or not. In economic feasibility we compare the cost and benefit involved in the system before, during and after the system has been completed. Here, the time is also considered to be as capital. Economic feasibility is done through several methods.

#### **2.4.1.1 Payback Analysis**

#### **2.4.1.2 Return on Investment**

ROI is the percentage rate that measures the relationship between the amounts business get back from the investment and the amount invested.

Lifetime ROI = (Estimated lifetime benefits - Estimated lifetime costs)/Estimated lifetime costs

So,

Estimated lifetime costs = Rs. 15000

Estimated lifetime benefits = Rs. 60000

Lifetime ROI = (60000-15000)/15000

=45000/15000

=3%

RMS is likely to give us 3% benefit from the project in our lifetime duration until its use for a particular organization.

#### **2.4.1.3 Net Present Value**

### **2.4.2 Technical feasibility**

Our system is built for any type of operating system. It can upgrade to the upgrades of software engineering principles.It uses JAVA as main programming language for production in which we can add new classes and modules as per the requirement.

**Hardware Specification**

* Laptop/Desktop

**Software Specification**

* SQL Server
* Intellij IDEA
* Windows/Linux
* Diagram Tool - draw.io

Each of the above resources are freely available and technical skills required are manageable.

Initially this desktop application will be provided locally through emails but later after it gets finished it will be available on different stores.Bandwidth application in this application is very low as it doesn't contain any multimedia aspect.

### **2.4.3 Legal feasibility**

RMS uses the freely available tools and will be easily available to the users. Only the maintenance cost will be charged from the main potential members.

This system is available for people for every background people. So it doesn’t cause any legal problems in the society.

### **2.4.5 Operational feasibility**

In operational feasibility,we look at the available resources whether they are available or not.

* RMS makes easy and fast for the access of the result and making it. Students can access 24 hrs after the result has been published. After the publish of the result students can access their reports 24 by 7. Next time the result will be updated. So it is feasible.
* This saves a lot of time for the admin and teachers along with students which gives a plus benefit for the user.
* After this system is developed, it will be surely used by the schools as they need this.
* There are sufficient developers so this project can be developed in time.
* The database will be well protected under certain security. Hence, information of the user will be secure.
* RMS has been started from the users needs of Result Making Software. So it will be feasible.

The above factors determines that RMS is operationally feasible.

### **2.4.6 Schedule feasibility**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Starting | Ending | Duration |
| Requirement gathering and feasibility study | 10/1/2022 | 11/10/2022 | 40 |
| System Analysis | 11/10/2022 | 12/15/2022 | 35 |
| System Design | 12/15/2022 | 1/15/2023 | 31 |
| Coding | 1/15/2023 | 2/5/2023 | 21 |
| Testing | 2/5/2023 | 2/13/2023 | 8 |
| Debugging | 2/13/2023 | 2/27/2023 | 14 |
| Implementation | 2/27/2023 | 3/7/2023 | 8 |
| Maintenance | 3/10/2023 | 3/30/2023 | 20 |
| Documentation | 10/1/2022 | 3/30/2023 | 180 |

### **2.4.7 Risk feasibility**

While doing the project there are certain risk involved in that process . So here are what could be the risk in our project.

1. Will the project run on Linux OS too?

Yes, this project is platform independent that means it can run on Linux OS too.

1. Will it get hanged or not ?

It doesn't use multimedia aspects so it takes less bandwidth. It is less likely to get hanged as it takes less memory and less internet package..

1. Does this need any technical skills?

As it will be based on GUI system. A user with little knowledge of English language will be able to use this application easily.

1. Will this create a market or not ?

As nowadays there is time to time lockdown and all that problem is good to make our project successful. Schools are forced to take exams and publish result online. So, our project will take up the market.

1. Can the change in project be tracked throughout the software life cycle process?

GIT will be used to track the change that happen in the application.

1. Will the old algorithm be applicable for the result making process?

In order to solve this, a new algorithm will be generated for the result making process.

## **2.5 Structuring System Requirements**

### **2.5.1 ER Diagram**

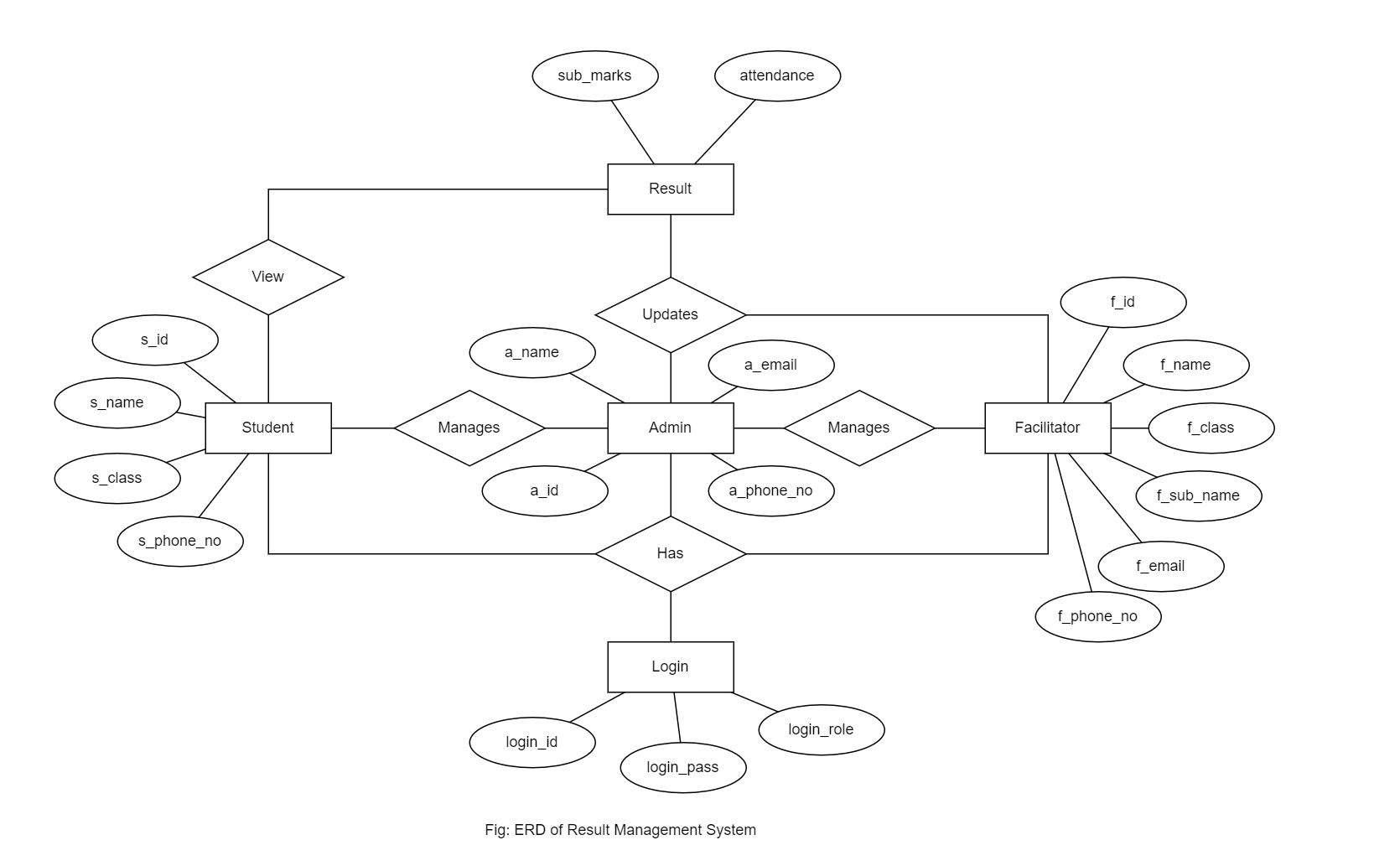


Fig 4: ER Diagram

The above diagram explains the relationship between the database where rectangle represents entity, oval represents attributes and Diamond represents relation. There are three entities with their respective attributes. Admin acts as one of the entities and has attributes like admin\_ id, admin\_ name, admin\_ phone\_no, admin\_email. And another entity Facilitator acts as another entity and it has attributes like facilitator\_ id, facilitator\_ name, facilitator\_class, facilitator\_sub\_name,facilitator\_email, facilitator\_phone\_no. The student is another entity and it has attributes like student\_ id, students\_ name,student\_class, and student\_phone\_no. The login includes login\_id,login\_pass and login\_role.

### **2.5.2 Process Modeling(DFD Level-0)**

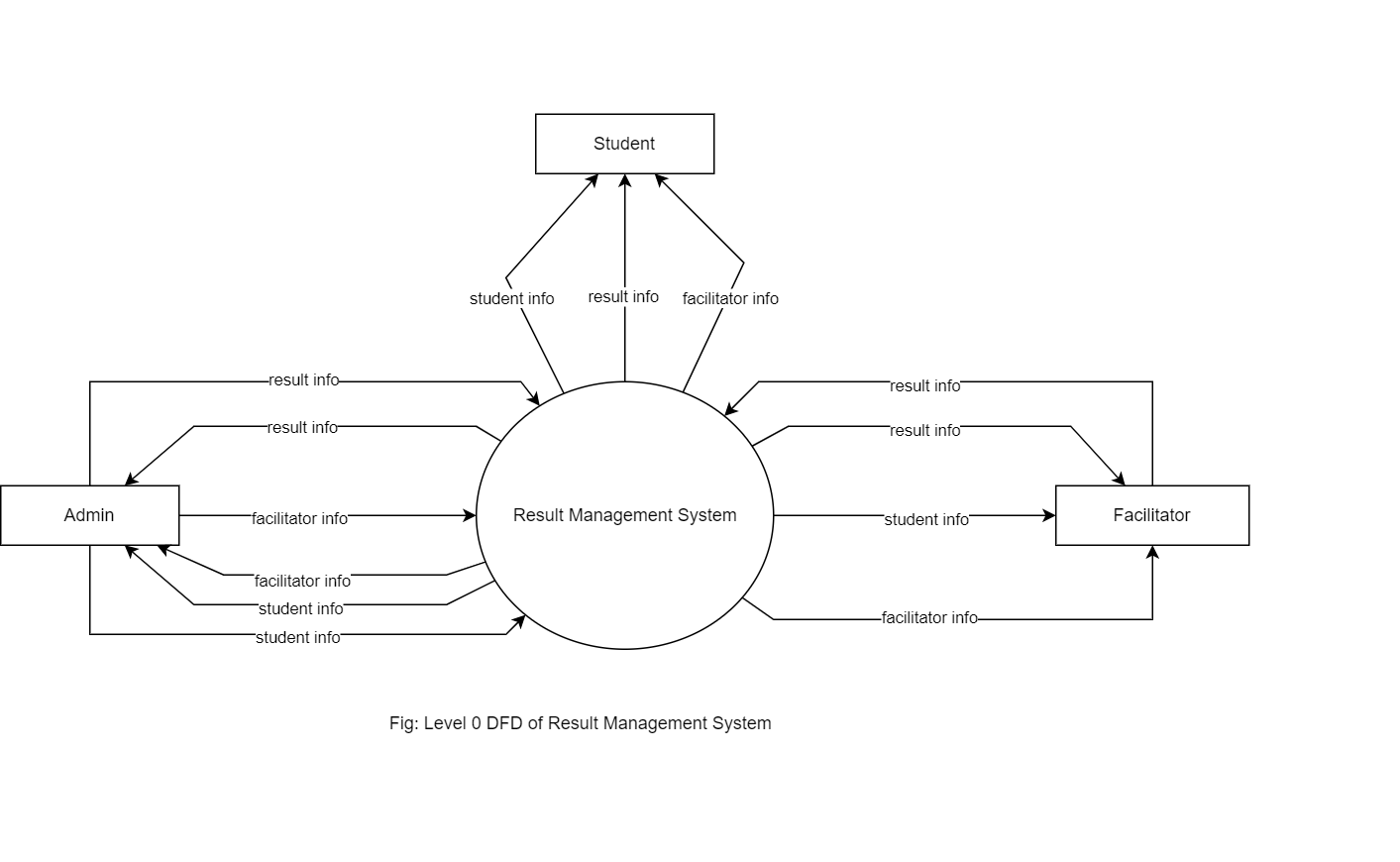


Fig 5: Level - 0 DFD

The above diagram shows the DFD zero level where system manages the students Information. It’s a basic overview of the whole Result Management system where student’s login to the system. System takes the students information and sends data to students ID. Admin performs updating data and adding new students records in addition admin manages student’s data, facilitators data and courses information and update report.

### **2.5.3 Process Modeling(DFD Level-1)**

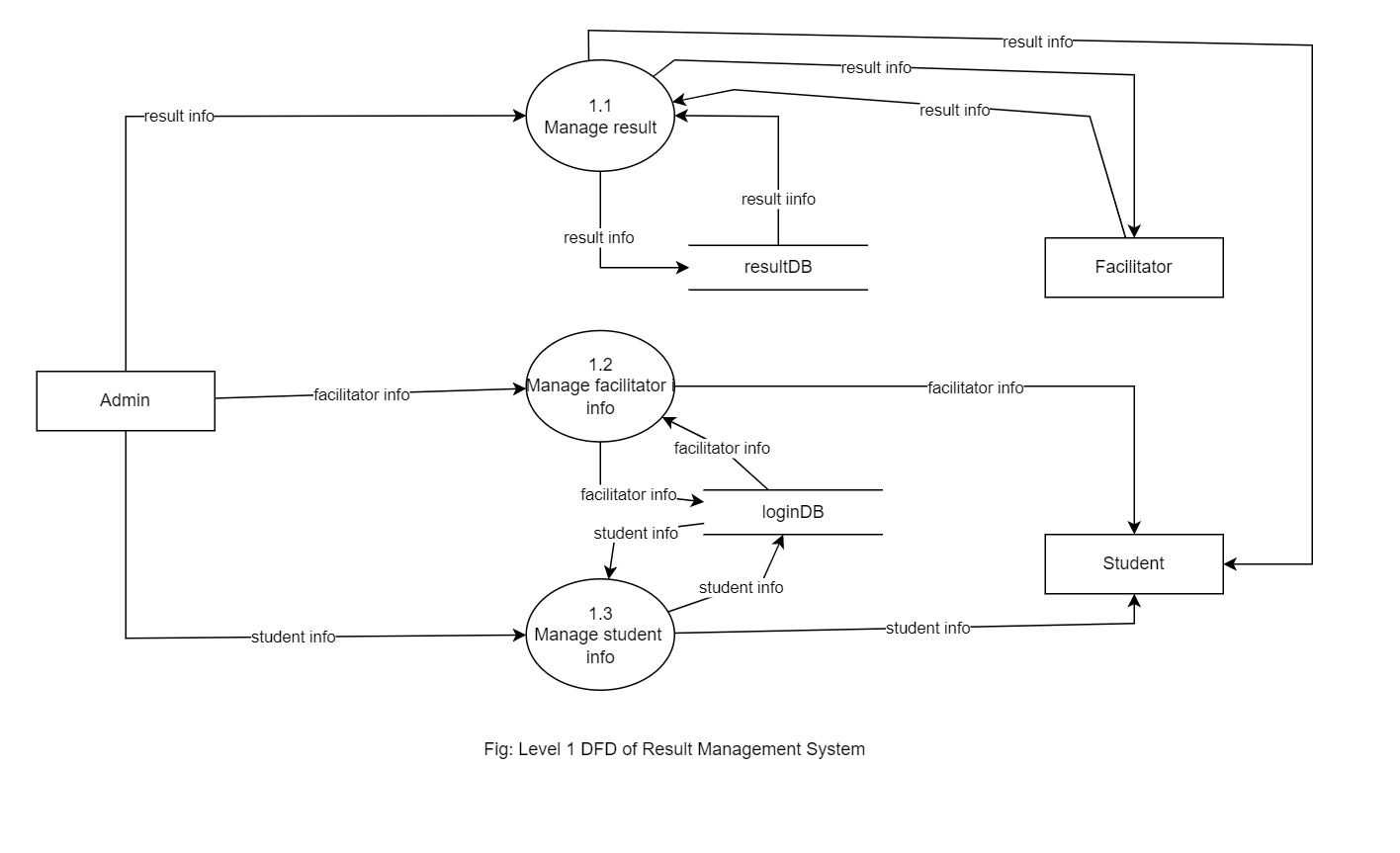


Fig 6: Level 1 DFD

The above diagram is the DFD level-1, where it explains the working process of system. These procedures require information such as record of students, enrolls, transactions and instructors from which served as the bases for the Result Management System. There is login process in the system which is authenticated and authorized by admin. Students’ login to the system via email\_ id then system verify it. If student\_ id is validated then the system opens the home page, if the id is invalidated then the system notifies unauthorized.

# **CHAPTER 3 : SYSTEM DESIGN**

## **3.1 System Architecture and Overview**

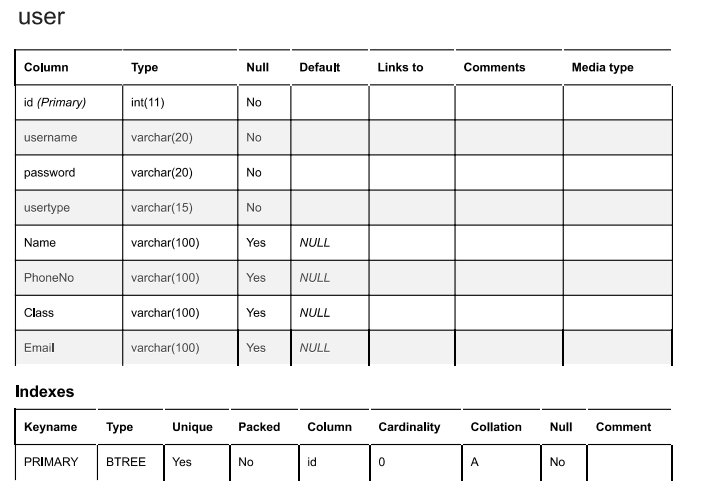
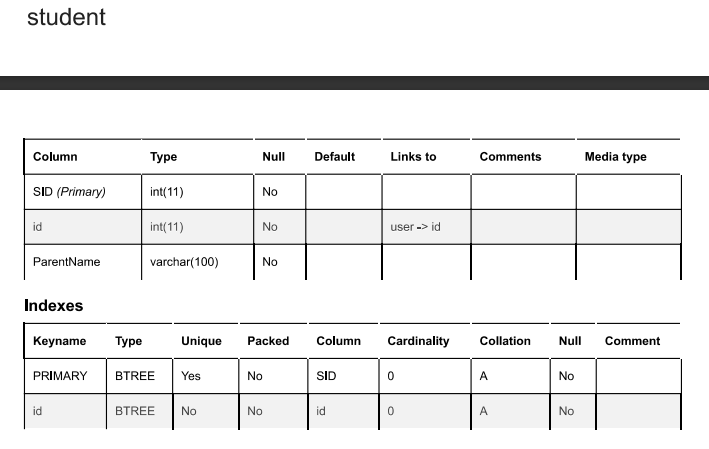
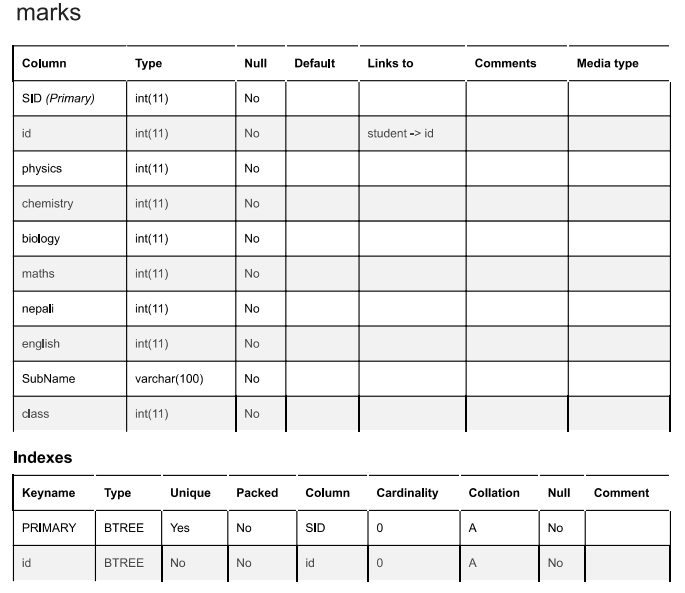
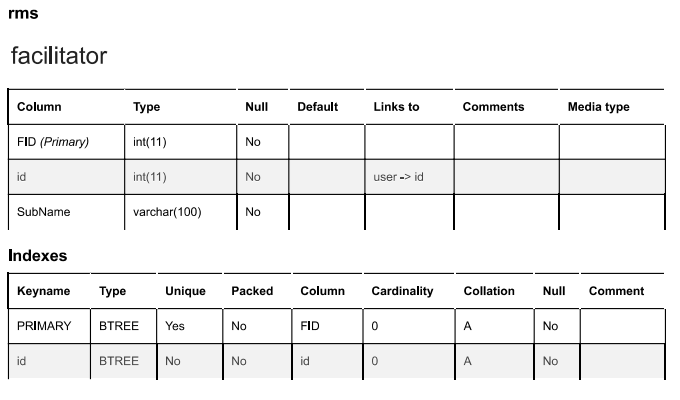
System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. The designs can be defined in graphical or textual modelling languages. Also, it is the process of creating or altering systems, along with the processes, practices, models. And methodologies used to develop them.

We have developed “Result Management System” which require Web-server for its functioning. To use this system we need a JAVA, laptop / desktop and the users.

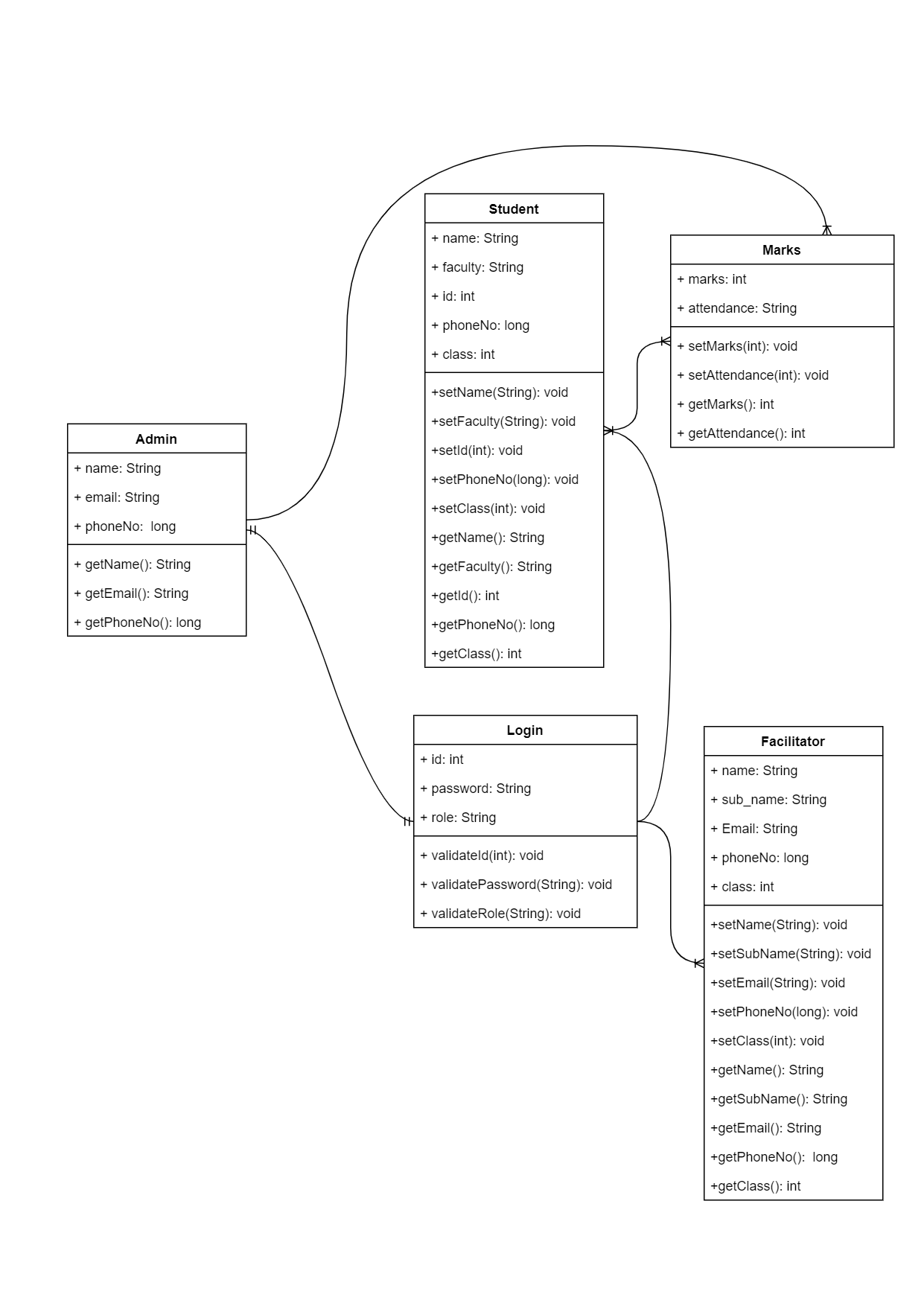
## **3.2 System Design**

### **3.2.1 Database Schema**

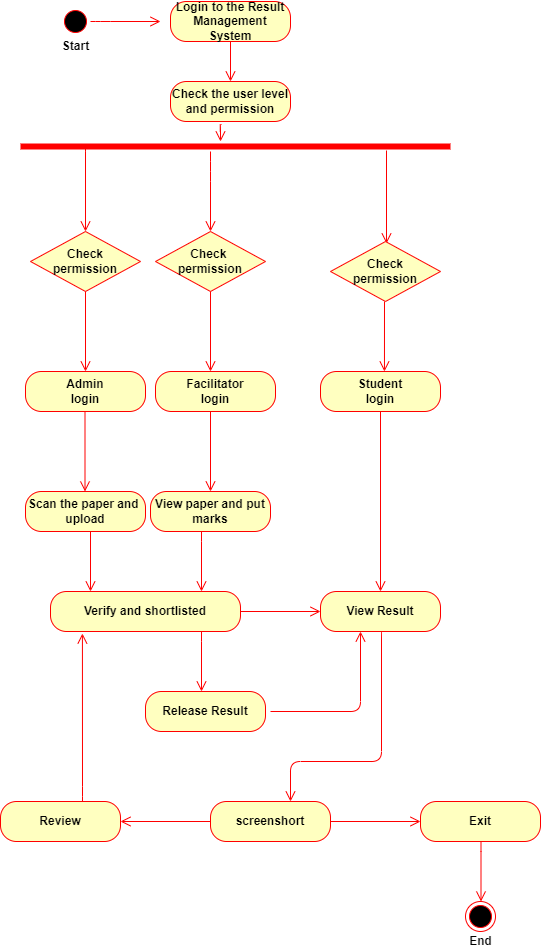
### **3.2.1 Data Dictionary**



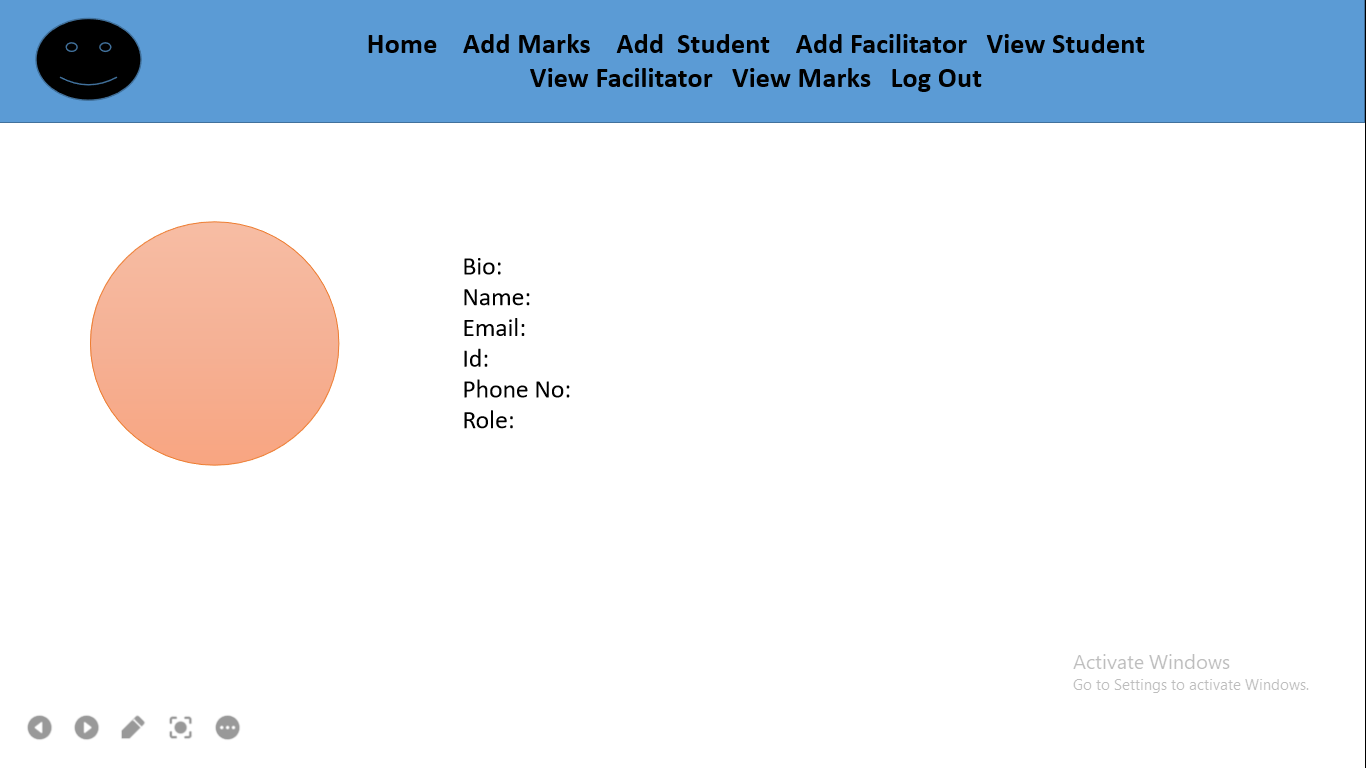
### **3.2.2 UML Class Diagram**

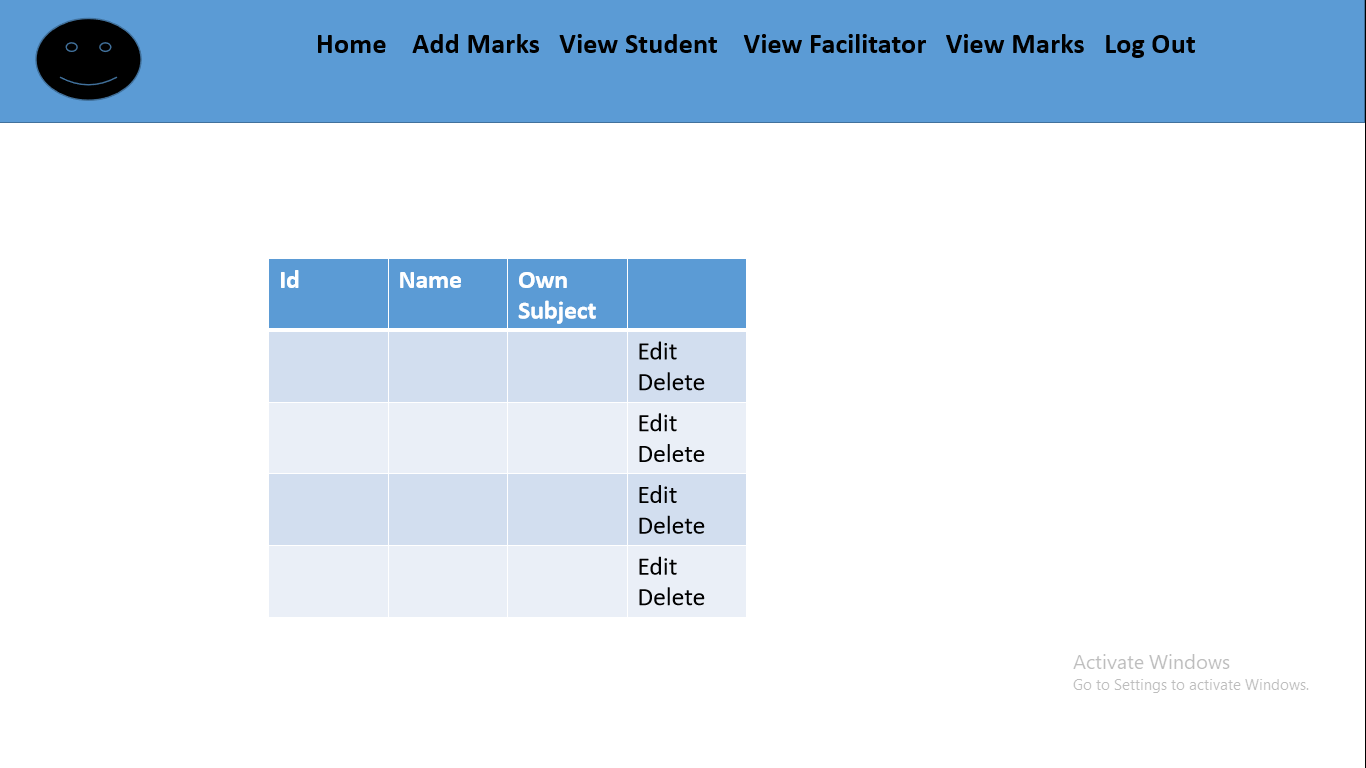
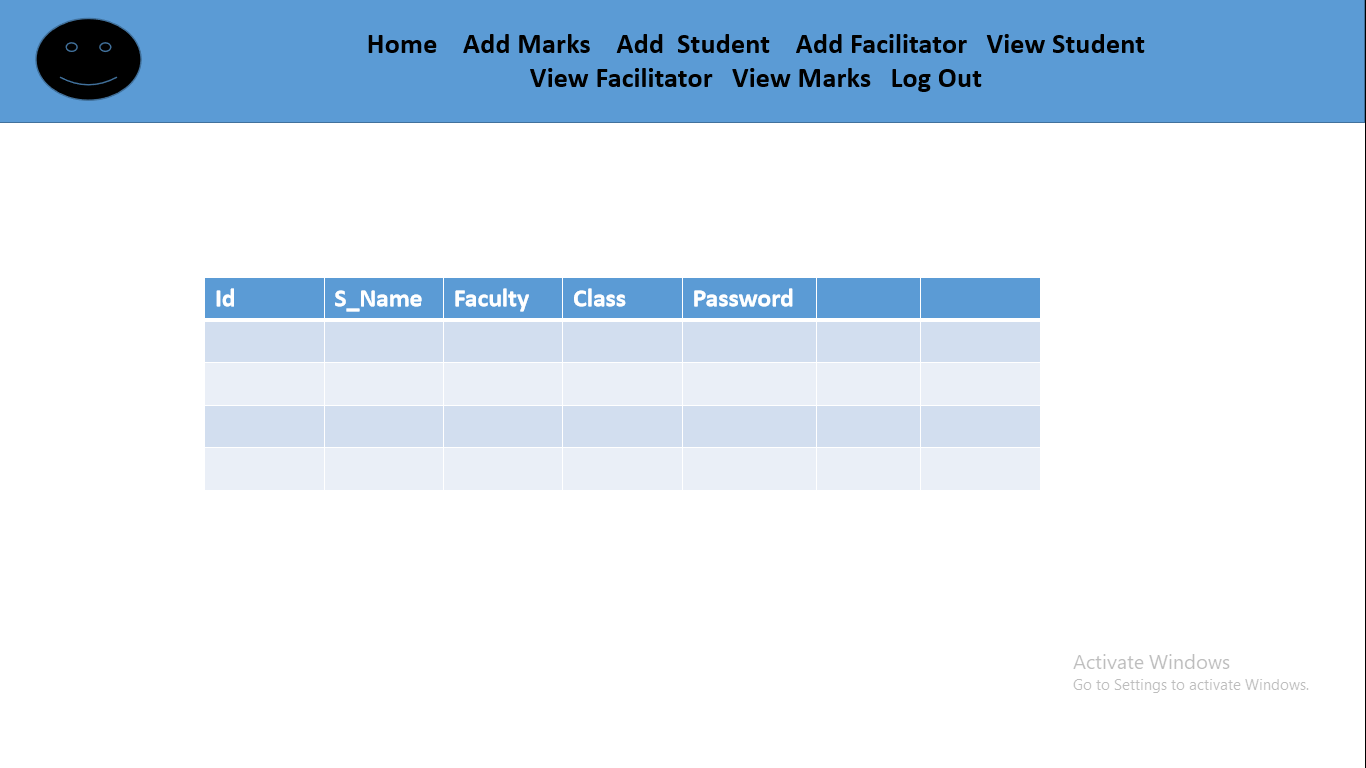
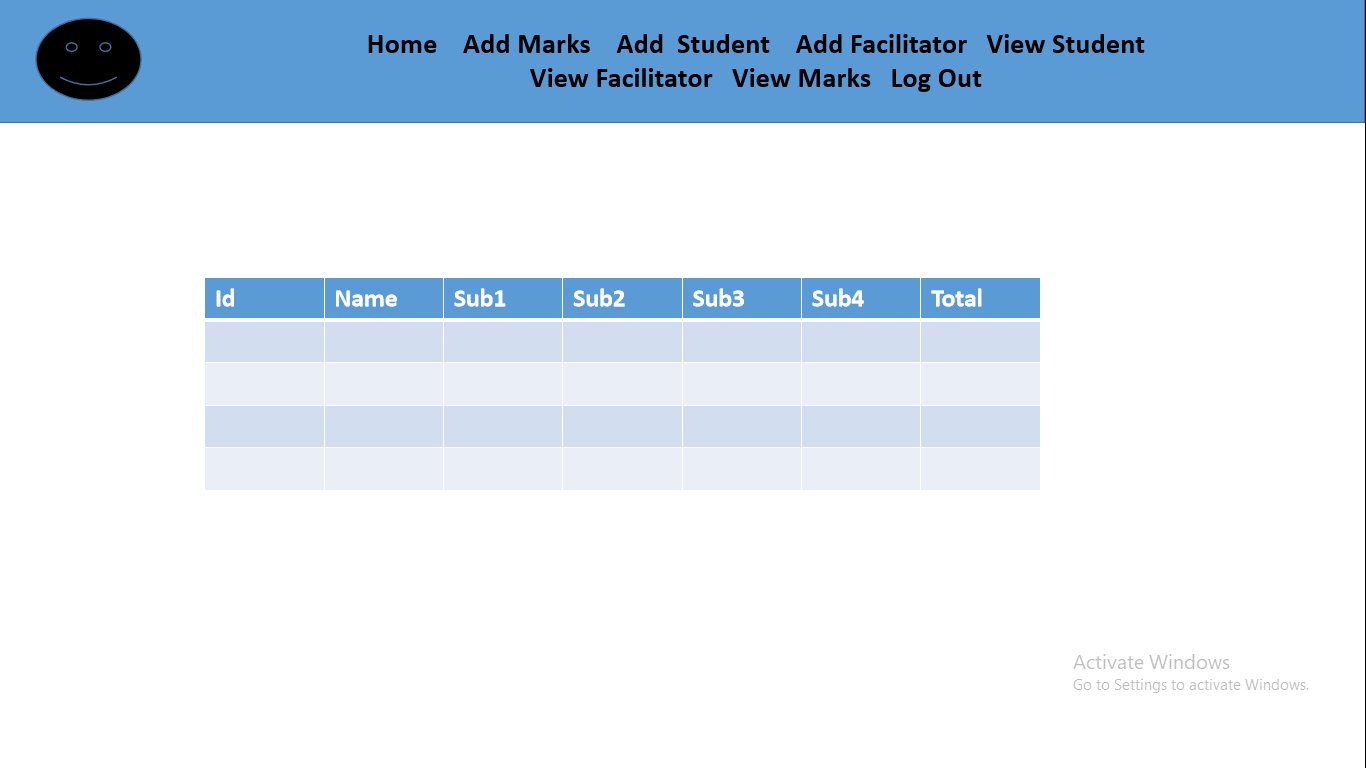
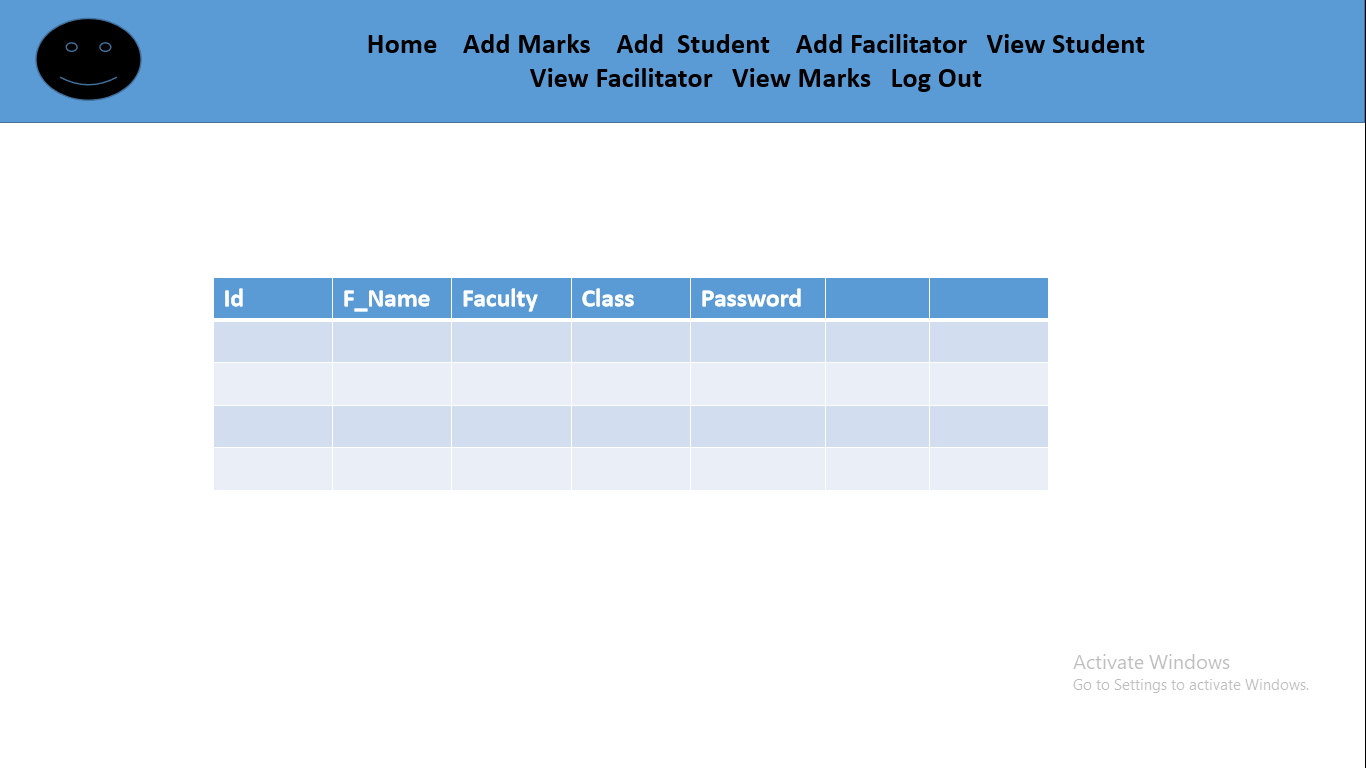
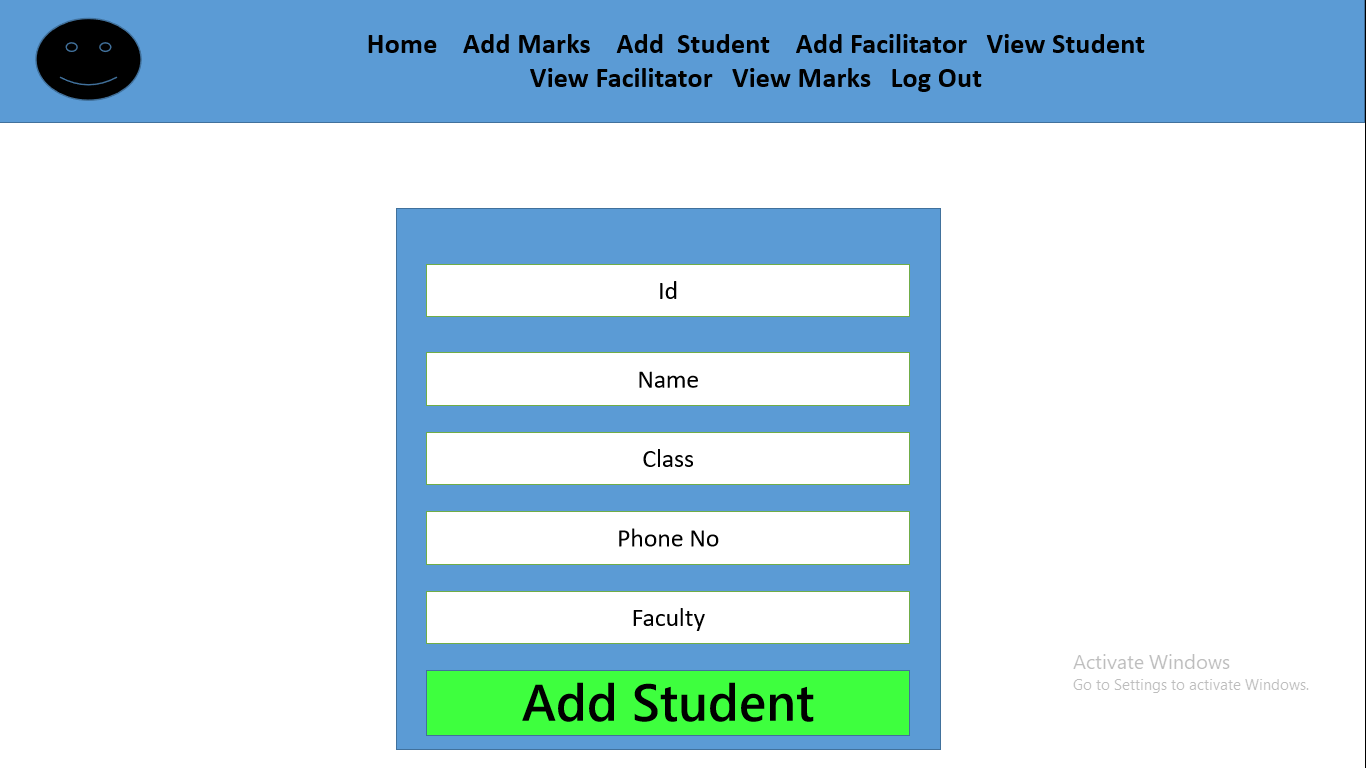
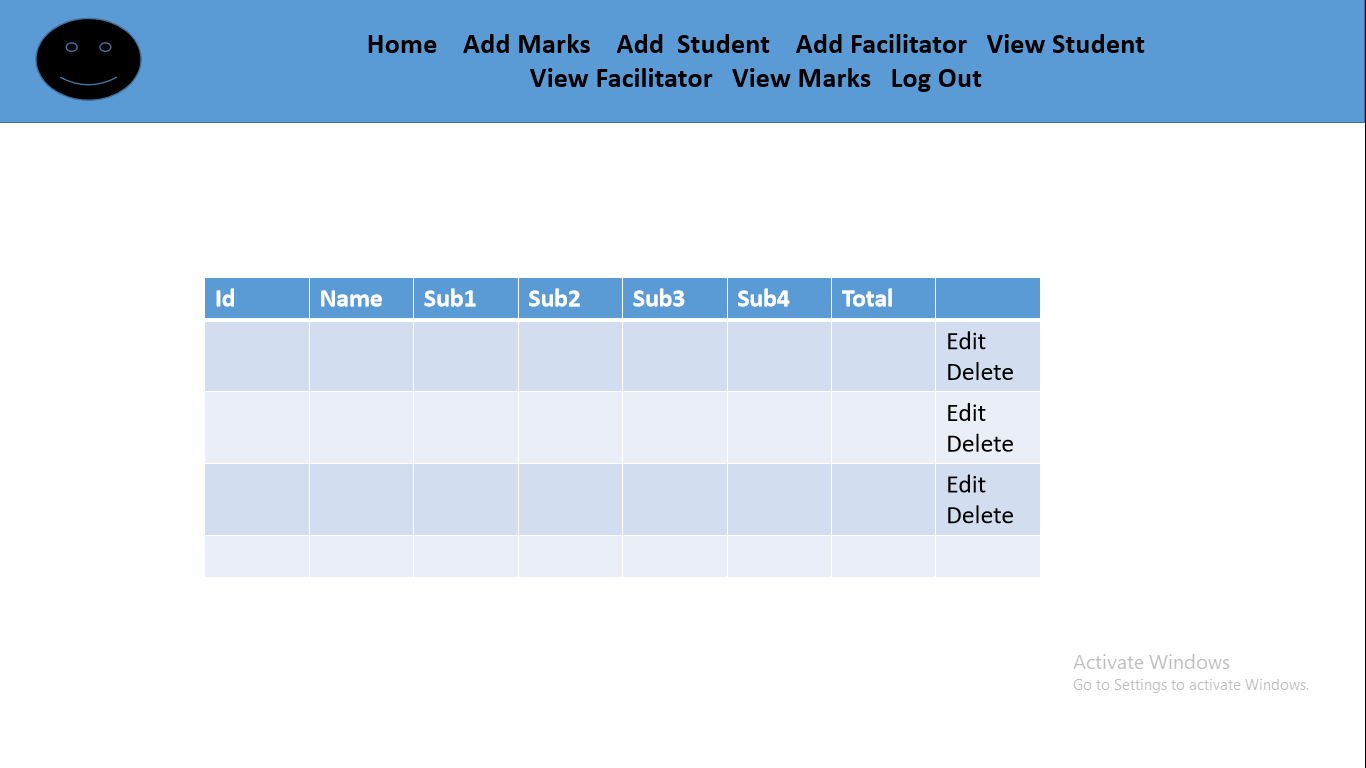
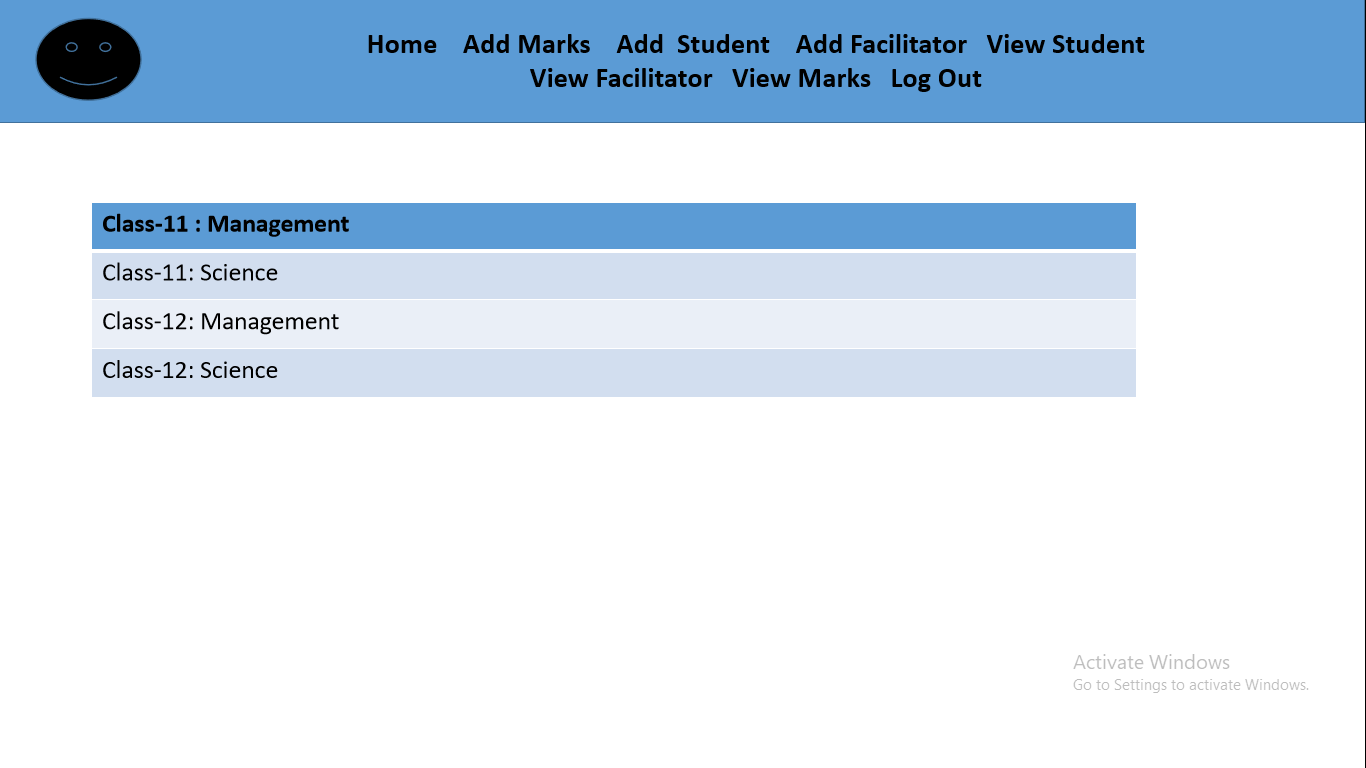
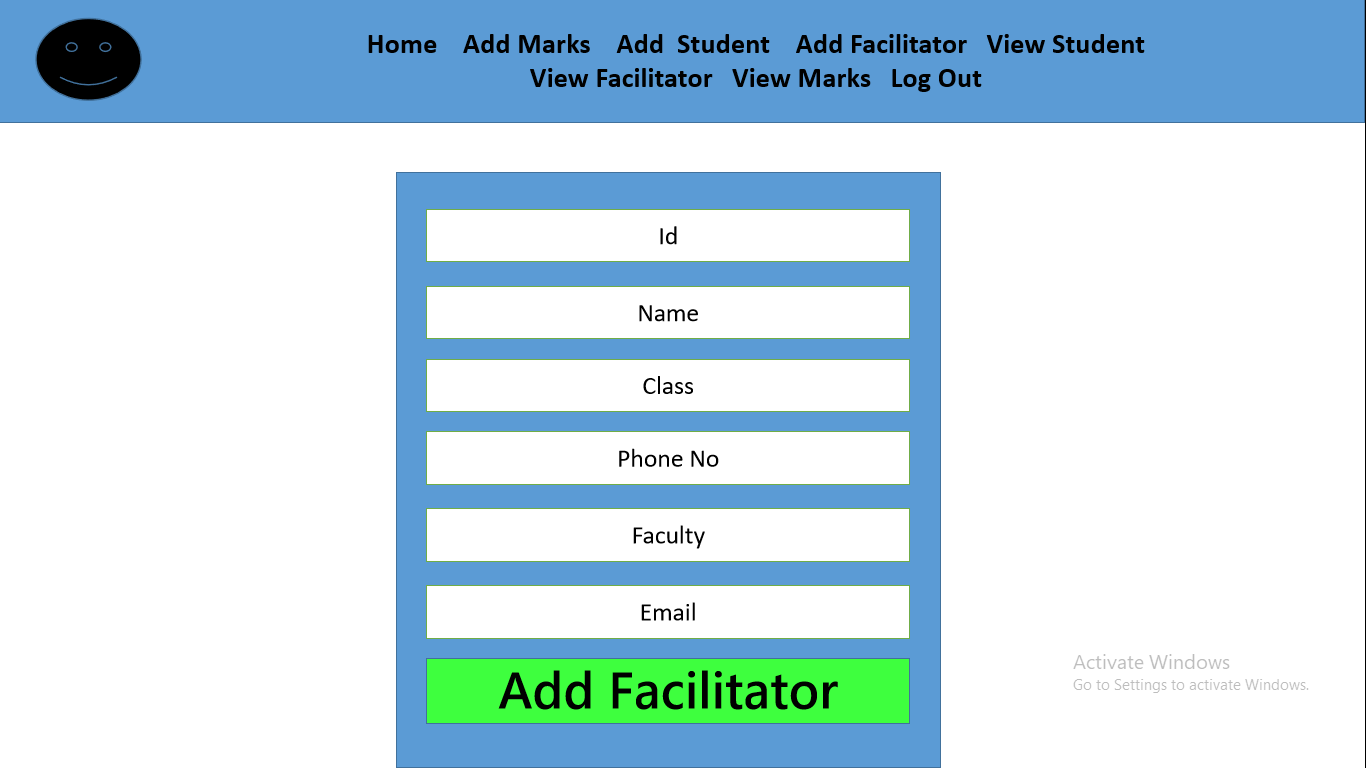


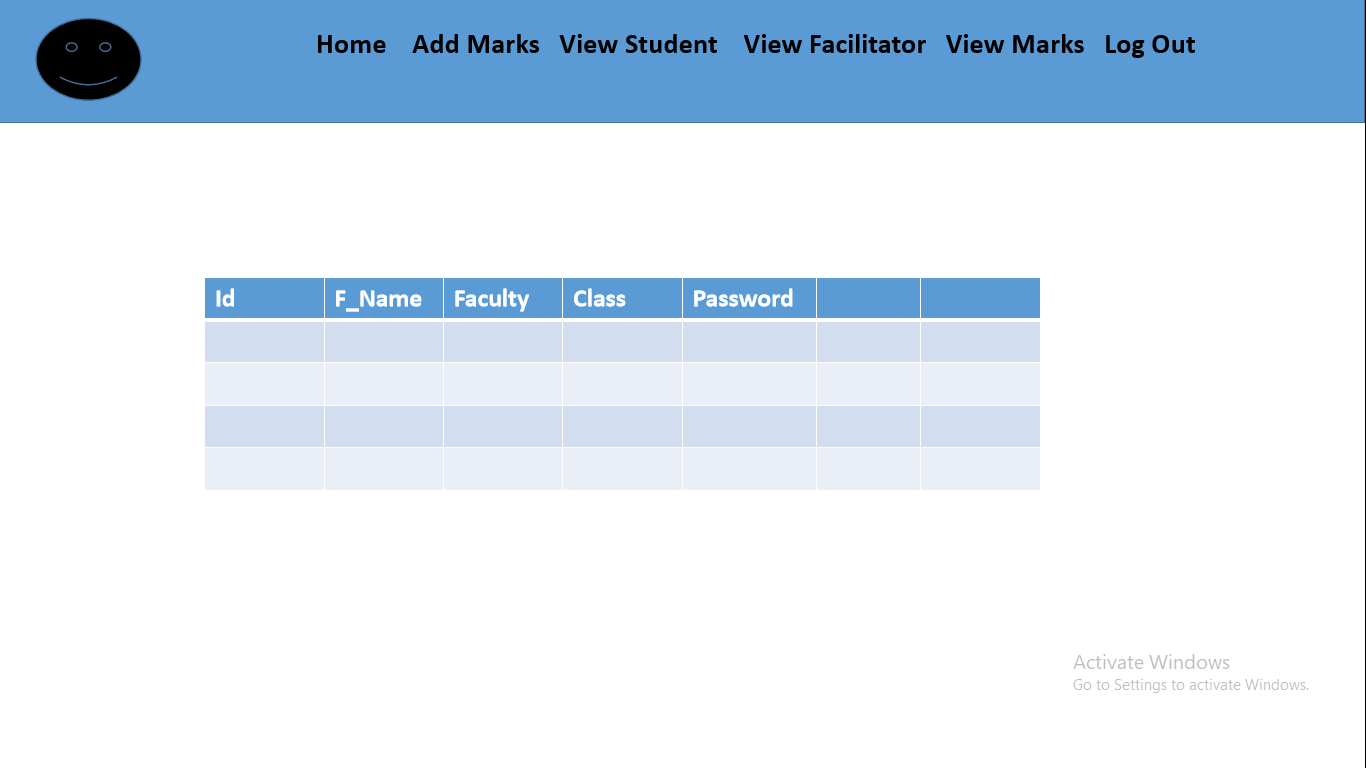
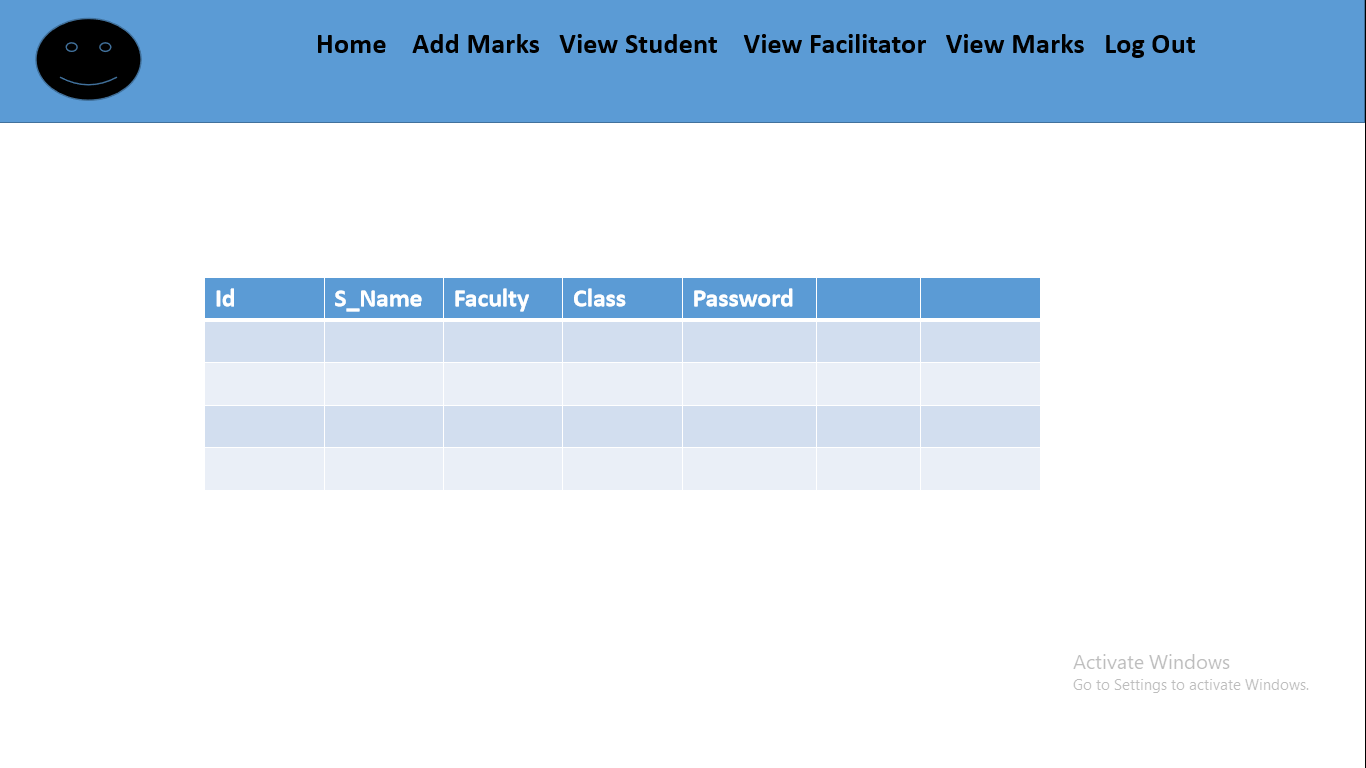
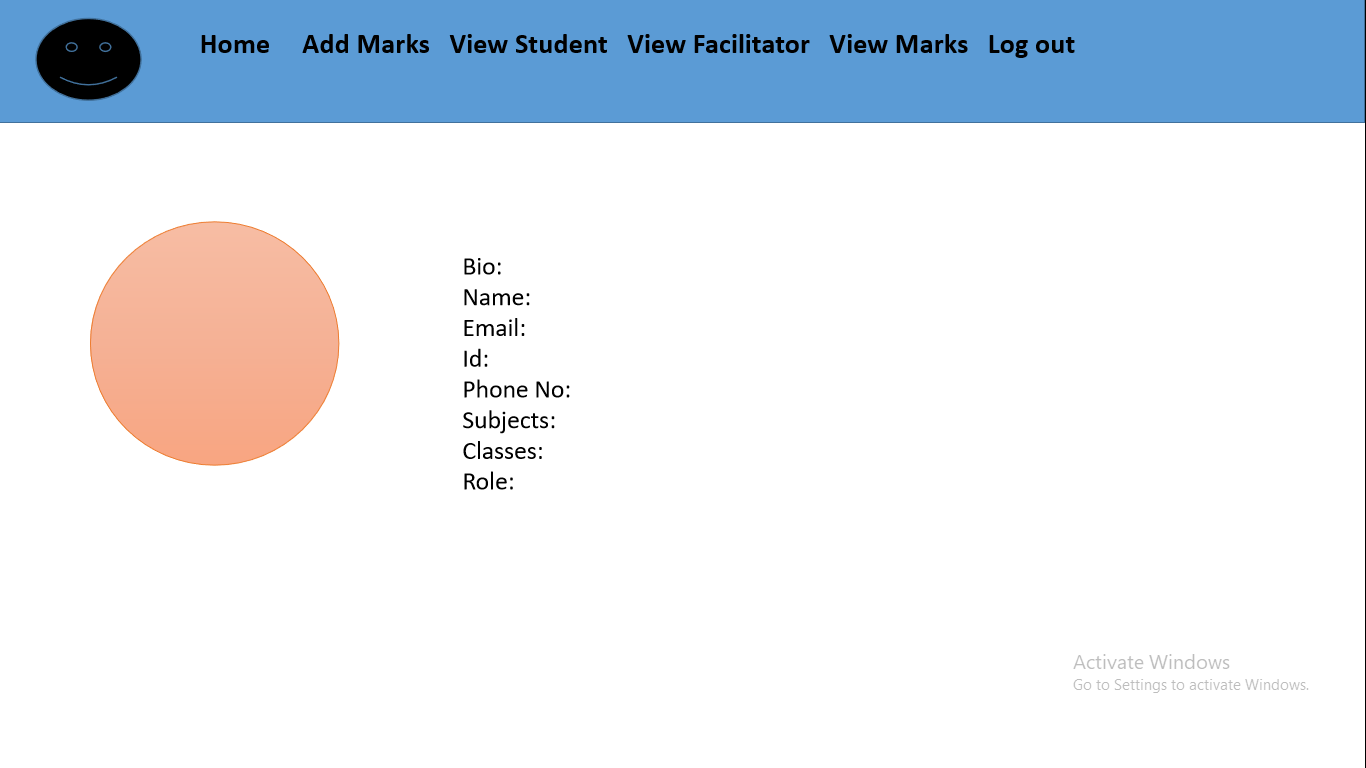
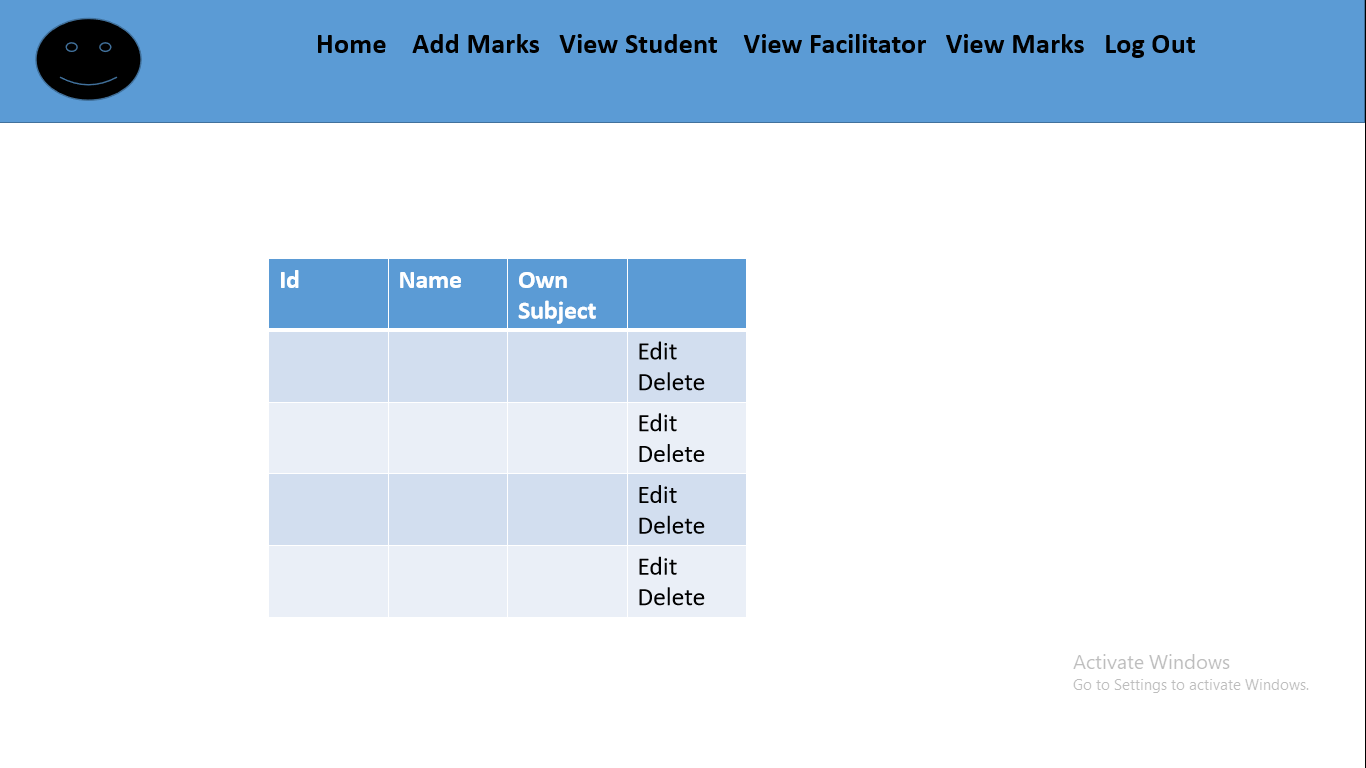
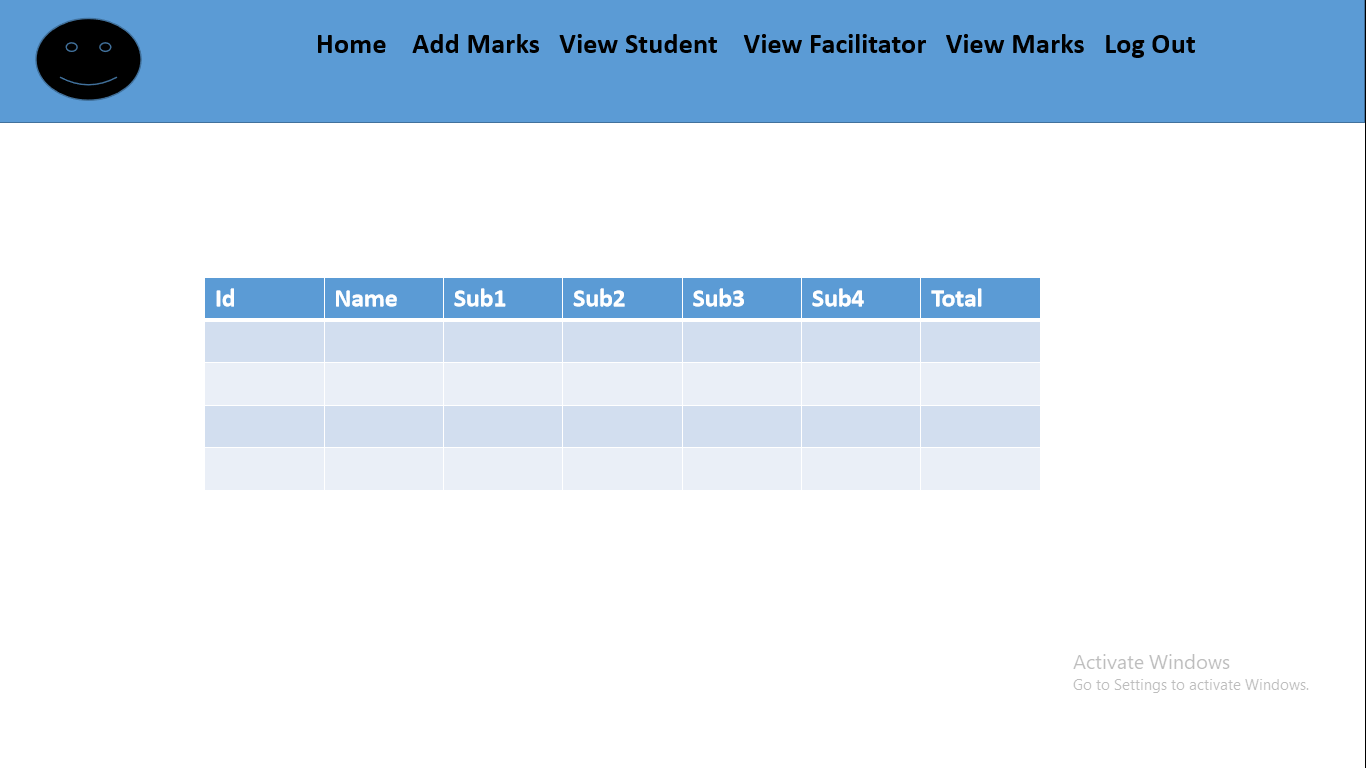
### **3.2.3 UML Activity Diagram**

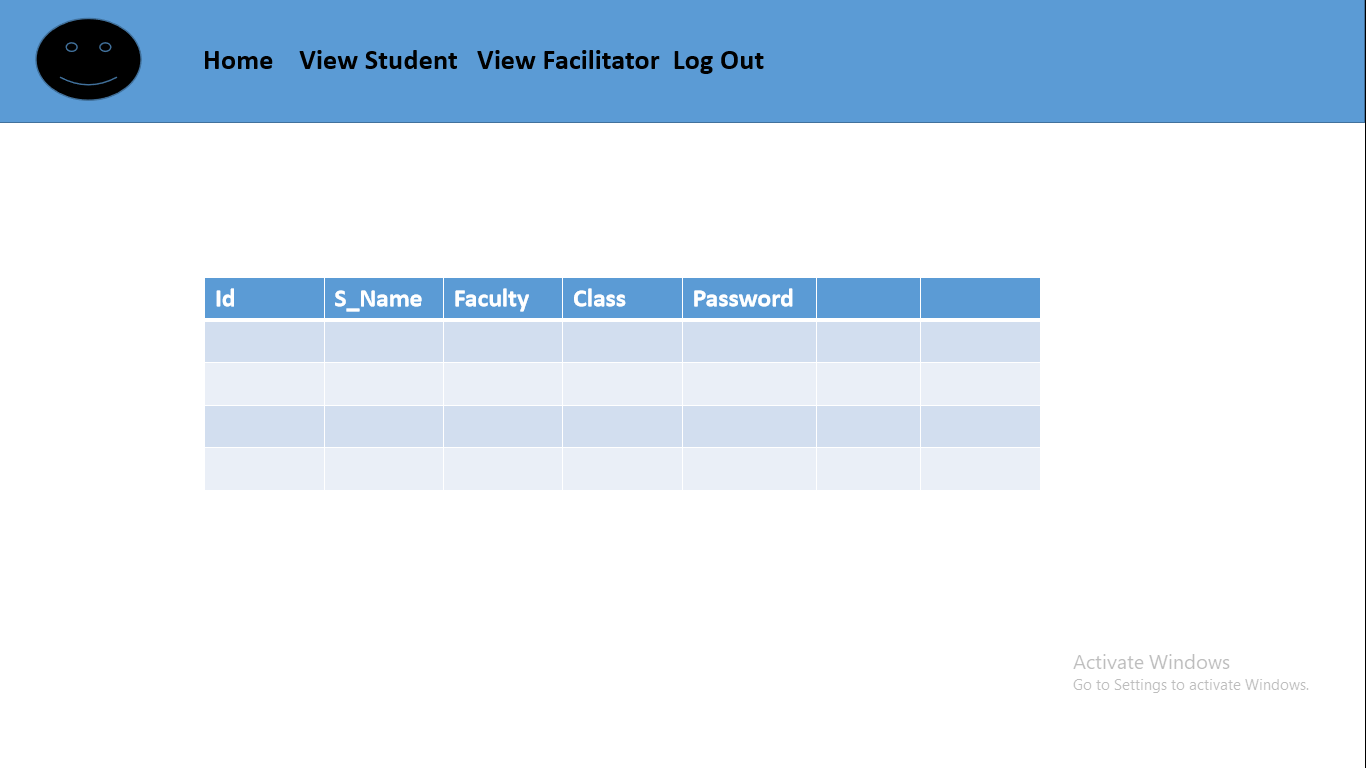
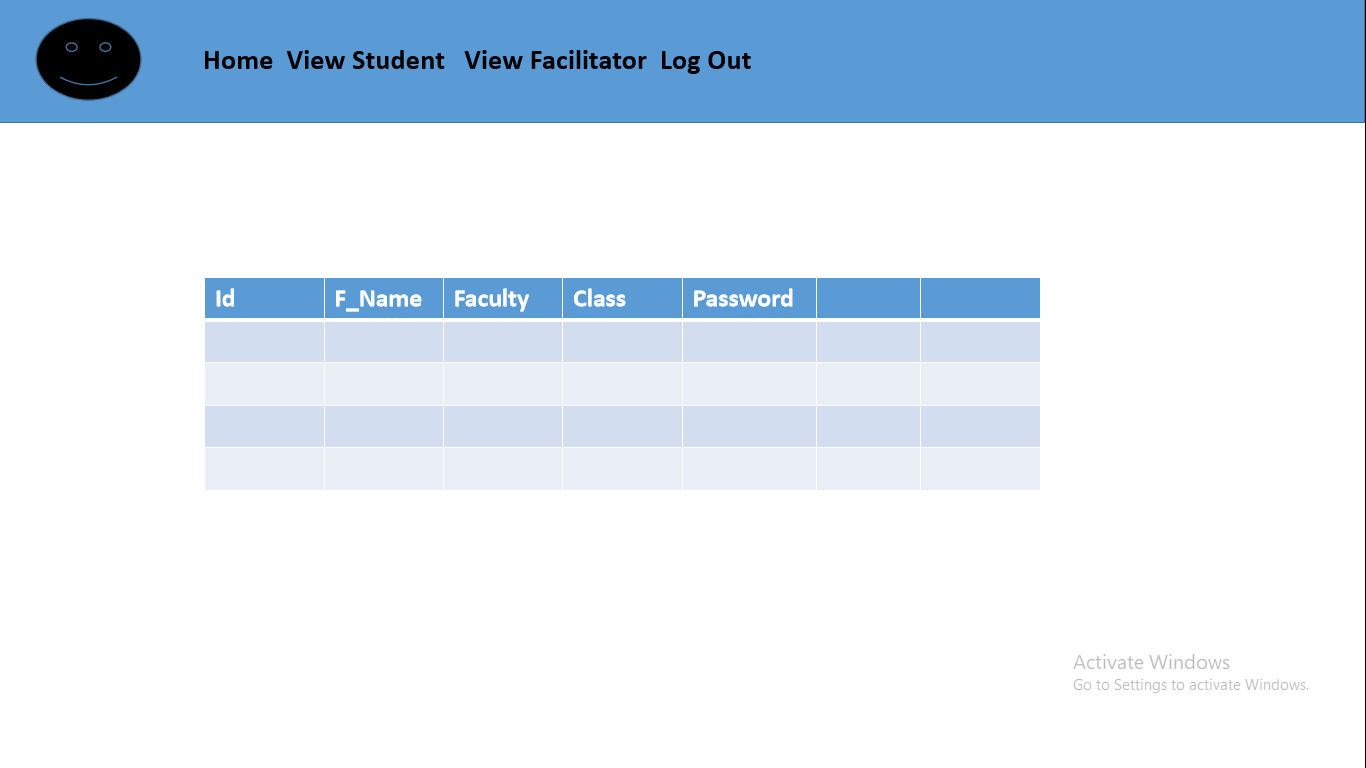
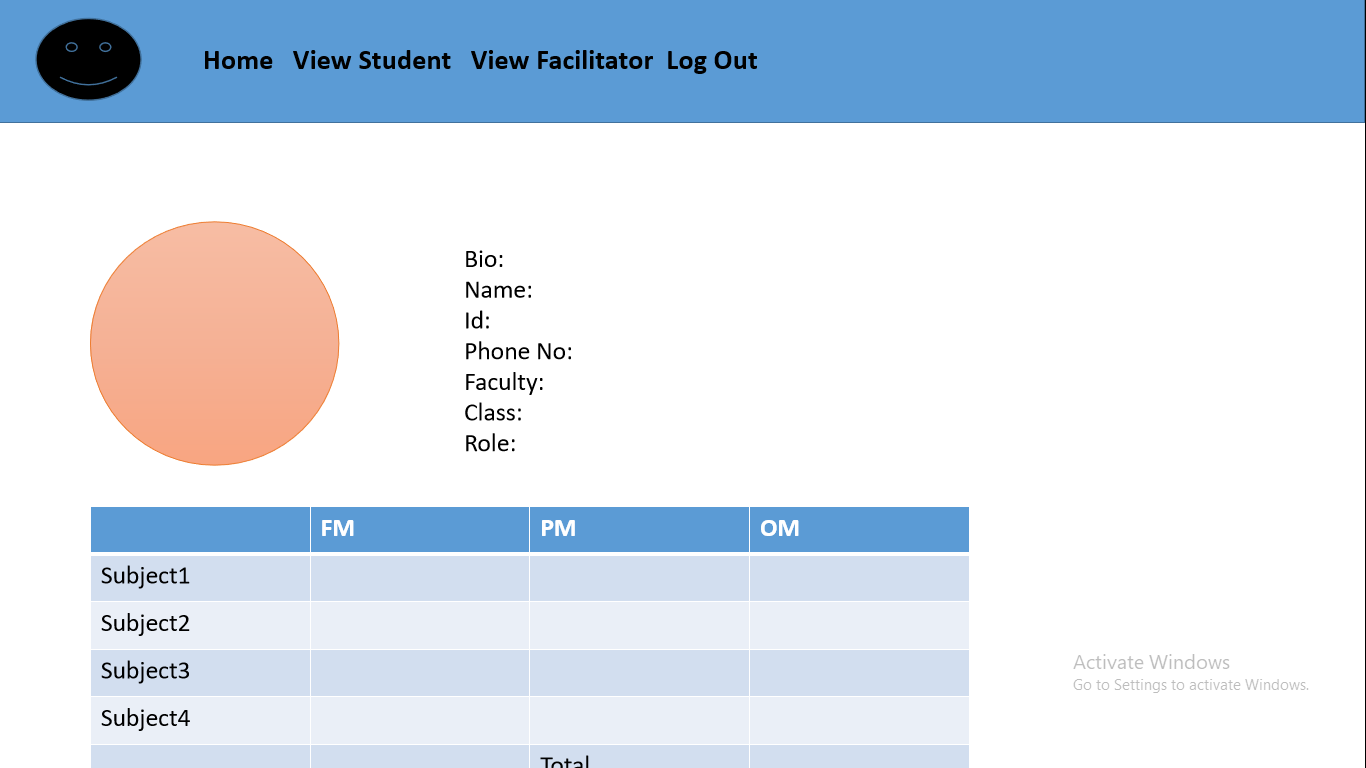


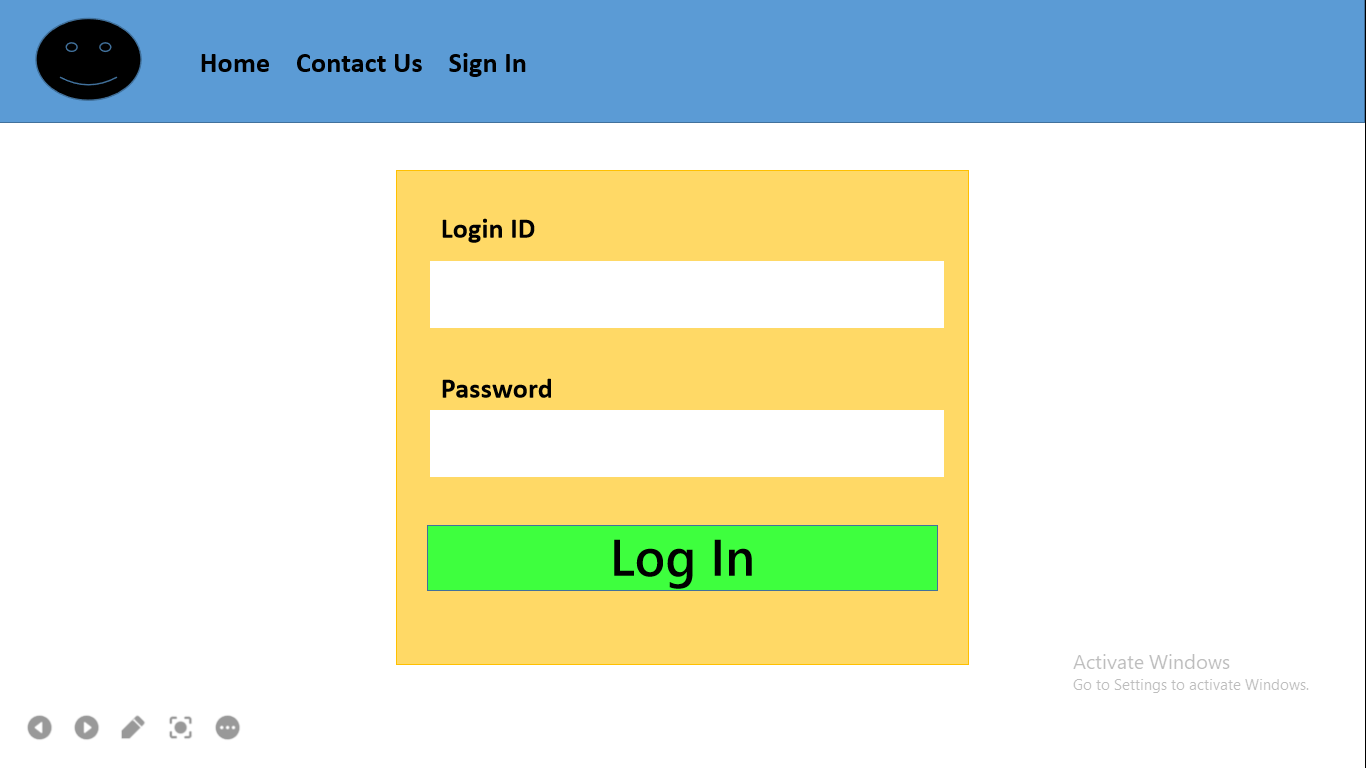
## **3.3 Interface Design**











# **CHAPTER 4 : IMPLEMENTATION AND TESTING**

## **Implementation and Overview**

Prototyping Model has been used to develop this application. The prototyping model is a technique for quickly building a function but incomplete model of the information system. There are several kind of prototyping but they all intend to reduce risk by building a quick and dirty replica or mock up of the intended system. It can be used to demonstrate technical feasibility when the technical risk is high. It can also be used to better understanding and elicit user requirements. In either case, the goal is to reduce risk and limit costs by increasing understanding of proposed solutions before committing more resources.

## **4.2 Technology used in my system**

### **4.2.1 JAVA**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. Java is used to develop mobile apps, web apps, games and much more. Java work on different platforms like Windows, Mac, Linux, Raspberry Pi, etc. It is a general purpose programming language intended to let programmers write once, run anywhere, meaning that compiled java code can run on all platforms support Java without the need to recompile.

### **4.2.2 Swing**

Swing ia graphical user interface (GUI) and a part of Oracle’s java Foundation Classes that are used to design different applications. Swing was developed to provide a more sophisticated and advanced set of GUI components than the earlier Abstract Window Toolkit. It includes New and improved Components that havee been enhancing the looks and functionality of GUI’s. Swing is more portable and more flexible that AWT, the Swing is built on top of the AWT. Swing support a pluggable look and feel and Swing provide more powerful components such as tables, list, colourchooser, etc.

### **4.2.3 JDBC**

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database.JDBC APL uses JDBC drivers to connect with the database. It is based on the X/ Open SQL Call Level interface.we can use JDBC API to handel database using Java program and can perform the activities like connection to the database, execute queries abd update statements to the database.

### **4.2.4 MySQL**

MySQL is the world’s open source database. With its proven performance, reliability and ease of use, MYSQL has become the leading database choice for web-based application, used by high profile web properties including Facebook, Twitter, YouTube,Yahoo! And many more.

## **4.3 Testing**

## **4.3.1 Integration Testing**

## **4.3.2 Unit Testing**

## **4.3.3 System Testing**

# **CHAPTER 5: CONCLUSION AND RECOMMENDATION**

## **5.1 Conclusion**

## **5.2 Recommendation**

## 

## 