1. **INTRODUCTION**
   1. **Description of Project**

We have developed E-FeedBack System to provide feedback in an easy and quick manner to the college principal and Head of Departments.

This project has five kinds of users Student, Staff, Head of Departments, Principal and Admin. The student can give feedback in this system provided by college staff. Once the Student gives the feedback, staff/lecturer can view his/her feedback result.

This feedback report was checked by the Head of Departments and Principal. Head of Department can view percentage of mark obtained by the lecturers/staffs and give this report to the principal and he give counseling to the college staff.

This system is designed to be simple to use, simple to understand and easy to implement and configure. It is provided as an Application Service Provision thus offering low setup costs.

* 1. **Purpose of the Project**

This project is developed to take feedback from students and is a window application based project. For a firm to run successfully it is compulsory to take feedback from its clients.

* 1. **Project Approach**

This section gives an outline of the way we developed the application, including the highest level milestones:

**Phase I:**

Gathering of information regarding the various technologies to be used to develop an e-FeedBack system.

**Phase II:**

We learned the technologies to be used for the developed during this phase by listening to tutorials which helped us gain knowledge about the implementation of the technologies.

**Phase III:**

Coding was next aspect and did it by using various technologies like Java and Oracle 10g Express edition.

**Phase IV:**

After running the application, we installed the application on 15-20 PC’s to check for the proper working of the functionalities that the application was aimed at.

**Phase V:**

Testing of the application was done by installing the application on college PC’s to check whether all the functionalities were being satisfied or not.

* 1. **Project Goals and objectives**

The application goals and objectives for this project will focus on implementing e-Feedback system that will do the following:

* Facilitates coordination and information sharing between both the administrators and the users.
* Provides an open, flexible, reliable technology base.
* Is easy to use.
* Ensuring that end users have input into the design of the application through the feedback forms provided.

* 1. **Scope**

As the feedback which is taken at the end of every semester, the same process is followed in this project. This project is designed to reduce the paper work, human resources and manual calculation.

* 1. **Roles and Responsibilities**

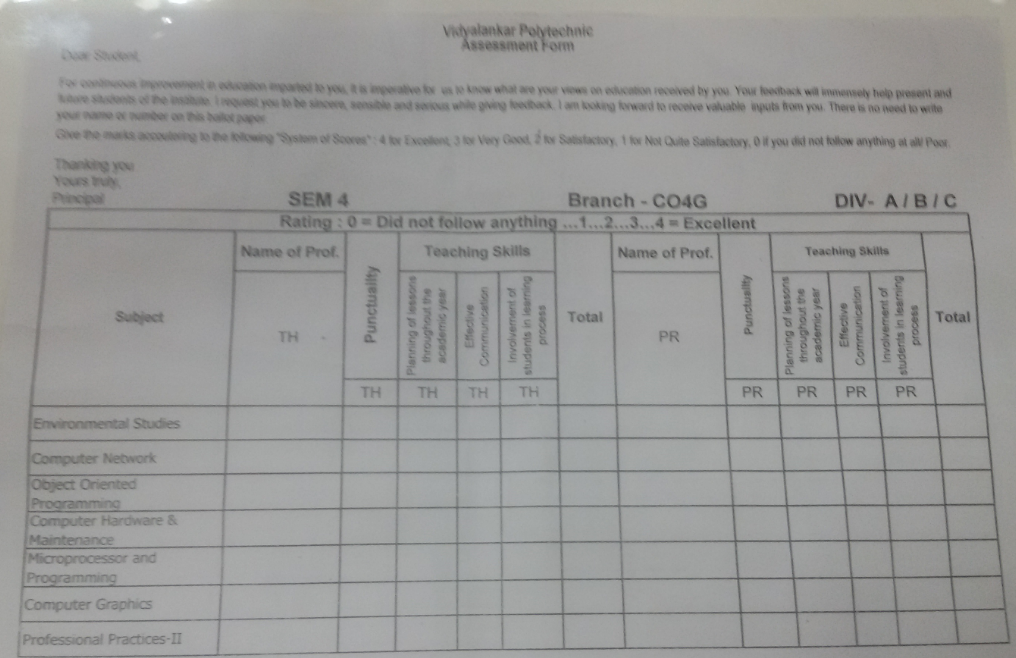
The student logs in with his unique ID and password, selects the feedback form and gives the feedback.

The administrator generates the final reports after the feedback is obtained from the students. He can generate the individual reports as well as overall reports depending on his need.

1. **REVIEW OF LITERATURE**

**2.1. EXISTING SYSTEM**

Coming to the existing system the feedback is done by manual process. In the existing system students can give feedback about the lecturers on the feedback from which is provided to students by college faculty. The students are suggested to give valid feedback based on few questions (like how punctual the lecturer is, how the lecturer plan to teach lessons, how lecturer communicate with students how lecturer involve in students). The feedback from contains the above mentioned questions and subject name. Once the feedback from is distributed amongst the student, students have to make mark his appropriate class (CO, IF or EJ) and division (A, B or C) and then they have to write the lecturer name with respect to mentioned subject name. The below diagram fig. 2.1 is the feedback form.



Student have to give the feedback in the range from 0 to 4. After the student filled the form, the person who distributed (i.e. college faculty) the form collect all the form and give it in the college office. Office member read one by one form and enter it in their database. Once all the data has been entered in the database feedback report is generated. This report is then checked by Head of Department and institute principal. After the report viewed by Head of Department and Principal a meeting is conducted in which Head of Department and Principal appreciate the lecturer who has score better feedback then others and also give advice to the lecturer whose feedback is less and motivated them with his experience.

So, the existing system is carries more time to do a piece of work for this reason we implemented “E-FeedBack System”.

**2.2. PROPOSED SYSTEM**

Here we aimed to design application based on client server architecture for issuing the feedback about the lecturers by students, this is named as E-FeedBack System. This system is to provide feedback in an easy and quick manner to the college principal and Head of Departments.

By using this system, we can make fast feedback about the staff by students.

This project has five kinds of users Student, Staff, Head of Departments, Principal and Admin. The student can give feedback on client system provided by college staff. Once the Student gives the feedback, staff/lecturer can view his/her percentage of marks by login to the system.

This feedback reports will be checked by the Head of Departments. Head of Departments can view overall percentage and view the percentage obtained to the lecturers and give this report to the principal and he can give counseling to the college staff.

As compared to the manual system, online system is very simple to use and also easy to understand.

**Highlights:**

* Saves your time
* Manage the entire process in easy and quick way
* Enhance the staff
* Improve the issuing standards

**In the proposed system generally used by five kinds of users:**

* Student
* Staff
* Head of departments
* Principal
* Admin

By using E-feedback system, we make feedback in better and quick way.

**The proposed system consists of five modules:**

* **Student:**

Student will have their unique username and password to login into the system. Once he/she login to system one java form will be displayed, which have the name of respected lecturer along with the subject they teach and the set of question (Punctuality, Planning of Lessons, Effective Communication, Involvement of Student) in tabular format. Student will fill this table with a range of number 1 to 4. Once the Student has fill the data/marks/grade he can submit this feedback by clicking on “submit” button.

* **Staff:**

Staff has its unique to login the system to view the feedback result obtained by him/her. By viewing at the feedback result he/she can improve his/her performance.

* **Head of Departments:**

In this module head of department can view feedback report/result obtained by lecturer/staff by login into the system with his/her username and password. Head of department can view the feedback report obtained by the lecturer under his/her department. Head of Department can view feedback result in 4 different ways as follows: -

1. Staff Wise Report
2. Subject Wise Report
3. Class Wise Report
4. All Staff Final Report
5. **STAFF WISE REPORT:**

In the module form, Head of Department can view the result obtained by individual staff. In this, first list of lecturer will be displayed to head of department from that he selects any one lecturer and can view overall feedback result. In this head of department can view percentage obtained by the selected lecturer in each questions (i.e. in Punctuality, Planning of Lessons, Effective Communication, Involvement of Student).

1. **SUBJECT WISE REPORT:**

As a name suggested, in this module Head of department can view subject wise feedback result. In this a list of all subject of his/her department will be displayed first. Then he/she has to select one subject. This module will display the information such as in which class which lecturer is teaching the select subject and how many marks has obtained by that lecturer.

1. **CLASS WISE REPORT**:

In this module, head of department can view class wise feedback result. Head of Department will first have to select division and semester. According to division and semester the feedback result will be displayed. Head of the Department can also export this report in excel worksheet.

1. **ALLSTAFF FINAL REPORT:**

In this module, head of department can view feedback result in ranking form. In this module, head of department can check who has is a top ranker lecturer in his/her department. Head of the Department can also export this report in excel worksheet.

* **Principal:**

In this module principal can view the feedback report in two different ways as follows: -

1. Class wise report
2. Overall staff result
3. **CLASS WISE REPORT:**

In this module principal can wise feedback result by selecting division, branch, and semester.

1. **OVERALL STAFF RESULT:**

In this module principal can view feedback result of all lecturer in his/her institute and can check who has a top rank in his institute.

* **Admin:**

Admin module will be used to manager. He will have his/her own login id and password to login into the system. This module has the following sub-module: -

1. Add Students
2. Add Staffs
3. Add Subjects
4. Add Droppers
5. Remove the login History
6. Create Hod account
7. Create Principal account
8. Increment semester
9. Delete Students
10. List of all Students who has not given the feedback
11. Allocating subject to a particular staff.
12. **ADD STUDENTS:**

In this module, admin can add n number of student account.

1. **ADD STAFFS:**

In this module, admin can add staff account.

1. **ADD SUBJECTS:**

In this module, admin can add subjects.

1. **CREATE HOD AND PRINCIPAL ACCOUNT:**

Admin can add hod account and principal account.

1. **REMOVE THE LOGIN HISTORY:**

Once the student has given the feedback his/her account is added in login history. Admin has responsibility to remove that history after the term end.

1. **INCREMENT SEMESTER:**

Admin has responsibility to increment the semester of students once the term end.

1. **DELETE STUDENT:**

This module helps to remove all third year student data.

1. **LIST OF ALL STUDENTS WHO HAS NOT GIVEN THE FEEDBACK:**

This module helps to check students list who has not given feedback.

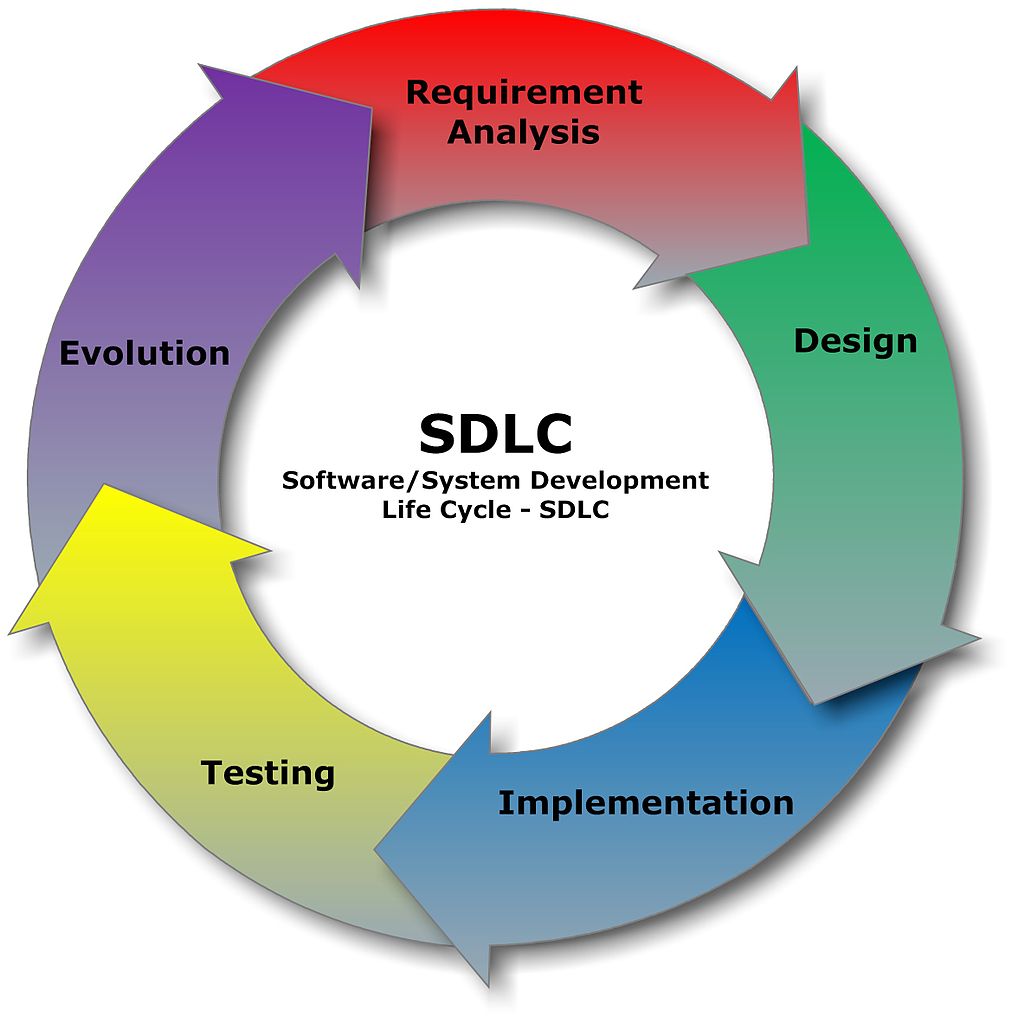
1. **ALLOCATING SUBJECT TO A PARTICULAR STAFF:**

This module helps to allocate the subject to staff.

These are the five modules present in the project so we have developed these things in very easy and clearly understandable way. So, by seen above description the proposed system having many advantages as compared to the existing system.

1. **PLAN OF WORK**

**3.1. Software Development Life Cycle**



**Planning**

Planning is an objective of each and every activity, where we want to discover things that belong to the project. An important task in creating a software program is extracting the requirement or requirement analysis. Customers typically has an abstract idea of what they want as an end result, but not what a software should do. Skilled and experience software should engineers recognize incomplete, ambiguous, or even contradictory requirements at this point.

Frequently demonstrating live code may help to reduce the risk that the requirement are incorrect. Once the general requirements are gathered from the client, an analysis of the of the development should be determined and clearly stated. This is often called a scope document. Certain functionality may be out of scope of the project as function of cost or as a result of unclear requirement at the start of the development.

**Implementation, testing and documenting**

Implementation is part of the process where software engineers program the code for the project. Software testing is an integral and important phase of the software development process. This part of the process ensures that defects are recognized as soon as possible.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development. This may also include the writing of an API, be it external or internal. The software engineering process chosen by the development team. The software engineering process chosen by the developing team will determine how much internal documentation (if any) is necessary.

**Deployment and Maintenance**

Deployment starts after the coding is appropriately tested, approved for release, and sold or otherwise distributed into a production environment. This may involve installation, customization, testing and possibly an extended period of evaluation. Software training and support is important, as software is only effective if it is used correctly. Maintaining and enhancing software to cope with newly discovered faults or requirements. Can take substantial time and efforts, as missed requirements may force redesign of the software.

**3.2. PROCESS MODELS**

There is different type of models used by a software team to do their work systematic that is step to step. The original process models have certainly given a guideline or roadmap for the whole software development process or software engineering.

The process models are properly structured so that at least structure of the process can be understood. The different process models are:

1. Waterfall model

2. Incremental Model

3. RAD Model

4.Prototype Model

5.Spiral Model

We have chosen the Waterfall Model for the development for our project.

**Waterfall Model**

The waterfall model is a sequential design process, often used in software development processes, in which is which progress is seen as flowing steadily onwards (like a waterfall) through the phases of

* Commutation
* Planning
* Modelling
* Construction
* Deployment



***Communication*** - In this requirement are collected from the customers via customer survey, magazine review, competitive product, etc.

***Planning*** - In this phase, scheduling is done so that its helps to track the progress of the project and there is no delay in delivery of the project, and the project cost is also estimated.

***Modelling*** - Here the project requirements are analyses into three domains are formed namely behavior domain, function domain and information domain and then designing is performed using

various diagram.

***Construction***- In this phase initially code is developed using the selected programming language and then it is preceded by testing to identify and fix the bugs.

***Deployment*** - This is the final stage of waterfall model and here once the software is ready it is delivered to the customer and help is provide in the form of support. After the delivery feedback is obtained.

***Advantages***

* This model is easy to understand.
* All the phases are completed one at time.
* It is systematic and sequential.
* It is used for small size software.

1. **Flow Diagram**
   1. **Flow Chart**

Flow chart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation can give a step by step solution to a given problem. Process operations are represented in these boxes, and arrows connecting them represent flow of control. Data flows are not typically represented in flowchart, in contrast with data flow diagrams; rather, they are implied by sequencing of operations. Flow charts are use in analyzing, designing, documenting and managing the process or program in various fields.

* The flowchart of our project is describing the general flow of data that is how the user is performing the appropriate gesture and how it is validated properly by the system so that each performed gesture corresponds to the respective feature of our project.
* We have constructed our flowchart in three sections (describing the three features) so that it becomes easy for us to understand the flow of our project.

C:\Users\hites\Desktop\banner.png

1. **ANALYSIS AND DESIGN**

**5.1. SYSTEM ANALYSIS**

**Systems analysis** is a [problem solving](https://en.wikipedia.org/wiki/Problem_solving) technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose". According to the [Merriam-Webster](https://en.wikipedia.org/wiki/Merriam-Webster) dictionary, systems analysis is "the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way". Analysis and synthesis, as [scientific methods](https://en.wikipedia.org/wiki/Scientific_method), always go hand in hand; they complement one another. Every synthesis is built upon the results of a preceding analysis, and every analysis requires a subsequent synthesis in order to verify and correct its results.

**5.1.1. GENERAL DESCRIPTION**

1. **PRODUCT DESCRIPTION**

**E-FEEDBACK SYSTEM** is a computerized system which will help user to take and manage the feedback about the lecturer in quick and easy manner. It reduces risk of paper such as file lost which contain final report, misunderstanding due to student handwriting and also it is time consuming. It can help the users to manage the feedback report more effectively and its save time.

1. **PROBLEM STATEMENTS**

* ***Time Consuming***- Manual system is required a lot of time rather computerized system.
* ***Paper wastages***- In manual system as feedback is taken on one piece of paper which means wastages of paper.
* ***Cost Consuming***- As it required paper to take feedback which make it costly.

1. **SYSTEM OBJECTIVE**

* ***Save Time***: User don’t need to calculate average and to generate report. Using this system, user can easily generate report with few number of clicks, which save a lot of time of user.
* ***Improvement in control and performance***: The system is developed to cope up with the current issues and problems of manual feedback system. The system can add user, validate user and is also bug free.

**5.1.2. SYSTEM RERUIREMENTS**

**5.1.2.1. PRODUCT REQUIREMENT**

* **EFFICIENCY REQUIREMENT**

When a E-FEEDBACK SYSTEM will be implemented user and college office members/staffs/principal/head of departments will easily access the required data in faster manner.

* **RELIABILITY REQUIREMENT**

The system should accurately perform member registration, member validation and report generation is various forms.

* **USABILITY REQUIREMENT**

The system is designed for a user friendly environment so that student and staff of college can perform the various tasks easily and in an effective way.

**5.1.2.2. ORGANIZATIONAL REQUIREMENT**

* **IMPLEMENTATION REQUIREMNTS**

In implementing whole system, we use java swing for front end and Oracle 10g for back end.

* **DELIVERY REQUIREMENTS**

The whole system is expected to be delivered in four months of time with a weekly evaluation by the project guide.

**5.1.3. SOFTWARE AND HARDWARE REQUIREMENTS**

This section describes the software and hardware requirements of the system

**5.1.3.1. Hardware requirements**

Our project is just a onetime investment product, which has the minimum requirements that every user already possess.

* A processing system (A laptop or computer)
* Cable
* Switch

**5.1.3.2. Software requirements**

The Software requirements of our project consisted of the following: -

* The front end of our project is coded in Java.
* We have made the use of swing components to develop the code.
* The back end of our project is coded in oracle.

**5.1.4. EXISTING VS PROPOSED SYSTEM**

* Existing system does not have any facility of teacher’s login or student login whereas proposed system will have a facility of student login as well as teacher’s login.
* In Existing system lecturer can view feedback report at any time whereas proposed system required time to generate.
* In Existing system feedback report is generated automatically whereas proposed system required office members to generate feedback report.
* In Existing system feedback of last four / five year of lecturer can be generate automatically whereas proposed system required office members to generate it.
* In Existing system maintaining the data is easy whereas it is difficult to maintain the data in proposed system.

**5.1.5. SOFTWARE TOOLS USED**

The whole Project is divided in two parts the front end and the back end.

**Front End: -**

Java Swing- **Swing** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element.

Swing is currently in the process of being replaced by JavaFX.

**Back End: -**

Oracle 10g- The **Oracle Application Server 10g** (the "g" stands for *grid*), consists of an integrated, standards-based software platform. It forms part of Oracle Corporation's Fusion Middleware technology stack. The heart of Oracle Application Server consists of Oracle HTTP Server (based on Apache HTTP Server) and OC4J (Oracle AS Containers for Java EE) which deploys Java EE-based applications. The latest version of OC4J offers full compatibility with the Java EE 1.4 specifications.

Oracle Application Server became the first platform designed for grid computing as well as with full life-cycle support for service-oriented architecture (SOA).

The current release of Oracle Application Server, 10g R3, does not feature a metadata repository tier, relying instead on metadata repositories provided in previous releases.

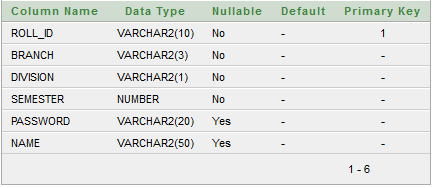
Following Oracle's acquisition of BEA Systems: “key features [will be] integrated with WebLogic Server with seamless migration”.

**5.2. SYSTEM DESIGN**

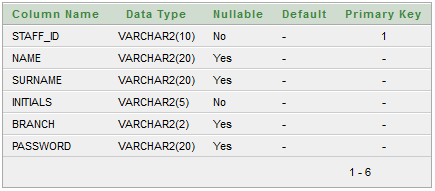
**5.2.1. TABLE DESIGN**

VARIOUS TABELS TO MAINTAIN INFORMATION

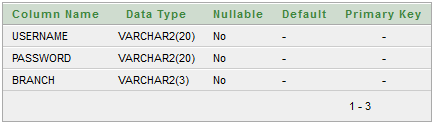
Student Table:



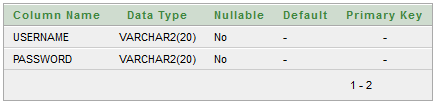
Staff Table:



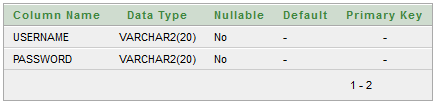
HOD Table:



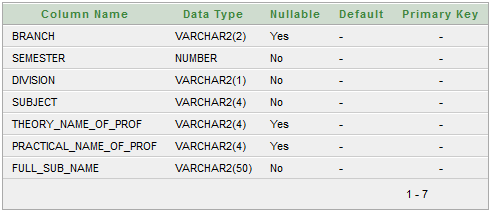
Principal Table:



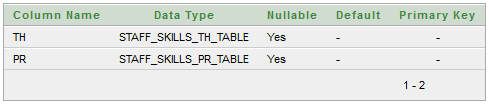
Admin Table:



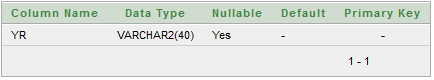
Subject Table:



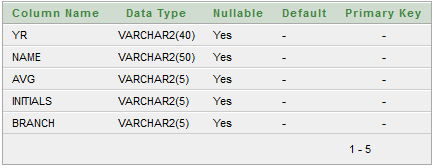
Feedback Result Table:



Years Table:



Year Wise Feedback Result Table:



**5.2.2. USE CASE DIAGRAM**

A use-case diagram at its simplest to a graphical representation of a user’s interaction with the system and depicting the specification of use case. A use case diagram can portray the different types of user of a system and the various ways that they interact with the system. They are used to model the distinct functionality of a system.

Elements of Use-case Diagram

Actor: It is external entity that interacts with the system and it is denoted by a stick man.

Use case: It specifies distinct function of a system and it is denoted by an oval.

System boundary box: It is use to collect all the use cases of a system and it is denoted by a rectangle.

Association: It is use to denote a functionality performed by an actor and it is denoted by directed line.

Use Case Diagram for Student:

C:\Users\hites\Desktop\studentusecase.png

Use Case Diagram for Staff:

C:\Users\hites\Desktop\staffusecase.png

Use Case Diagram for Head of Department:

C:\Users\hites\Desktop\HODusecase.png

Use Case Diagram for principal:

C:\Users\hites\Desktop\PRINCIPALusecase.png

Use Case Diagram for Admin:

C:\Users\hites\Desktop\ADMINusecase.png

**5.2.3. CLASS DIAGRAM**

Class diagrams are arguably the most used UML diagram type. It is the main building block of any object oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class.

In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

Below is an image of a class diagram.

C:\Users\hites\AppData\Local\Temp\diagramclass.png

**5.2.4. E-R DIAGRAM**

In software engineering, an **entity–relationship model** (**ER model**) is a [data model](https://en.wikipedia.org/wiki/Data_modeling) for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a [database](https://en.wikipedia.org/wiki/Database) such as a [relational database](https://en.wikipedia.org/wiki/Relational_database). The main components of ER models are [entities](https://en.wikipedia.org/wiki/Entities) (things) and the relationships that can exist among them.

Entity–relationship modeling was developed by [Peter Chen](https://en.wikipedia.org/wiki/Peter_Chen) and published in a 1976 paper. However, variants of the idea existed previously, and have been devised subsequently such as supertype and subtype data entities and commonality relationships.

The first stage of [information system](https://en.wikipedia.org/wiki/Information_system) design uses these models during the [requirements analysis](https://en.wikipedia.org/wiki/Requirements_analysis) to describe information needs or the type of [information](https://en.wikipedia.org/wiki/Information) that is to be stored in a [database](https://en.wikipedia.org/wiki/Database). The [data modeling](https://en.wikipedia.org/wiki/Data_modeling) technique can be used to describe any [ontology](https://en.wikipedia.org/wiki/Ontology_%28computer_science%29) (i.e. an overview and classifications of used terms and their relationships) for a certain [area of interest](https://en.wikipedia.org/wiki/Universe_of_discourse). In the case of the design of an information system that is based on a database, the [conceptual data model](https://en.wikipedia.org/wiki/Conceptual_data_model) is, at a later stage (usually called logical design), mapped to a [logical data model](https://en.wikipedia.org/wiki/Logical_data_model), such as the [relational model](https://en.wikipedia.org/wiki/Relational_model); this in turn is mapped to a physical model during physical design. Note that sometimes, both of these phases are referred to as "physical design."

## **Entity**

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

## **Attributes**

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).

C:\Users\hites\Desktop\er.png

1. **PROJECT IMPLEMENTATION**

NetBeans.svg

**NetBeans** is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

**NetBeans Platform**

The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

* User interface management (e.g. menus and toolbars)
* User settings management
* Storage management (saving and loading any kind of data)
* Window management
* Wizard framework (supports step-by-step dialogs)
* NetBeans Visual Library
* Integrated development tools

NetBeans IDE is a free, open-source, cross-platform IDE with built-in-support for Java Programming Language.

## **NetBeans IDE**

**NetBeans IDE** is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactoring’s, version control (supporting CVS, Subversion, Git, Mercurial and Clearcase).

### **GUI design tool**

Formerly known as *project Matisse*, the GUI design-tool enables developers to prototype and design [Swing](https://en.wikipedia.org/wiki/Swing_%28Java%29) [GUIs](https://en.wikipedia.org/wiki/Graphical_user_interface) by dragging and positioning GUI components.

The GUI builder has built-in support for JSR 295 (Beans Binding technology), but the support for JSR 296 ([Swing Application Framework](https://en.wikipedia.org/wiki/Swing_Application_Framework)) was removed in 7.1.

### **NetBeans JavaScript editor**

The NetBeans JavaScript editor provides extended support for [JavaScript](https://en.wikipedia.org/wiki/JavaScript), Ajax, and [CSS](https://en.wikipedia.org/wiki/CSS).

JavaScript editor features comprise [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [refactoring](https://en.wikipedia.org/wiki/Code_refactoring), [code completion](https://en.wikipedia.org/wiki/Autocomplete) for native objects and functions, generation of JavaScript class skeletons, generation of [Ajax](https://en.wikipedia.org/wiki/Ajax_%28programming%29) [callbacks](https://en.wikipedia.org/wiki/Callback_%28computer_science%29) from a template; and automatic [browser compatibility](https://en.wikipedia.org/wiki/JavaScript#Compatibility_considerations) checks.

CSS editor features comprise [code completion](https://en.wikipedia.org/wiki/Autocomplete) for styles names, quick navigation through the navigator panel, displaying the CSS rule declaration in a List View and file structure in a Tree View, sorting the outline view by name, type or declaration order (List & Tree), creating rule declarations (Tree only), refactoring a part of a rule name (Tree only).

The NetBeans 7.4 and later uses the new [Nashorn] JavaScript engine developed by Oracle.

Oracle logo.svg

## Oracle Database Architecture

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a [server](https://docs.oracle.com/cd/B19306_01/server.102/b14220/glossary.htm#i432724) reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

QL and PL/SQL form the core of Oracle's application development stack. Not only do most enterprise back-ends run SQL, but Web applications accessing databases do so using SQL (wrapper by Java classes as JDBC), Enterprise Application Integration applications generate XML from SQL queries, and content-repositories are built on top of SQL tables. It is a simple, widely understood, unified data model. It is used standalone in many applications, but it is also invoked directly from Java (JDBC), Oracle Call Interface (OCI), Oracle C++ Call Interface (OCCI), or XSU (XML SQL Utility). Stored packages, procedures, and triggers can all be written in PL/SQL or in Java.

1. **TESTING METHODS**

**7.1. THE BOX APPROACH:**

Software testing methods are traditionally divided into white box and black box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

1. **WHITE - BOX TESTING:**

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural box testing) is a method of testing software that tests internal structure or workings of an application. In white – box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). While white – box testing can be applied at the unit level, integration and system levels of the software testing process, it is usually done at unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, this might not detect unimplemented parts of the specification or missing requirements.

* **White – box test design techniques include:**

1. Control flow testing.
2. Data flow testing.
3. Branch Testing.
4. Path Testing.
5. Statement coverage.
6. Decision coverage.

* **Overview:**

White – box testing is a method of testing the application at the level of the source code. The lost cases are derived through the use of the design techniques mentioned above: control flow testing, data flow testing, branch testing, path testing, statement coverage, decision coverage. White – box testing is the use of these techniques as the guidelines to create an error free environment by examining any fragile code. These white – box testing techniques are the building blocks of white – box testing, whose essence is the careful testing of the application at the source code level to prevent any hidden errors later on. These different techniques exercise over visible path of the source code to minimize errors and create an error-free environment. The whole point of white – box testing is the ability to know which line of code is being executed and being able to identify what the correct output should be.

1. **BLACK – BOX TESTING:**

Black – box testing is a method of testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white – box testing). This method of test can be applied to virtually every level of software: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well. Specific knowledge of the application’s code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it.

Typical black – box test design techniques include:

1. Decision table.
2. State transaction table.
3. Equivalence partitioning.
4. Boundary value analysis.
5. **GRAY – BOX TESTING:**

Gray – box testing (International English spelling: grey-box testing) is a combination of white – box testing and black – box testing. The aim of this testing is to search for the defects if any due to improper structure or improper usage of applications. Gray –box testing is also known as translucent testing. A black – box tester is unaware of the internal structure of the application. A gray – box tester partially knows the internal structure, which includes access to the documentation of internal data structures as well as the algorithms used. Gray – box testers require both high level and detailed documents describing the application, which they collect in order to define the test cases.

**7.2. TESTING LEVELS**

* Stub test:

A test performed on a subset of a program.

1. Individual events or modules of a program are tested.
2. Testing of an isolated subset of a program is carried out.

* Unit or program test:

A test performed on an entire program.

1. All the events and modules are tested as an integrated unit.

* System test:

A test performed on entire system.

1. This test ensures that the application programs written and tested in isolation work properly when integrated into the total system.

* System acceptance test:

A test performed on the financial system wherein users conduct verification, validation and audit test.

1. Here we use real data over an extended time period.
2. Extensive test that addresses: verification testing, validation testing and audit testing.

* Verification testing:

It runs the system in a simulated environment using simulated data.

1. Alpha testing.
2. Simulated environment using simulated testing.
3. Checks for errors and omissions regarding end-use and design specifications.

* Validation testing:

1. Runs the system in a live environment using real data.
2. Beta testing.
3. Live environment using real data.

* System performance: (throughput & response time)

It determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.

* Peak workload performance:

It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example: Test the system by activating all terminals at the same time.

* Human engineering:

It determines how users will use the system when processing data or preparing reports.

* Methods and procedure:

It determines the clarity of documentation on operation and use of system by having users do exactly what manuals request. For example: Powering down system at the end of the week or responding to paper-out light on printer.

* Audit testing:

Certifies that the system is free of errors and is ready to be placed into operation.

**7.3. TEST CASES**

A test cases in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites. A test case is usually a single step, or occasionally a sequence of steps, to test the correct behavior/functionality, features of an application. An expected result or expected outcome is usually given.

Test Cases for student:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name | | E-FEEDBACK SYSTEM | | |
| Test ID | | 1 | | |
| Test Module | | Student Login | | |
| Testing Method | | Unit Testing | | |
| Description of Module | | Login should be signing in by entering his/her username and password. | | |
| Sr. No. | Input | Expected Results | Actual Results | Status |
| 1.1 | Username= “13203b0044”  Password= “13203b0044”  (correct username and password) | It must accept username and password and display the feedback form. | It is accepting username and password and displaying the feedback form. | Pass |
| 1.2 | Username= “13203b0044”  Password= ““  (right username and empty password field) | It must display message that You Have Not Enter Your Password. Please Enter your password” | Message is displayed “You Have Not Enter Your Password. Please Enter your password” | Pass |
| 1.3 | Username= ”1233221”  Password=  ”13203b0044”  (wrong username and right password) | It must display message that “your username or password is invalid” | Message id displayed “Your username or password is invalid” | Pass |

Test Case for Staff:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name | | E-FEEDBACK SYSTEM | | |
| Test ID | | 2 | | |
| Test Module | | Staff Login | | |
| Testing Method | | Unit Testing | | |
| Description of Module | | Login should be signing in by entering his/her username and password. | | |
| Sr. No. | Input | Expected Results | Actual Results | Status |
| 2.1 | Username= “VP0003”  Password= “VP0003”  (correct username and password) | It must accept username and password and display the next form. | It is accepting username and password and displaying the next form. | Pass |
| 2.2 | Username= “VP0003”  Password= ““  (right username and empty password field) | It must display message that You Have Not Enter Your Password. Please Enter your password” | Message is displayed “You Have Not Enter Your Password. Please Enter your password” | Pass |
| 2.3 | Username= ”123322”  Password=  ” VP0003”  (wrong username and right password) | It must display message that “your username or password is invalid” | Message id displayed “Your username or password is invalid” | Pass |

Test Case for HOD:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name | | E-FEEDBACK SYSTEM | | |
| Test ID | | 3 | | |
| Test Module | | HOD Login | | |
| Testing Method | | Unit Testing | | |
| Description of Module | | Login should be signing in by entering his/her username and password. | | |
| Sr. No. | Input | Expected Results | Actual Results | Status |
| 3.1 | Username= “co”  Password= “co”  (correct username and password) | It must accept username and password and display the next form. | It is accepting username and password and displaying the next form. | Pass |
| 3.2 | Username= “co”  Password= ““  (right username and empty password field) | It must display message that You Have Not Enter Your Password. Please Enter your password” | Message is displayed “You Have Not Enter Your Password. Please Enter your password” | Pass |
| 3.3 | Username= ”co”  Password=  ” admin”  (wrong username and right password) | It must display message that “your username or password is invalid” | Message id displayed “Your username or password is invalid” | Pass |

Test Case for principal:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name | | E-FEEDBACK SYSTEM | | |
| Test ID | | 4 | | |
| Test Module | | Principal Login | | |
| Testing Method | | Unit Testing | | |
| Description of Module | | Login should be signing in by entering his/her username and password. | | |
| Sr. No. | Input | Expected Results | Actual Results | Status |
| 4.1 | Username= “admin”  Password= “admin”  (correct username and password) | It must accept username and password and display the next form. | It is accepting username and password and displaying the next form. | Pass |
| 4.2 | Username= “admin”  Password= ““  (right username and empty password field) | It must display message that You Have Not Enter Your Password. Please Enter your password” | Message is displayed “You Have Not Enter Your Password. Please Enter your password” | Pass |
| 4.3 | Username= ” administration”  Password=  ” admin”  (wrong username and right password) | It must display message that “your username or password is invalid” | Message id displayed “Your username or password is invalid” | Pass |

Test Case for admin:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name | | E-FEEDBACK SYSTEM | | |
| Test ID | | 5 | | |
| Test Module | | Admin Login | | |
| Testing Method | | Unit Testing | | |
| Description of Module | | Login should be signing in by entering his/her username and password. | | |
| Sr. No. | Input | Expected Results | Actual Results | Status |
| 5.1 | Username= “admin”  Password= “admin”  (correct username and password) | It must accept username and password and display the next form. | It is accepting username and password and displaying the next form. | Pass |
| 5.2 | Username= “admin”  Password= ““  (right username and empty password field) | It must display message that You Have Not Enter Your Password. Please Enter your password” | Message is displayed “You Have Not Enter Your Password. Please Enter your password” | Pass |
| 5.3 | Username=” administration”  Password=  ” admin”  (wrong username and right password) | It must display message that “your username or password is invalid” | Message id displayed “Your username or password is invalid” | Pass |

1. **FEASIBILITY ANALYSIS**

**8.1. Feasibility Assessment**

Feasibility study is a test of system proposal according to the workability, impact on the organization, ability to meet user needs and effective use of the available resources. The objective of feasibility study is not to solve the problem but to acquire a sense of its scope.

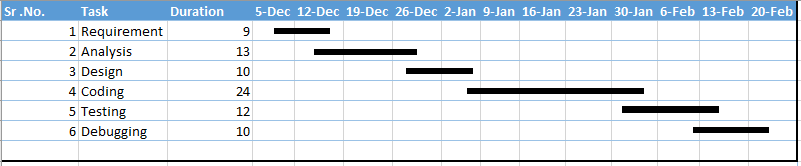
**8.1.1. Time Feasibility:**

Time Feasibility corresponds to sufficient time is available to complete the project.

Parameters considered:

1. Schedule of Project
2. Time by which the project has to be completed.
3. Reporting period.
4. Considering all the above factors it was decided that the allotted time that in 3 months was sufficient to complete the project.

**Gantt Chart:**



**8.1.2. ECONOMICS FEASIBILITY:**

Economic analysis is the most frequently used method for evaluating the effectiveness of a client system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expecting from a client system and compare them with cost.

**8.1.3. TECHNICAL FEASIBILITY:**

Technical feasibility centers on the existing system and to what extent it can support the proposed system. The benefits such as high accuracy, minimum response time and user friendliness of the proposed system over weights cost for designing and implementing the new system.

**8.1.4. SCHEDULE FEASIBILITY:**

The time schedule required for the development of the project is important, since more development time affects machine time and cost of delay in the development of other systems.

1. **PROS AND CONS**

**9.1. ADVANTAGES:**

* Save your time – by giving feedback on online system when compared to the manual process this saves time of user.
* Manage the entire process in easy and quick way – the entire process of giving feedback and viewing that report after giving feedback can manage easily The result is generated is automatically hence it gives easy and quick feedback.
* Enhance the staff – find the details about the lecturer’s interest in teaching to the students.
* Students should be told how the data will be used, when and how results will be published, and that their contribution is important and taken seriously;
* Questionnaires should be distributed and collected (anonymously) in class, ideally in the penultimate week of the course, with the aim of achieving a response rate of at least 75% on each course;
* The Head of Department should discuss a dispassionate summary of the feedback on each course with the lecturers concerned;
* An analysis of questionnaire returns which concern the department as a whole should be published within the department, and referred to as appropriate in the Annual Review report.

**9.2. DISADVANTAGES:**

* If the system crashes, then you cannot give your feedback.

1. **FUTURE DEVELOPMENT**

As the number of institutions are following the process of taking the feedback from its clients in order to know how best their service is. In this context our application can be used.

So this means that our project can found its use in many customer services organizations that may include:

* Any Private Business Organization
* IT Firm
* Government Firm
* It can also be used in Online Shopping World allowing the users who are registered with their site to give their feedback regarding the quality of services provided to them.

Many Organizations take feedback from their clients on a paper and later manually calculate the ratings individually. This consumes more time and there are many possibilities of committing mistakes. Hence to overcome all these difficulties they require an automated application which calculates the ratings at the background and displays only the result in their required format. For this purpose, our application can be used.

1. **CONCLUSION**

The Project “E-FeedBack” system is designed in order reduce the burden of maintaining bulk of records of all the student’s feedback details of who study in an Educational Institution. Inserting, retrieving and updating the feedback details of a student are easy when it is compared to the manual feedback and storing. Maintaining the project is also easy which can be easily understandable. Maintaining the details in the database is manageable.

1. **Appendix**

**Screens:**

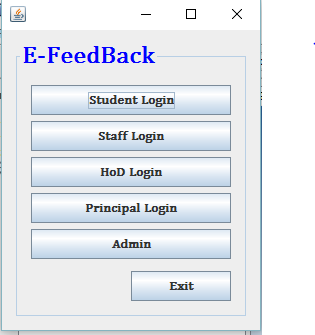


figure of first page

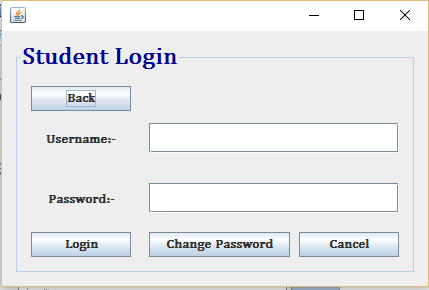


Figure of student login page

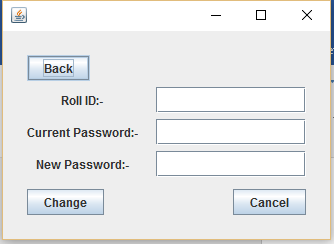


figure when student click on change password

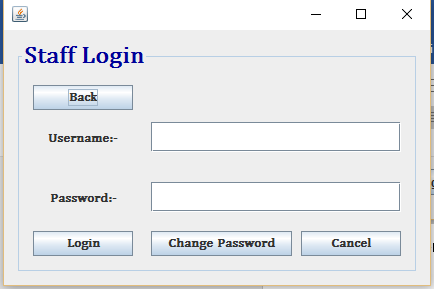


figure of staff login page

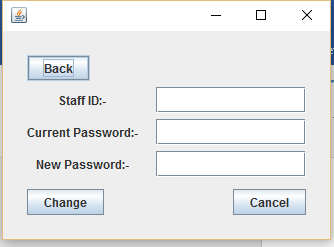


Figure when staff click on change password

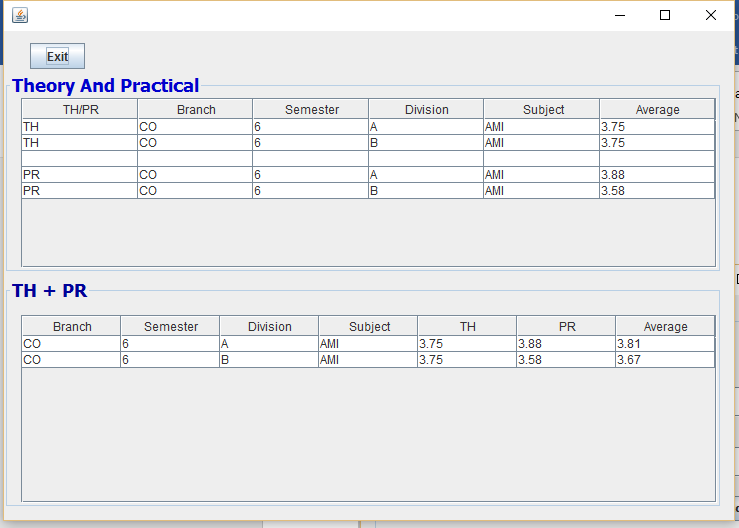


Figure when staff login into the system.

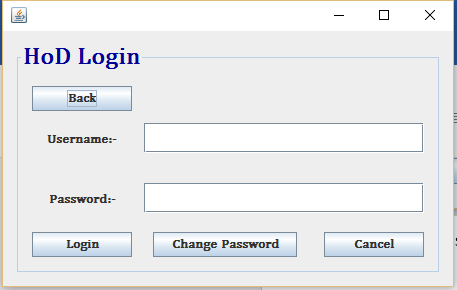


figure of Head of Department login page

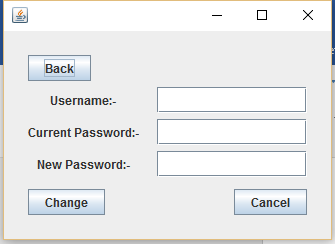


Figure when Head of Department click on change password

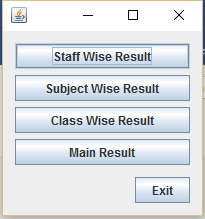


Figure when Head of Department login into the system

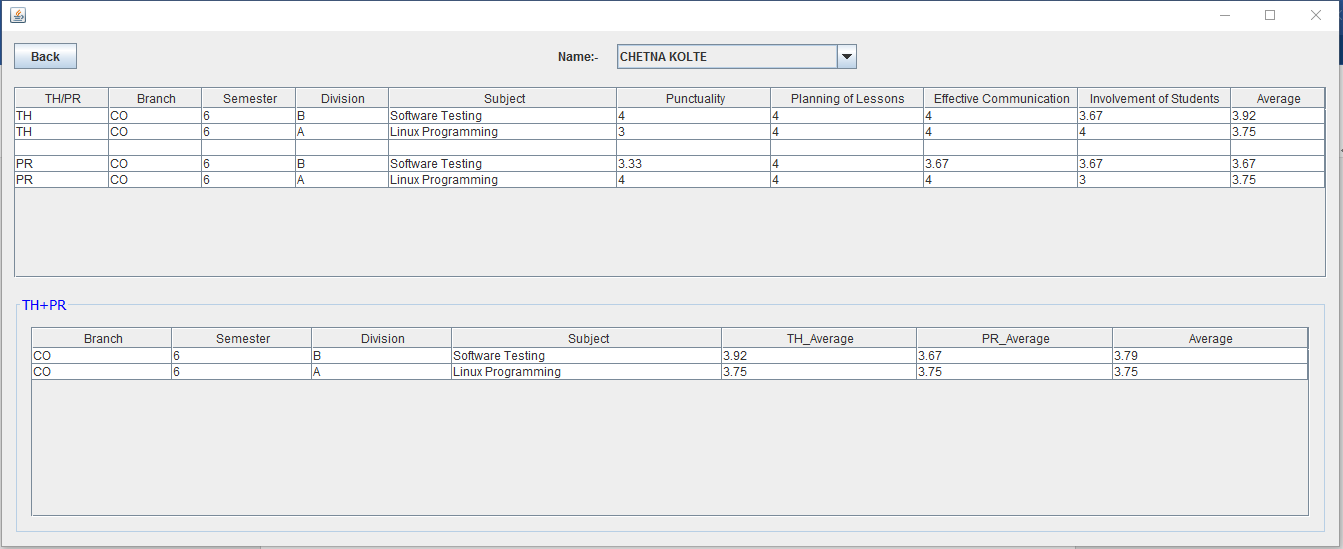


Figure when head of department clicks on staff wise result

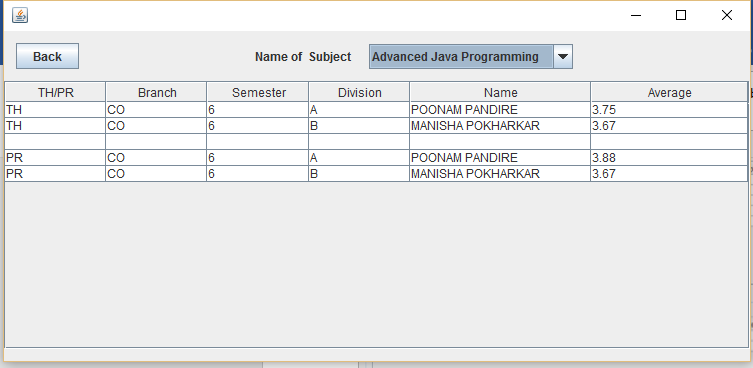


Figure when head of department clicks on subject wise result.

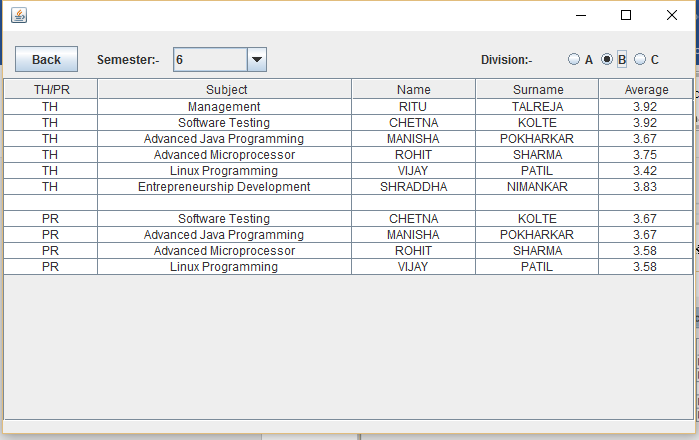


Figure when head of department clicks on class wise result

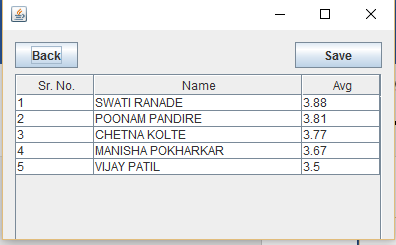


Figure when head of department clicks on main result

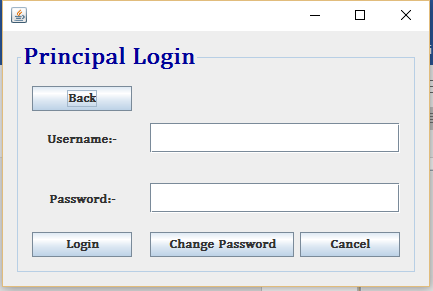


Figure of principal login page

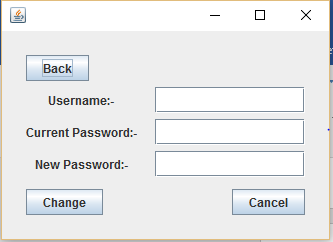


Figure when principal click on change password

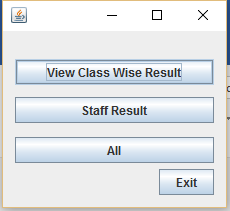


Figure when principal login into the system

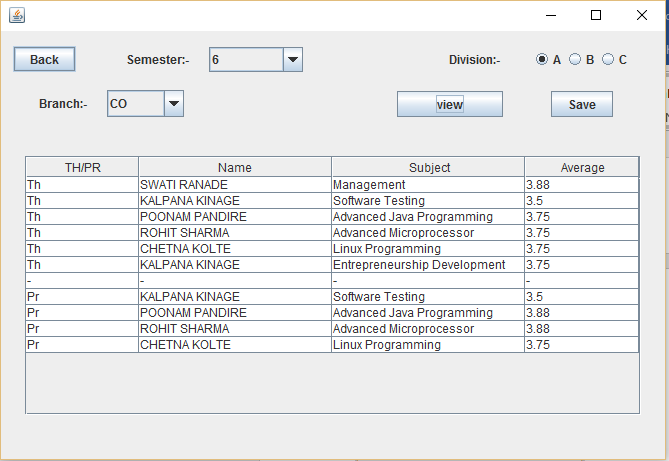


Figure when principal clicks on view class wise result

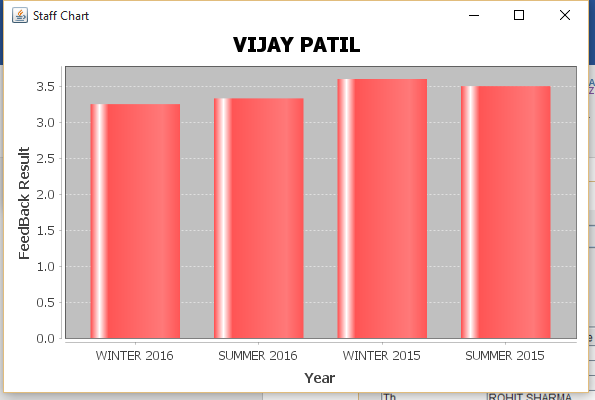


Figure when principal clicks on staff result

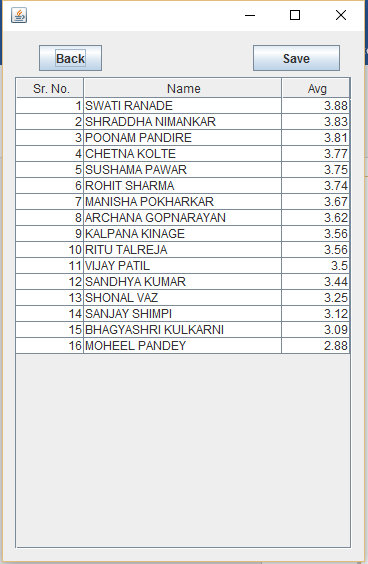


Figure when principal clicks on all

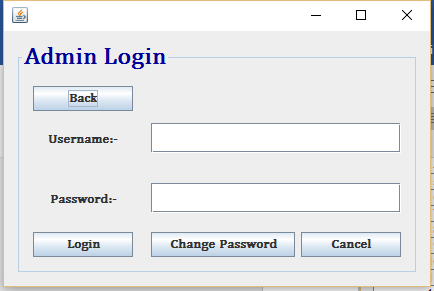


Figure of admin login page

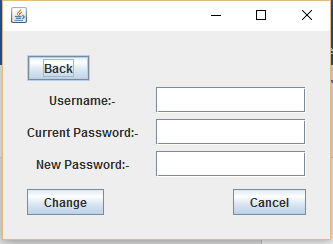


Figure when admin clicks to change password

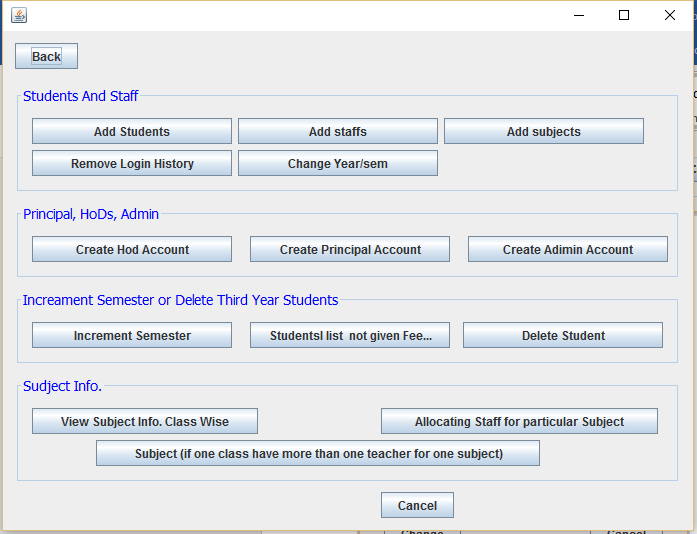


Figure when admin login into the system

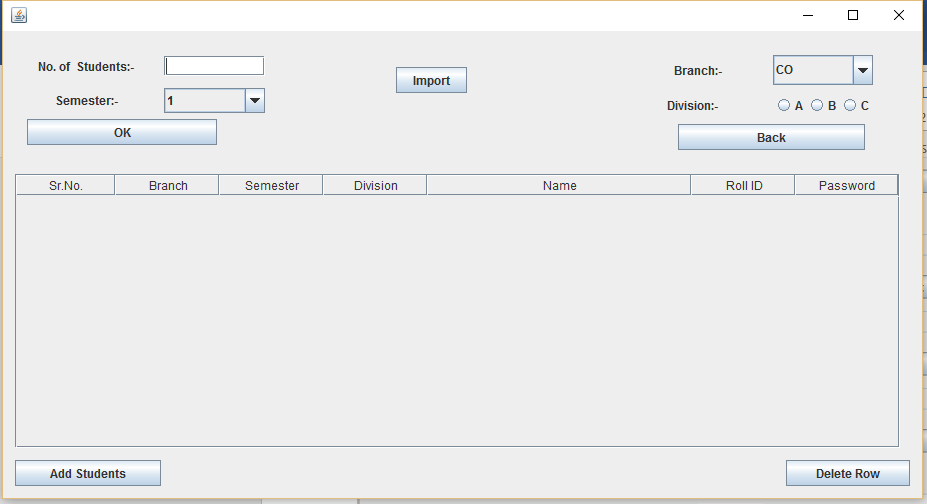


Figure when admin clicks on add students

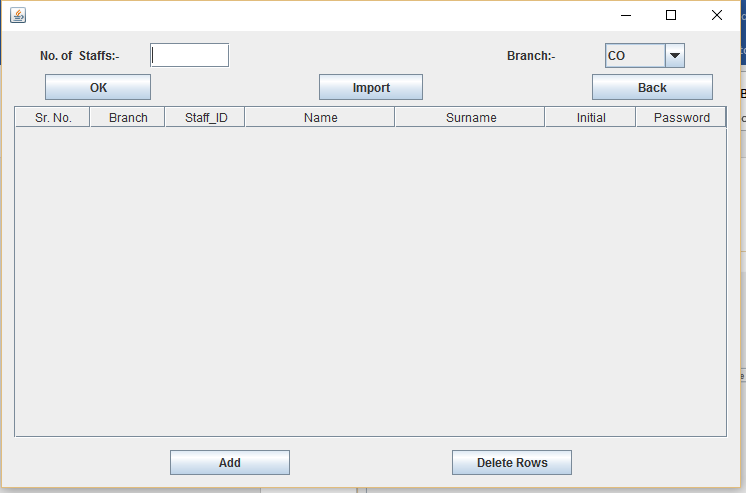


Figure when admin clicks on add staffs

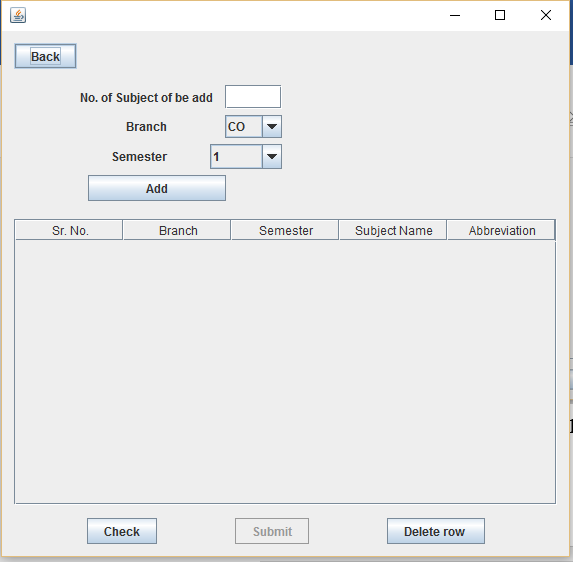


Figure when admin clicks on add subject

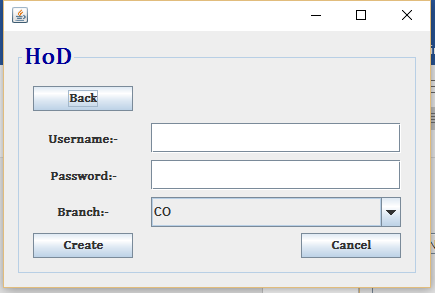


Figure when admin clicks on Create Hod Account

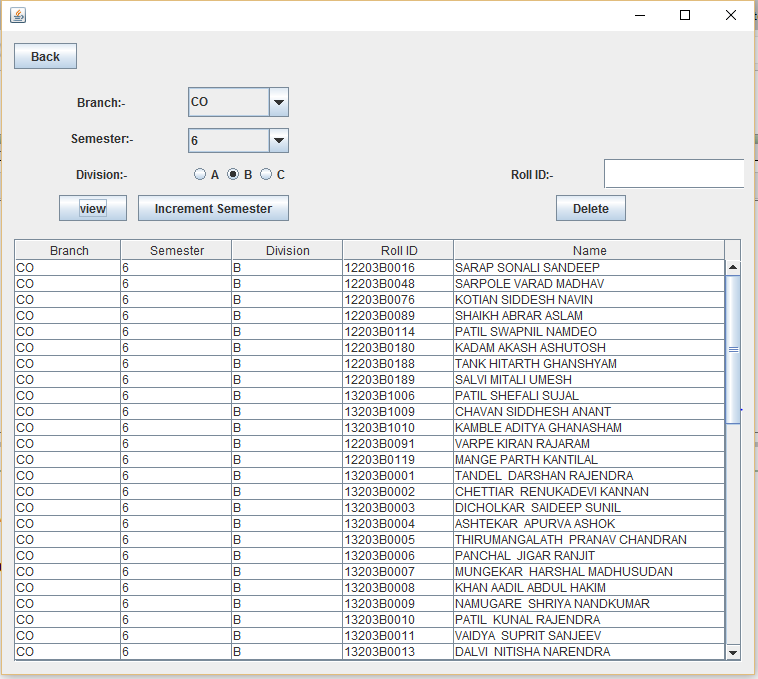


Figure when admin clicks on increment semester

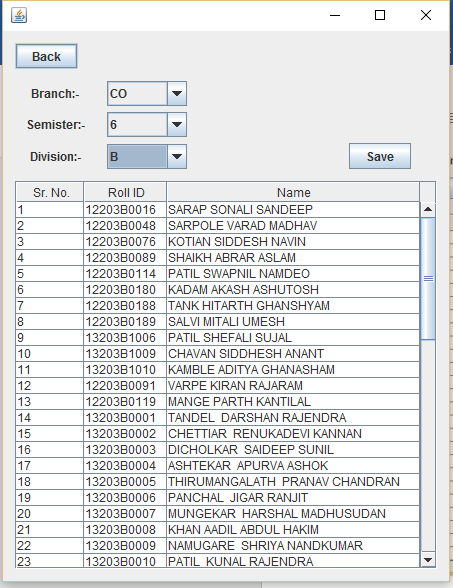


Figure when admin clicks on students not given the feedback

1. **BIBLIOGRAPHY**

**ADDENDUM:**

[http://www.w3schools.com/](http://www.w3schools.com/ado/default.asp)

http://stackoverflow.com/

<http://www.roseindia.net/programming/tutorials/MysqlTutorials.shtml>

<http://poi.apache.org/>

<http://www.roseindia.net/jdbc/Jdbc-odbc-connection.shtml>

**TEXT BOOKS**:

* Software Engineering by Roger S. Pressman (Tata Mc-Grew hill,5th edition)
* Data Base Management System by Raghu Ramakrishnan (Tata Mc-Grew hill,3rd edition)
* Unified Modeling Language 2 tool kit by Nons-Erik Eriksson, Magnus penkee, Brian Lyons, David ode (2nd Edition)
* JAVA Complete Reference by Herbert Schildt ,7th edition.
* Java Database Programming with JDBC by Patel moss.