A Project Report on

FACULTY ASSESSMENT & STUDENT COUNSEL SUITE

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Computer Application
of
KANNUR UNIVERSITY

by
NEENA THOMAS
Reg. No:DB13BCAR13

ANU BABU
Reg. No:DB13BCAR10

Don Bosco Arts & Science College
Angadikadavu, Kannur, 670706
Kerala

2016

DON BOSCO ARTS & SCIENCE COLLEGE
Angadikadavu
Iritty, Kannur

CERTIFICATE

This is to certify that the project entitled **"Faculty Assessment & Student Counsel Suite"** *submitted in partial fulfillment of the requirement for the award of the degree Bachelor of Computer Application of Kannur University, Kannur, is a result of bonafide work carried out by* **Ms.NEENA THOMAS (Reg. No: DB13BCAR13)** *in the sixth semester.*

Project Guide

Head of the Department

Angadikadavu

External Examiner

Date:

DECLARATION

I, Neena Thomas, sixth semester BCA student of Don Bosco Arts & Science College, Angadikadavu, under Kannur University do hereby declare that the project entitled "**Faculty Assessment & student Counsel suite**" is the original work carried out by me in the sixth semester under the supervision of Mrs Sindu P M, Head of the Dept. of BCA, Don Bosco Arts & Science College, Angadikadavu, in partial fulfilment of the requirement for the award of the degree Bachelor of Computer Application, Kannur University.

Angadikadavu Neena Thomas

Date: Reg.No:DB13BCAR13

ACKNOWLEDGEMENT

First of all I thank the lord Almighty for his immense grace and blessings showered on meat every stages of this work. I am greatly indebted to our Principal Fr.Francis Karackat SDB, Don Bosco Arts & Science College, Angadikadavu for providing the opportunity to take up this project as part of my curriculum.

I am deeply indebted to my project guideMrs.Sindu P M, Head of the Dept. of BCA, for his assistance and valuable suggestions as guide. They made this project a reality.

I express my sincere thanks to Mrs. Mullu Joy, Miss.Soniya Jose and Mrs.Vineetha Mathew, lecturers of department of BCA, for their valuable suggestions during the course of this project. They critical suggestions helped me to improve the project work.

Acknowledging the efforts of everyone, their chivalrous help in the course of the project preparation and their willingness to corroborate with the work, their magnanimity through lucid technical details lead to the successful completion of my project.

I would like to express my sincere thanks to all my friends, colleagues, parents and all those who have directly or indirectly assisted during this work.

1 CONTENTS

Chapters **Page No.**

. INTRODUCTION

1.1. Project Overview

My project is about “Faculty assessment and student mentoring” and named as **“Faculty assessment corner and student counsel suite”** and it is a web based application. Faculty assessment means that get a feedback from the students about their teachers in each semester in college and student mentoring means that help the students to get the required help about the educational policies of colleges.

A web based interface for getting feedback from students of college is developed, which makes feedback collection easy and mentoring is the process that includes different types of activities such as guiding students towards colleges and help the students in their problems.

Currently they are using record books and various forms to maintain these details. It will waste lots of time so the online faculty assessment and student mentoring which makes the process easy.

The main operations done by the system are,

- Maintain the student records.
- Maintain the Faculty details.
- Perform the assessment.
- Perform mentoring.
- Generate reports.

2. SYSTEM ANALYSIS

System analysis is the process of collecting and interpreting facts, understanding problems and using the information to suggest Improvement on the system. This will help to understand the existing system and determine how computers make its operation more effective. The aim of this analysis is to collect detailed information on the system and the feasibility study of the proposed system.

2.1. Existing system

Coming to the existing system the feedback and mentoring is done by manual process. In the existing system students can give feedback about the lecturers by using paper and pen and teachers also keep the details of student and their details in a record. So both this process can perform in online system without waste their time in writing. After giving feedback by every student papers are collected by the principal and calculate the overall grade for each subject and each lecturer. After that those all grade report is viewed by the lectures which is given by the principal. And each teacher have minimum of ten students and they make a meeting with the students in each semester twice and keep the details in a record.

So, the existing system is carries more time to do a piece of work for this reason the online system is implemented.

2.1.1. Problem with existing system

- Information stores and maintained manually in paper and record books.
- Searching for a particular data or record is difficult and time consuming.
- Errors and loss of data may happen.
- Less security in the data that we collected.
- Details of teachers and students record keeping is difficult.
- Mark calculation of faculty assessment is difficult and time consuming.

2.2 Proposed system

Here we aimed to design online web application for issuing the feedback about the lecturers by students and student mentoring system. This online application makes this process easy and quick manner to the users.

This project has three kinds of actors Student, Faculty, and Admin. The student can give feedback in online system provided by college staff. First of staff can prepare questions and add, update these questions to the online system. After that it was viewed by the students and can give feedback about the lecturers.

And teachers Keep the details of students and can give feedback about the students to the principal.

2.2.1. Advantages of proposed system.

- Cut your time – by giving feedback on online system when compared to the manual process this saves time of user.
- Manage the entire process – the entire process of giving feedback and viewing that report after giving feedback can manage easily.
- Enhance the staff – find the details about the lecturer's interest in teaching to the students and the teacher become more dedicated in their work.
- Data and records can easily find from the data base using search by different criteria.
- Occurrence of errors and loss of data is very less.
- Online mark calculation of faculty assessment is very easy and within the time.
- Protect data from unauthorized access.

2.3 Requirement collection

There are many methods to collect and analyse requirements. They are conducting interview, sending out questionnaires, studying similar systems etc. In Faculty assessment corner and student counsel suite I prefer the interview method. In this method I interviewed some people those who involved in the current system process. We can categories those people into three. They are faculties, admin and students. For the interview, I asked some questions. Some of the questions and answers are given below.

Interview with Admin

No.	Questions	Answers
1	How you keep feedback reports and mentoring reports?	We are keeping feedback reports in a record. For keeping mentoring reports each students have separate record. And we are write this record manually.
2	How you collect feedback?	For collecting reports, we will go to each class and collect their feedback.
3	How you calculate marks?	After collecting feedback and corresponding mark from the students calculate the mark manually based on some criteria.
4	Do you feel difficulty in the existing system? If yes, what are the	Yes, maintain student records and feedback reports are difficult. For

	difficulties	preparing these records we are wasting our time. And also for calculating marks we need more time and effort.
5	What are the things you expecting if we computerize the current system?	We think that keep record on computer gives more security than books and also more easy. And calculation of mark become very easy.
6	Which medium do you prefer to input data	Mouse

Interview with Faculty

No.	**Questions**	**Answers**
1	How you keep mentoring reports?	We are keeping mentoring reports in a record. Each students have separate record.
2	How you prepare mentoring reports?	Each teacher have minimum of ten students and they make a meeting with the students in each semester twice and keep the details in a record manually by using paper and pen.

3	Do you feel difficulty in the existing system? If yes, what are the difficulties	Yes, maintain student records are difficult. For preparing these records we are wasting our time.
4	Which medium do you prefer to input data	Mouse

Interview with Student.

No.	**Questions**	**Answers**
1	Are you facing any difficulty with giving feedback?	yes
2	If yes, what are they?	Main difficulty is regarding the time management.
3	What are the things you expecting if we computerize the current system?	Time management is easy.
4	Which medium do you prefer to input data	Mouse

2.4 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spent on it. Feasibility study lets the developer foresee the future of the project and the usefulness.

Feasibility study is a test of system proposed regarding its workability, impact on the organization, ability to meet the needs and effective use resources. Thus when a new project is proposed, it normally goes through a feasibility study before it's approved for development.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as technical, economical and behavioral feasibilities.

The proposed system is theoretically investigated to check the feasibility and found that they are more reliable and reliable in the cases given below. There are three aspects in the feasibility study portion of the preliminary investigation.

- **Economic Feasibility**
- **Technical Feasibility**
- **Behavioral Feasibility**

The proposed system must be evaluated from a technical point of view first, and if technical feasible their impact on the organization must be assessed. If compatible, the operational system can be devised. Then they must be tested for economic feasibility.

2.4.1. Economic Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors which affect the development of a new system is the cost it would require. Since the system developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.4.2. Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system

requirement in the terms of input, output, programs, procedures and staff. Having identified an outline system, the investigation must go on suggest the type of equipment, required method developing the system, of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology.

Through the technology may become obsolete after some period of time, due to the fact that newer version of some software supports older versions, the system may still be used. So there are only minimal constraints involved with this project. The system has been developed using VB.Net, the project is technically feasible for developed.

2.4.3. Behavioral Feasibility

People are inherently resistant to change and computers have been known to facilitate change. The System is designed in user friendly manner and we need to provide any special training for the persons using this software. The operating system used is Windows 7, which is also user friendly. It does not have any operational barriers. So no need to provide any special training for using this application software and hence it is behaviorally feasible.

2.5 System Specifications

2.5.1 Software Specifications

The software required for the application depends on the following factors.

- ✓ The flexibility of the software
- ✓ Software contracts
- ✓ Limitation of the software

Software

Operating system	:	Windows XP/Vista/Seven
Front End	:	asp.net

Back End : SQL

2.5.2 Hardware Specifications

The hardware required for the application depends on the following factors.

- ✓ Determining size and capacity requirements.
- ✓ Computer evaluation and measurement.
- ✓ Financial factors.
- ✓ Maintenance and support.

Hardware:

Processor	:	Intel(R) Pentium(R) 38050 @ 1.90GHz 1.90GHz (32 bit)
RAM	:	4.00 GB
HDD	:	50GB or More
Monitor	:	Display Panel (1366*768)
Keyboard	:	Standard PS/2 keyboard
Mouse	:	Standard Mouse 3 Buttons
Printer	:	Laser/Inkjet

2.6 Identification of Actors

An actor is someone or something that interacts with the system. An actor is he /she what uses the system. An actor exchanges information with

the system. Asking certain questions as detailed below can identify the actors of the system.

We can identify the actors through a list of questionnaires**.**

1	Who will use the main functionality of the system?	Administrator, Faculty, Student
2	Who will lead support from the system and do their daily tasks?	Faculty, Student
3	Who will maintain and administrate the system?	Administrator
4	With which other systems, does this system need to interact?	Database
5	Who was interest in the result produced by the system?	Administrator, Faculty, Student

The answers to these questions bring out the actors of the system us.

- **Administrator**
- **Faculty**
- **Student**

2.7 Identification of Use Cases

2.7.1 Use Case Diagram

In the analysis section, we identified the actors such as

- Administrator
- Faculty
- Student

Here we need to specify the use cases of each actor.

Identification of use cases

A use cases represents the functionality of an actor. It is defined as a set of actions performed by a system, which yields an observable result. An ellipse containing its name inside the ellipse or below it represents it. It is placed inside the system boundary and connected to an actor with an association. This shows how the use cases and the actor interact.
To find out the use cases, ask the following questions to each of the actors.

- ✓ Which functions does the actor require from the system? What does the actor need to do?
- ✓ Does the actor need to read, create, destroy, modify or store some kind of information in the system?
- ✓ Does the actor has to calculate something? And want to provide information for others?
- ✓ Could the actor's daily work be simplified or made more efficient by adding new functions to the system (typically functions which are currently not automated in the system)

Use Cases for the actor Administrator

1	Which functions does the Administrator require from the system? What does the Admin need to do?	Administrator requires the following functionalities from the system such as add faculties, remove faculties, approve/reject students, change password, add feedback, calculate marks, and view mentoring reports, search faculty and students.
2	Does the Administrator need to read, create, destroy, modify or store some kind of information in the system?	Yes, the administrator may login to the system as add faculties, remove faculties, change password, approve/reject students, add feedback, calculate marks, and view mentoring

		reports, search faculty and students.
3	Could the Administrator's daily work be simplified or made more efficient by adding new functions to the system?	Yes, the system can reduce his/her daily work.

The above queries give the following use cases for the actor Administrator

- ✓ Login
- ✓ Register Faculty
- ✓ Edit/Remove Faculties
- ✓ Change password
- ✓ Approve/reject Students
- ✓ View Mentoring Report
- ✓ View Feedback
- ✓ Submit feedback
- ✓ Search
- ✓ Generate report

Use Cases for the actor Faculty

1	Which functions does the faculty require from the system? What does the user need to do?	faculty require the following functionalities from the system such as add mentoring reports, view feedback, add/view chat, search student and generate reports.
2	Does thefaculty need to read, create, destroy, modify or store some kind	Yes, the faculty may login to the system add/edit mentoring reports, view feedback, add/view chat with

	of information in the system?	student search and generate report.
3	Could the faculty's daily work be simplified or made more efficient by adding new functions to the system?	Yes, the system can reduce his/her daily work.

The above queries give the following use cases for the actor faculty

- ✓ Login
- ✓ View feedback
- ✓ View daily feedback
- ✓ Add/edit mentoring reports
- ✓ Add/ Reply chat
- ✓ Search student
- ✓ Generate report

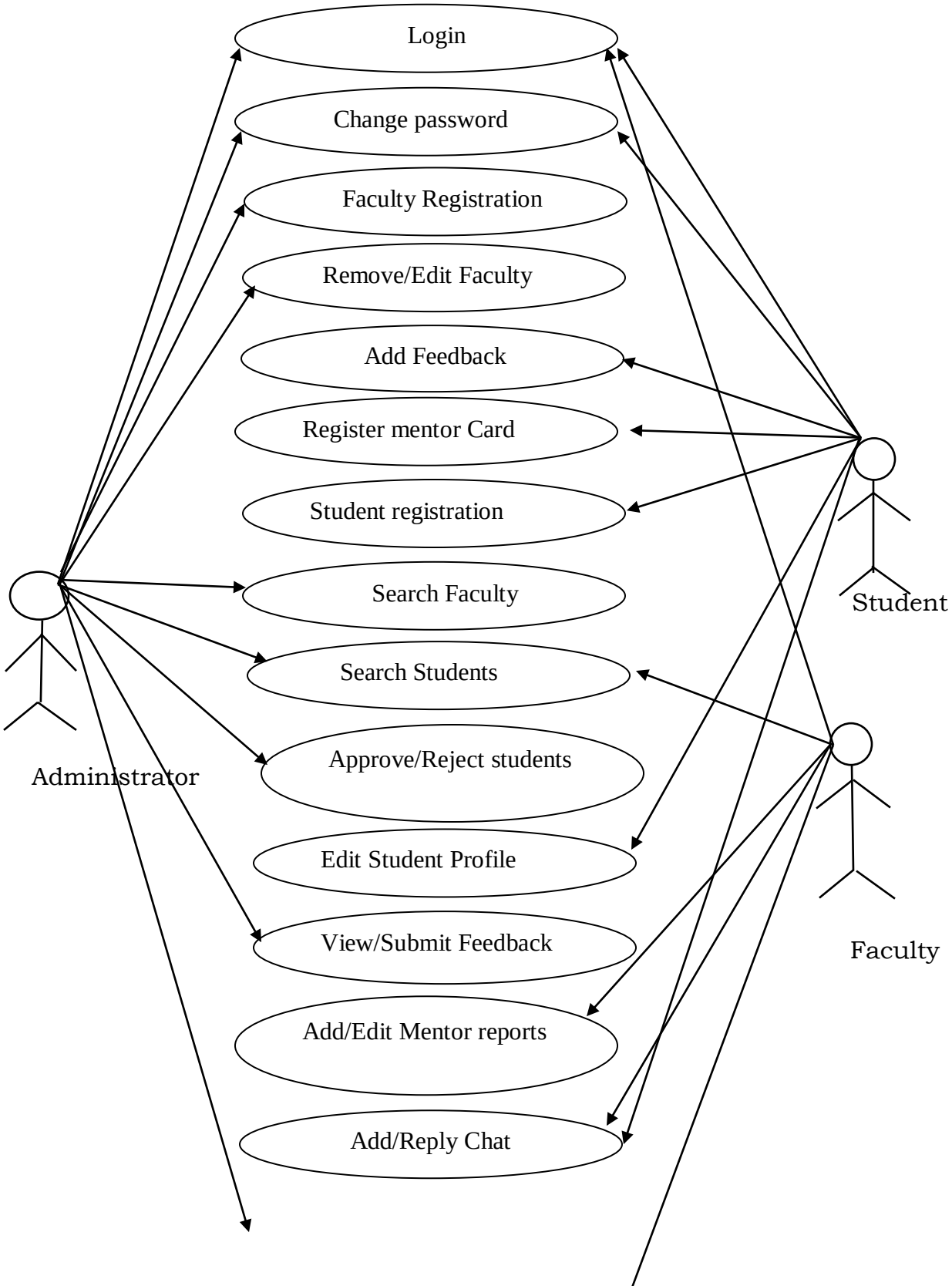
Use Cases for the actor Student

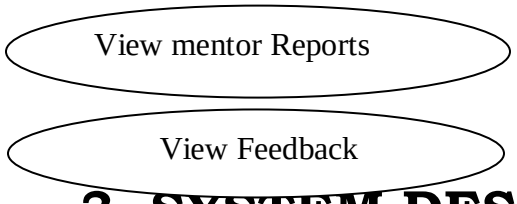
1	Which functions does the student require from the system? What does the user need to do?	Students require the following functionalities from the system such as register mentoring card, add/view chat, edit student profile, change password and add feedback.
2	Does the student need to read, create, destroy, modify or store some kind of information in the system?	Students require the following functionalities from the system such as register mentoring card, add/view chat, edit student profile, change password and add feedback.
4	Could the student's daily	Yes, the system can reduce his/her

	work be simplified or made more efficient by adding new functions to the system?	daily work.

The above queries give the following use cases for the actor student

- ✓ Login
- ✓ Register mentoring card
- ✓ Edit Student Profile
- ✓ Change Password
- ✓ Add Feedback
- ✓ Add daily feedback
- ✓ Add/replay Chat


Login
Change password
Faculty Registration
Remove/Edit Faculty
Add Feedback
Register mentor Card
Student registration
Search Faculty
Search Students
Approve/Reject students
Edit Student Profile
View/Submit Feedback
Add/Edit Mentor reports
Add/Reply Chat
Administrator
Student
Faculty



3. SYSTEM DESIGN

3.1 Database Design

A data base is a collection of inter related data stored with minimum redundancy to serve many users quickly and efficiently. In database design data independence, accuracy, privacy and security are given higher priority. Database design is an integrated approach to file design. This activity deals with the design of the physical data base. All entities and attributes have been identified while creating the database. The data base design deals with the grouping of data into number of tables so as to.

- ✓ Reduce duplication of data.
- ✓ Minimize storage phase.
- ✓ Retrieve the data efficiently.

Following are some guidelines for database design:

- ✓ Design a relational schema so that it is easy to explain its meaning. Do not combine attributes from multiple entity and relationship type into a single relation.
- ✓ Design the database schema so that no insertion, deletion or modification anomalies are present in the relation.
- ✓ As far as possible, avoid placing attributes in the base relation whose values may frequently be null.
- ✓ Design relation schema so that they can be joined with equality conditions on attributes that are either primary keys or foreign keys in a way that no spurious tuples are generated.

3.1.1 Table Design

Login table

Field name	Data type	Key	Description
Login_id	Int	Primary key	Login id of the user
username	nvarchar(50)		User name of the user
password	nvarchar(25)		Password
Type	Int		To identify the user

Student registration Table

Field name	Data type	Key	Description
Id	Int	Primary key	id of the student
Sname	nvarchar(50)		name of the student
username	nvarchar(50)		User name of the student
password	nvarchar(25)		Password
dept.	nvarchar(50)		Department of the student
Course	nvarchar(50)		course of the student
Batch	nvarchar(50)		Batch of the student
Sem	nvarchar(50)		Semester of the student
admno	nvarchar(50)		Admission no of the student
address	nvarchar(50)		address of the student
City	nvarchar(50)		City of the student
Phone no	bigint		Phone number of the

			student
email	nvarchar(50)		email of the student

Faculty Registration Table

Field name	**Data type**	**Key**	**Description**
Fid	Int	Primary key	id of the Faculty
fname	nvarchar(50)		name of the Faculty
username	nvarchar(50)		User name of the s Faculty
password	nvarchar(25)		Password
address	nvarchar(50)		address of the faculty
City	nvarchar(50)		City of the faculty
Phone no	bigint		Phone number of the faculty
email	nvarchar(50)		email of the faculty
Qualification	nvarchar(50)		Qualification of the faculty

Faccapprisal Table

Field name	**Data type**	**Key**	**Description**
Id	Int	Primary key	id
academicyear	nvarchar(50)		name of the student
Sem	nvarchar(50)		User name of the student
fname	nvarchar(25)		Password
Course	nvarchar(50)		Subject taught
programme	nvarchar(50)		Current program
batch	nvarchar(50)		Batch of the student
semester	nvarchar(50)		Current Semester

date	nvarchar(50)		Corresponding date
tmark	nvarchar(50)		Total mark of the faculty
suggestion	nvarchar(50)		Suggestion about faculty

Mentoring card Table

Field name	**Data type**	**Key**	**Description**
mcid	Int	Primary key	id of the student
dept	nvarchar(50)		name of the student
pgme	nvarchar(50)		User name of the student
batch	nvarchar(25)		Password
Rno	nvarchar(50)		Department of the student
admno	nvarchar(50)		Admission number of the student
sname	nvarchar(50)		name of the student
Dob	nvarchar(50)		Date of birth
gendr	nvarchar(50)		gender
Pob	nvarchar(50)		Place of birth
plcetype	nvarchar(50)		Place type
bloodgrp	nvarchar(50)		bloodgroup
relgn	nvarchar(50)		religion
caste	nvarchar(50)		caste
catogry	nvarchar(50)		catogory
Stay	nvarchar(50)		place
familytype	nvarchar(50)		Family type
fathername	nvarchar(50)		Name of father
occupation	nvarchar(50)		Occupation of father

mothername	nvarchar(50)		Name of mother
occupation	nvarchar(50)		Occupation of mother
noofbrothers	nvarchar(50)		No.of brothers
noofsisters	nvarchar(50)		No.of sisters
housename	nvarchar(50)		housename
post	nvarchar(50)		post
place	nvarchar(50)		place
district	nvarchar(50)		district
Pin	nvarchar(50)		Pin code
mail	nvarchar(50)		Mail id of the student
phone	nvarchar(50)		Phone number
Languge known	nvarchar(50)		Language known
memberof	nvarchar(50)		Membership in any of the organization
disability	nvarchar(50)		Have any disability
hospitalized	nvarchar(50)		Hospitalized before
futureplan	nvarchar(50)		Future plan
career	nvarchar(50)		Career goal
strength	nvarchar(50)		Personal strength
improve	nvarchar(50)		Area need to improve
rolemodel	nvarchar(50)		Role model of the student
bestfriend	nvarchar(50)		Best friend of the student
phoneno	bigint		Phone number
mentorname	nvarchar(50)		Name of mentor
place	nvarchar(50)		place
date	nvarchar(50)		Current date

Mark list

Field name	Data type	Key	Description
Mid	Int	Primary key	id of table
mcid	int	Foreign Key	id of mentor card
qulification			Qualification of student
tmark			Total mark obtained

Student daily feedback Table

Field name	Data type	Key	Description
cid	int	Primary Key	id
programme	nvarchar(50)		Current program
Sem	nvarchar(50)		Current semester
Faculty name	nvarchar(50)		Faculty name
chat	nvarchar(50)		chat
date	nvarchar(50)		Current date
towhom	nvarchar(50)		To whom

Student Chat Table

Field name	Data type	Key	Description
cid	int	Primary Key	id
name	nvarchar(50)		Student name
mentorname	nvarchar(50)		Mentor name
chat	nvarchar(50)		chat
date	nvarchar(50)		Current date

Faculty add mentor report Table

Field name	Data type	Key	Description
Mid	int	Primary Key	id
Sname	nvarchar(50)		Name of student
Admission no	nvarchar(50)		Admission no of the student
programme	nvarchar(50)		Programme of the student
Course	nvarchar(50)		Course of student
Sem	nvarchar(50)		semester
Date	nvarchar(50)		Current date
report	nvarchar(50)		report

Teacher chat Table

Field name	Data type	Key	Description
chatid	int	Primary Key	id
sender	nvarchar(50)		From sender
reciever	nvarchar(50)		To reciever
admno	nvarchar(50)		Admission no of student
Date	nvarchar(50)		Current date
Chat	nvarchar(50)		chat

Admin Search Table

Field name	Data type	Key	Description
Asid	int	Primary Key	id
programme	nvarchar(50)		programme
Sem	nvarchar(50)		Current semester

fname	nvarchar(50)		name

3.2 Architectural Design

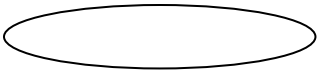
3.2.1 Data Flow Diagram

A data flow diagram is a graphical technique that depicts data flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A Level 0 DFD also called a fundamental system model or context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in next level i.e., Level 1 DFD. Each of the processes represented at level 1 are sub functions of overall system depicted in the context model.

Data flow diagram symbols:

Source/Destination of Data

Data flow

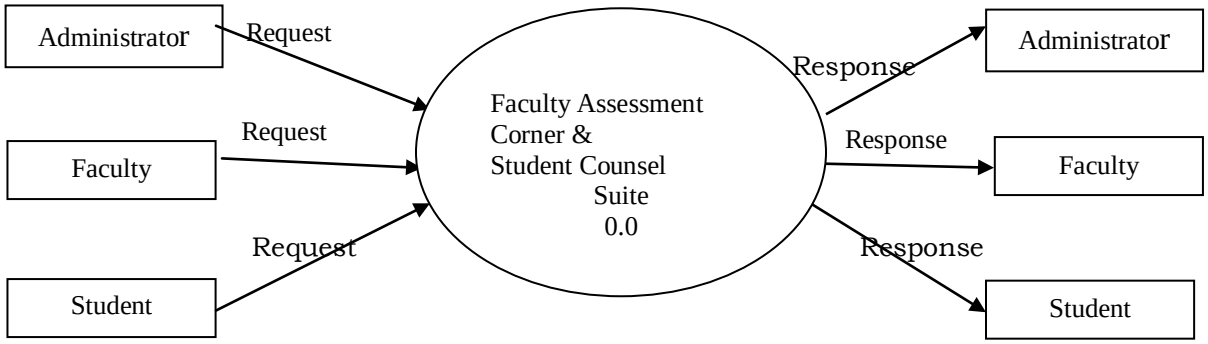


Process

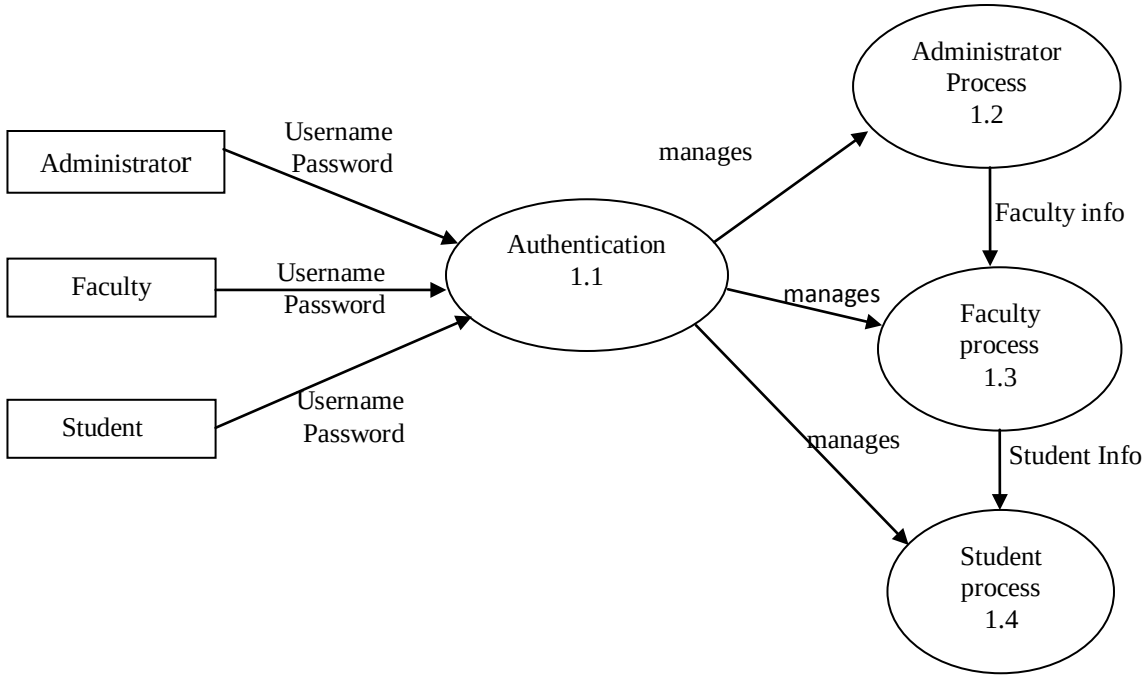
Storage

Data Flow Diagram (DFD)

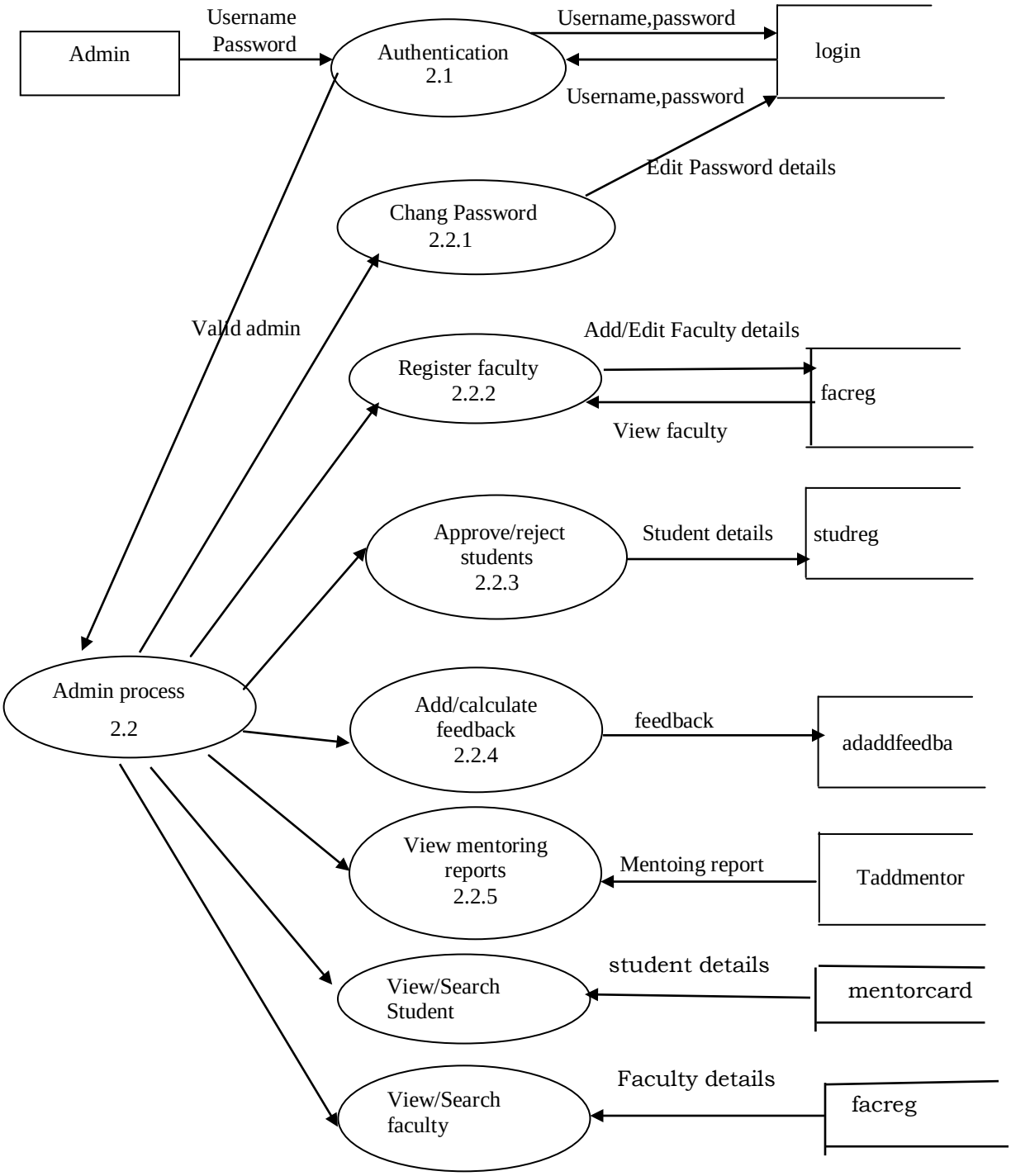
Level – 1



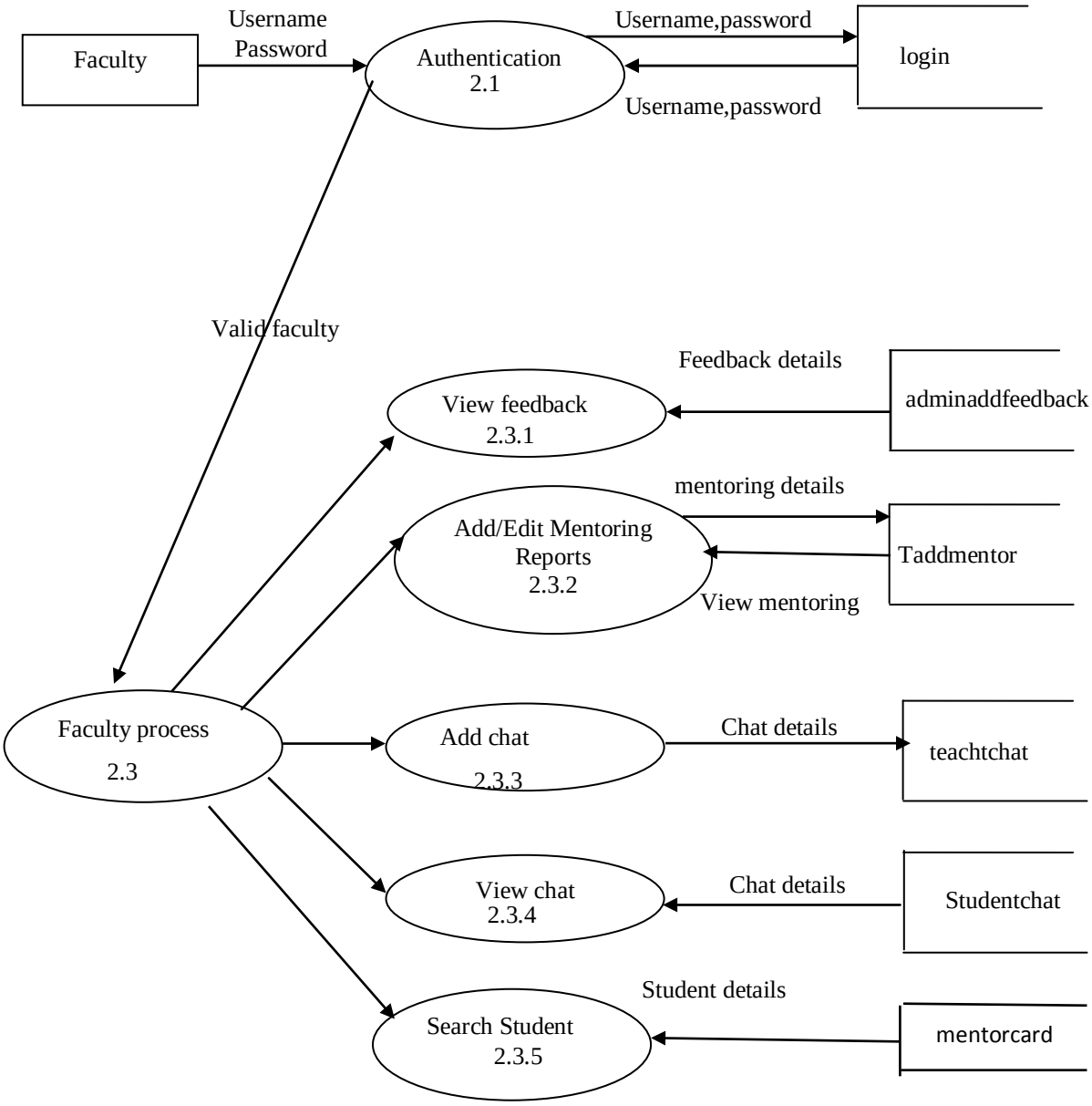
Level -1



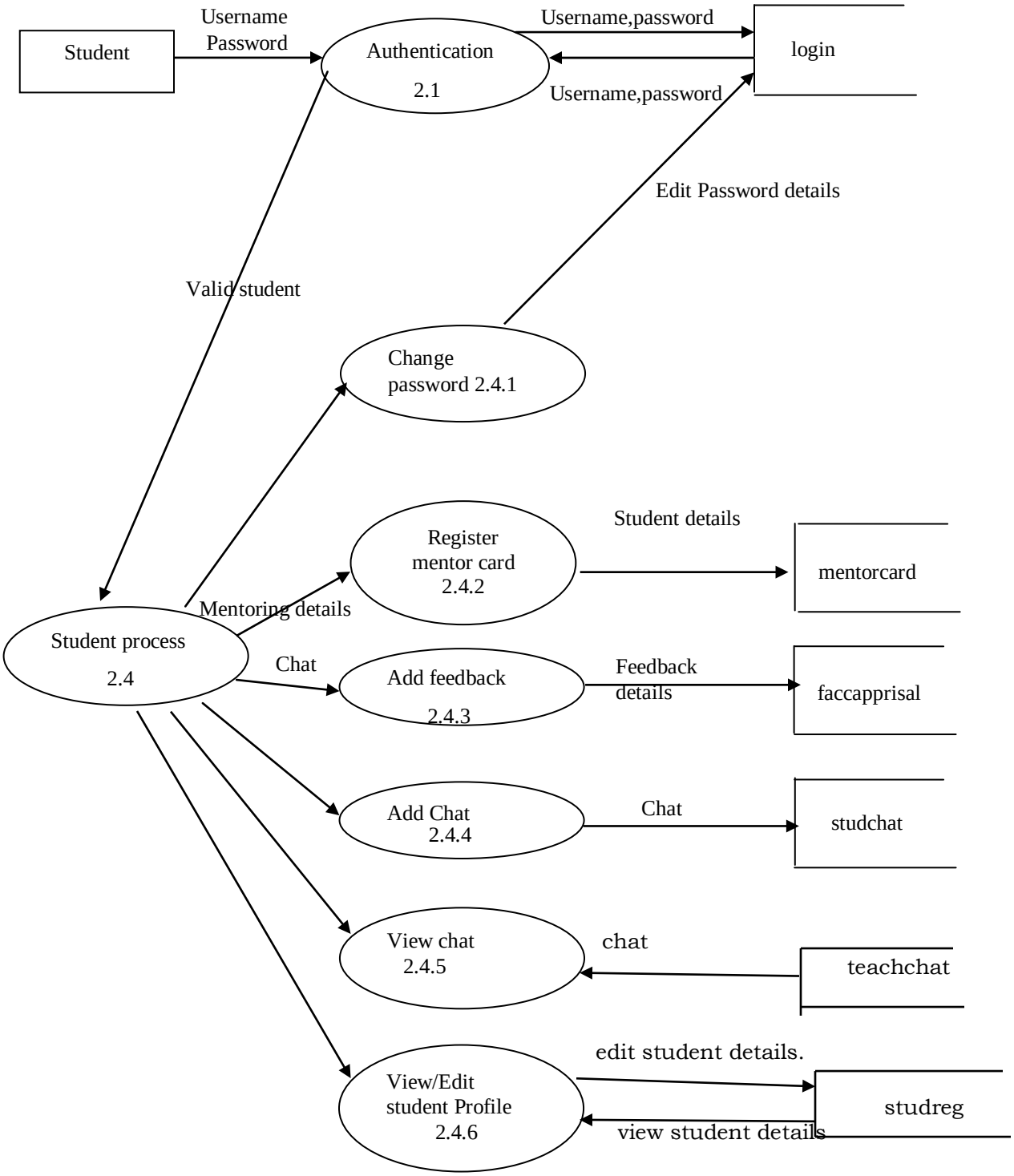
Level -2 Administrator



Level – 2 Faculty



Level – 2 Student



3.3 Interface Design

An interface design elements for the software tell how information flows into and out of the system and how it is communicated among the components as part of the architecture.

3.3.1 Input Design

Input design is the link between the information system and actors and those steps that are necessary to put transaction data into a usable form for processing data entry. Instructing the computer to read data from a written printed document can active the activity of putting data into the computer for processing or it can occur by keying data directly into the system. The design of input focusing on controlling the errors, avoid delay, and keeping the process simple. System analyst decides the following input design details

- ✓ What data to input?
- ✓ What medium to use?
- ✓ How the data is arranged and coded?

In my project named faculty assessment corner & student counsel suite, I tried to include the following design constrains provided in the software engineering.

1: Avoid scattering of fields in the forms

In all forms of the software the textboxes (which provided to input some data), label (which label the text boxes), combo box (list a set of values) etc all are arranged in a neat and well format. It provides a simple look to the pages. The buttons are placed at the bottom of the page and easily accessible to the user. The menu are arranged below the heading and at a minimum level of menus are arranged with pages. Menu provides the continuity to the pages.

2: User only needs to enter a minimum amount of data

All forms contain a minimum amount data, but most essentials. No page provides or wanted bulky of data. It provides more easiness to the user. It creates more the software to the end user. Also the operation continues by single click.

3: Avoid confusion in the forms

All forms have a well-defined menus and each menu name indicate its purpose. So the user can easily access various forms without confusion. Each form and its sub forms are well labelled. So the user can easily identify the forms and work on that.

3.3.2 Output Design

Designing computer should proceed in well thought out manner. The term output means any information produced by the information system weather printed or displayed. Output design is a process that involves designing necessary output that have to be used by various users according to requirement. The efficient intelligent output design should remove the system relationship with the users and help in decision making.

When designing the output, system analyst must accomplish the following:

- ✓ Determine the information present
- ✓ Decide whether to print, display the information and select output medium
- ✓ Arrange information in acceptable format.

In my project, the outputs are in the form of reports. They are well format and it provides the output in a correct and neat format.

3.4 Procedural Design

The system has two modules. They are…

- ✓ Administrator module
- ✓ Faculty module
- ✓ Student module

1. Administrator module

Administrator is the one who controls the organization. He can add or remove staffs, store and retrieve child details, missing complaints, success stories and reminders and generate report

Following are the functions that can be performed by the administrator

- ✓ Login
- ✓ Register Faculty
- ✓ Edit/Remove Faculties
- ✓ Change password
- ✓ Approve/reject Students
- ✓ View Mentoring Report
- ✓ Add/Calculate Feedback
- ✓ Search
- ✓ Generate report

Login

Administrator login to the system and use his/her services

Register faculty

The administrator register the details of new faculties and gives access rights to them.

Edit/Remove staff

The administrator edit or removes the faculty details from the system.

Change password

The administrator changes the password of faculties and himself.

Approve or Reject students

The administrators approve or reject students into the system.

View mentoring reports

The administrator view the mentoring reports in the system.

Add or submit feedback

The administrators add or calculate feedback in the system.

Search

The administrator search for students and faculties information that already in the system.

Generate report

The administrator generates various reports on various data that present in the system.

2. Faculty module

Faculty is the one who interact with the system every time. View feedback, add/edit mentoring report, add/view chat, search student are the main duty of the staff.

Following are the functions that can be performed by the staff:

- ✓ Login
- ✓ View feedback
- ✓ Add/edit mentoring reports
- ✓ Add/Reply chat
- ✓ Search students
- ✓ Generate report

Login

Staff login to the system and use his/her services

View feedback

The faculty view feedback from admin and from students.

Add/Edit mentoring report

The faculty adds mentoring details into the system.

Add/Reply chat

The faculty add chat into the system and view chat from the students.

Search students

The faculty searches for student's information that already in the system.

Generate report

The faculty generates various reports for the system.

3. Student module

Faculty is the one who interact with the system every time. View feedback, add/edit mentoring report, add/view chat, search student are the main duty of the staff.

Following are the functions that can be performed by the staff:

- ✓ Login
- ✓ Register
- ✓ Edit Student Profile
- ✓ Change Password
- ✓ Add Feedback
- ✓ Add/Reply Chat

Login

Student login to the system and use his/her services

Register

The students register mentoring card

Edit student profile

The students can edit their profile.

Change password

The administrator changes the password of faculties and himself.

Add feedback

The students add feedback about their faculty.

Add/Reply chat

The faculty add chat into the system and view chat from the students.

4. CODING

4.1. Description of Software Used

Microsoft.net

.NET is an integral part of many applications running on Windows and provides common functionality for those applications to run.Microsoft.NET is prefabricated infrastructure for solving common problems in internet applications. Microsoft.NET has been getting enormous amount of publicity lately, even for this industry Microsoft. NET is an add-on-run-time environment that runs on windows 2000 operating system. Later versions of .NET will probably be made part of the operating system, the U.S. department of justice willing.

Microsoft.net framework

The .NET Framework includes the runtime and compiles time services required to run a .NET application. Compile time is when the developer is compiling the source. Runtime is when the compiled code is executing in memory. At the center of the runtime execution of .NET code is the Common Language Runtime (CLR). The CLR is a virtual machine that runs as a process on the computer on which it is installed.

COMMON LANGUAGE RUNTIME

The Common Language Runtime (CLR) serves as the execution environment for the .NET Framework. The CLR is responsible for managing the compiled code of .NET applications, which can be written in different languages including VB, C#, Java, and Perl. The cross-language integration is achieved through the two major components of CLR: Intermediate Language and Metadata.

Intermediate Language (IL) is an assembly language that runs on almost any type of CPU. IL achieves this versatility by using stacks to handle

functions that normally occur in registers. As managed code, IL is just-in-time (JIT) compiled when .NET applications are executed. JIT compilers convert IL into machine language that is specific to the host CPU. [During runtime, JIT compilers have the luxury of choosing stacks, registers or other stores to implement IL stacks.] Various JIT compilers are provided by the CLR, making it possible for different computer architectures to execute IL. Unlike other assembly languages, IL integrates high-level concepts which make CLR code more robust. As a high level language, IL is strongly typed and uses the ideas behind structured-exception handling, deployment support, and component interaction. Thus a range of software can run on the .NET Framework as long as the compiler can produce IL.

Metadata, the second component of the CLR, is a description of the implemented code. The Metadata is responsible for ensuring that the CLR executes the code securely. To prevent modules of software from breaking type definitions, Metadata stores information regarding classes, methods, and types. Registers are no longer required to keep track of information because relevant data is stored with the compiled code or IL. By housing information on classes and registrations, Metadata allows the CLR to function more efficiently since programs are less likely to get hung up on version and inheritance dependencies.

Language(c#)

Microsoft Visual C# is a powerful but simple language aimed primarily at developers creating Applications by using the Microsoft .NET Framework. It inherits many of the best features of C++ and Microsoft Visual Basic, but few of the inconsistencies and anachronisms, resulting in a cleaner and more logical language. The advent of C# 3.0 has seen several important new features added to the language, including Generics, Iterates, and anonymous methods. By design, C# is the programming language that most directly reflects the underlying CLI.

Most of its intrinsic types correspond to value-types implemented by the Common Language Infrastructure (CLI) framework. However, the language specification does not state the code generation requirements of the compiler: that is, it does not state that a C# compiler must target a Common Language Runtime, or generate Common Intermediate Language (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or FORTRAN

Features of c#

- C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.

- The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.

- The language is intended for use in developing software components

  Suitable for deployment in distributed environments.

- Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.

- Support for internationalization is very important.

- C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended

to compete directly on performance and size with C or assembly language.

Microsoft SQL Server 2008

Microsoft SQL Server 2008 provides the Microsoft Windows Server System integrated server software with a database platform for the next generation of connected, scalable, and reliable enterprise applications. The breadth and depth of innovation in this version is in response to t he needs of customers. This white paper is targeted to database administrators, to give you an understanding of the new features in and capabilities of SQL Server 2005. From many enhancement of existing features, to an entirely new security model, database administrative is now more productive and in tune with needs of the administrator.

At the core of SQL Server 2008 are new infrastructure application capabilities. SQL Service Broker is a distributed application framework that provides a new form of scalability and reliability for asynchronous message delivery. Though not new, Microsoft SQL server Notification Services, Reporting Services, and SQL Server Mobile Edition (formerly called SQL Server CE) are all greatly enhanced in SQL Server 2005.

SQL (Structured Query Language) is a database computer language designed for the retrieval and management of data in relational database scheme creation and modification, and database object access control management.

4.2 Coding Principle

The input to the coding phase is the design document. During coding phase, modules identified in the design document are coded according to the module specification. Objectives of coding phase are, to transform design into code and unit test the code.

Coding Guidelines

- ✓ Code should be easy to understand.
- ✓ Don’t take pride in cryptic code.
- ✓ Code should be well documented.
- ✓ Comments should be present.
- ✓ Functions should be small.
- ✓ Do not use Go-to statement.
- ✓ Do not use the same variable for multiple purposes.

5. SYSTEM TESTING

Testing is a methodology for evaluating the project. The good test has a high probability of finding errors. Testing is generally two types.

White box testing

White-box testing is also called as glass-box testing, is a test case design method that goes to the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

- ✓ Guarantee that all independent paths within a module have been exercised at least once.
- ✓ Exercise all logical decision on their true and false sides.
- ✓ Execute all loops at their boundaries and within their operational sides.
- ✓ Exercise internal data structure to ensure their validity.

White box testing was successfully conducted on our system. All independent paths within a module have been executed at least once and all logical decisions have been exercised on their true and false sides.

Black box testing

Black-box testing is also called as behavioral testing, focuses on the functional requirement of the software. It is a complementary approach that is likely uncover a different class of errors than white box methods. Black box testing attempts to find errors in the following categories

- ✓ Incorrect or missing functions.
- ✓ Interface errors.
- ✓ Error on data structures or external database access.
- ✓ Behavior or performance errors.
- ✓ Initialization and termination errors.

Black box testing was successfully conducted on our system. The system was divided into a number of modules and testing was conducted on each module. We have tested the system for incorrect or missing functions, interface and performance errors.

TEST CASES

Login form

No.	**Test scenario**	**Expected result**	**Observed result**	**result**
1.	Enter wrong user name and password	Display login form again with invalid message.	Message displayed.	Pass
2.	Enter correct user name and wrong password	Display login form again with invalid message.	Message displayed.	Pass
3.	Enter correct user name and correct password of administrator or faculty or student.	Administrator or faculty or student can login to the system. MDI form load with	Login to their pages.	Pass

		menu.		
4.	Press login button without filling the user name and password.	Display warning message to fill the details.	Warning message displayed	Pass

Register faculty

No.	**Test scenario**	**Expected result**	**Observed result**	**result**
1.	Form displayed	Display form with all controls.	Form loaded with all controls.	Pass
2.	Click register button without data	Display warning message to fill the details.	Warning message displayed.	Pass
3.	Click register button with data	Display message saved success fully and clear the fields	Message displayed and all fields cleared	Pass
4.	Click cancel button	Exit the form	Exited	Pass

Register Mentor card

No.	**Test scenario**	**Expected result**	**Observed result**	**result**
1.	Form displayed	Display form with all controls.	Form loaded with all controls.	Pass
2.	Click register button	Display	Warning	Pass

	without data	warning message to fill the details.	message displayed.	
3.	Click register button with data	Display message saved success fully and clear the fields	Message displayed and all fields cleared	Pass
4.	Click cancel button	Exit the form	Exited	Pass

Search

No.	**Test scenario**	**Expected result**	**Observed result**	**result**
1.	Form displayed	Display form with all controls.	Form loaded with all controls.	Pass
2.	Click search button without ID on text box	Display warning message to fill the details.	Warning message displayed.	Pass
3.	Click search button with ID on text box	Display data in fields with respect to the ID.	Record displayed	Pass
4.	Click update button without data	Display warning message to fill the details.	Warning message displayed.	Pass
5.	Click update button with data	Display message	Message displayed and	Pass

		update success fully and clear the fields	all fields cleared	
6.	Click cancel button	Exit the form	Exited	Pass

View

No.	**Test scenario**	**Expected result**	**Observed result**	**result**
1.	Form displayed	Display form with all controls.	Form loaded with all controls.	Pass
2.	Click all tab pages one by one.	Display tab page with all corresponding data	Tab pages displayed with corresponding data	Pass
4.	Click Edit button with data	Display form and edit the data successfully	Message displayed and all fields cleared	Pass
5.	Click Delete button	Delete the data successfully	Delete the data and display new form	Pass
6.	Click cancel button	Exit the form	Exited	Pass

Add feedback

No.	Test scenario	Expected result	Observed result	result
1.	Form displayed	Display form with all controls.	Form loaded with all controls.	Pass
2.	Click save button without data	Display warning message to fill the details.	Warning message displayed.	Pass
3.	Click save button with data	Display message saved success fully and clear the fields	Message displayed and all fields cleared	Pass
4.	Click cancel button	Exit the form	Exited	Pass

6. SYSTEM IMPLEMENTATION

Implementation is the process of deploying the new system in the operational environment. Proper implementation is essential to provide a reliable system to meet the organizational requirement. There are four types of implementation methods. They are Direct Changeover, Phased Implementation, Parallel Run and Pilot Approach. The most commonly used implementation methods are Pilot Approach and Parallel Run.

The implementation method I used to implement faculty assessment corner and student counsel suite is Parallel Run. i.e., the new system will work parallel to the existing system. The new system will replace the existing system completely after six months.

Faculty assessment corner and student counsel suite is developed as a desktop application, as usual some desktop development technologies are used in the implementation of the project. The language I selected to program this software is ASP.Net.The reason I selected ASP.Net is that is a simple and powerful language that especially developed to create desktop application.

Technologies used in the development of the software are:

- ✓ Programming language - ASP.Net
- ✓ DBMS - SQL Server 2008
- ✓ Development tool - Visual Studio 2012
- ✓ Development platform - Windows 7

The front end is ASP.Net and back end is SQL server 2008. The system developed on Visual studio 2012 development tool in Windows 7 operating system.

7. CONCLUSION

The project faculty assessment corner and student counsel suite is for computerizing the working of the college. The software takes care of all the requirement of College and is capable to provide easy and effective storage of information related to students and faculty details, feedback system and mentoring reports. This desktop application offers several features, which is very useful for the administrator, faculties and students. This project allows different actors of the system to work within the same environment. Using the facility and flexibility of ASP.Net and SQL server, the software has been developed in a neat and simple manner, thereby reducing the actor's work.

Faculty assessment corner and student counsel suite providing following features to its users:

- ✓ A user with minimum knowledge can be able to operate the system easily.
- ✓ The software is developed with modular approach
- ✓ All modules in the system have been tested with valid and invalid data.
- ✓ Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

FUTURE ENHANCEMENT

As a future venture, it is suggested to make some changes to provide more services and to provide information at right time in right manner.

They are:

- ✓ All the data cards for the data collections can be computerized.

- ✓ This software can be converted to a web application and used by all college for their needs**.**
- ✓ Using this software can easily find out the information related to students and faculties.

8. REFERENCE

- ✓ Steven Holzer, “Visual Basic.net Programming Black Book” Paraglyph Press USA , 2005.

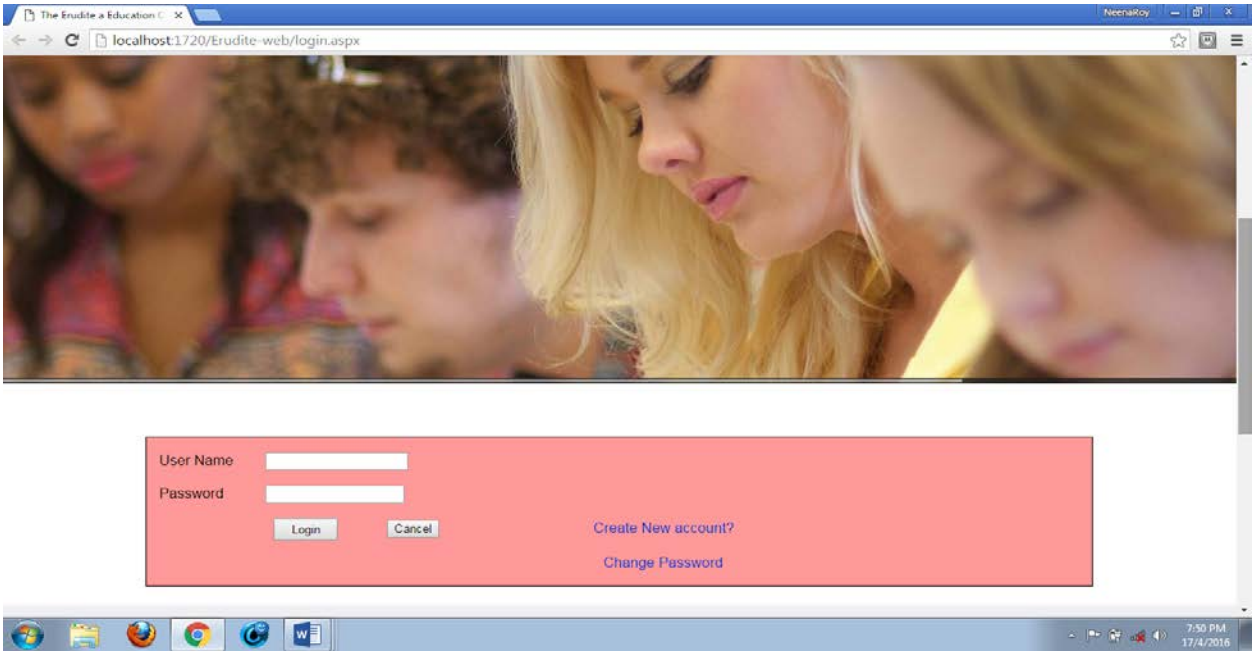
- ✓ Jeffery R. Shipario, “The Complete Reference Visual Basic.Net” Tate McGraw-Hill, 2002.

- ✓ Patrick Dalton, Paul Whitehead, “SQL SERVER BLACK BOOK” Dream Tech New Delhi, 2005.

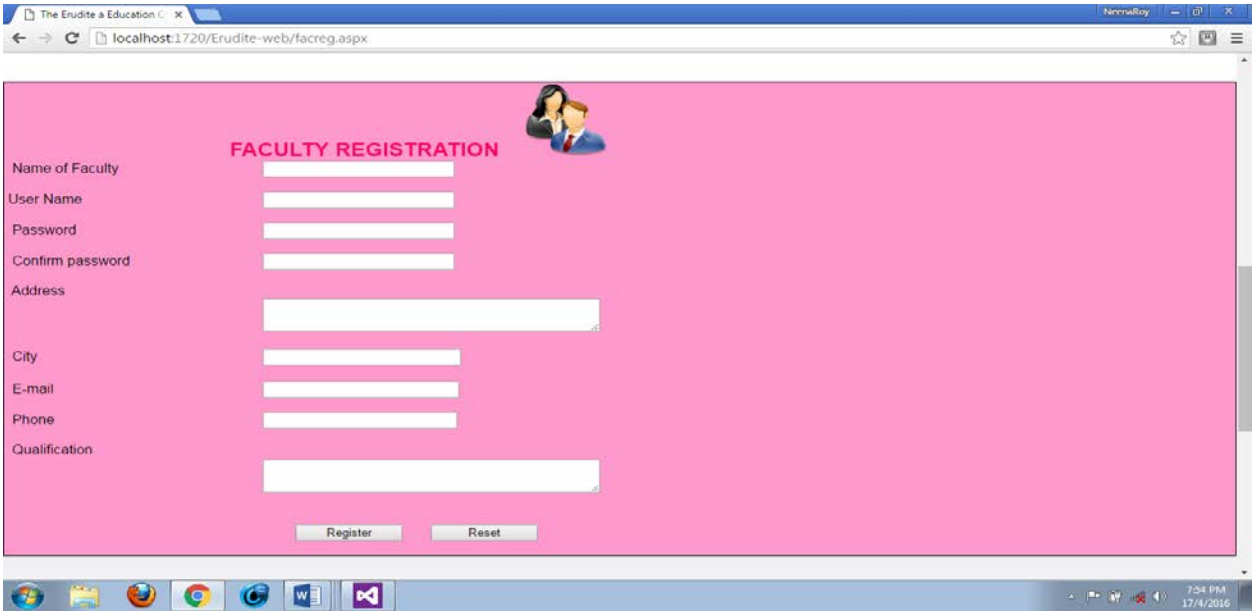
- ✓ Roger Pressman, “Software Engineering A Practitioner's Approach”, McGraw-Hill Company, 2001.

9. APPENDIX

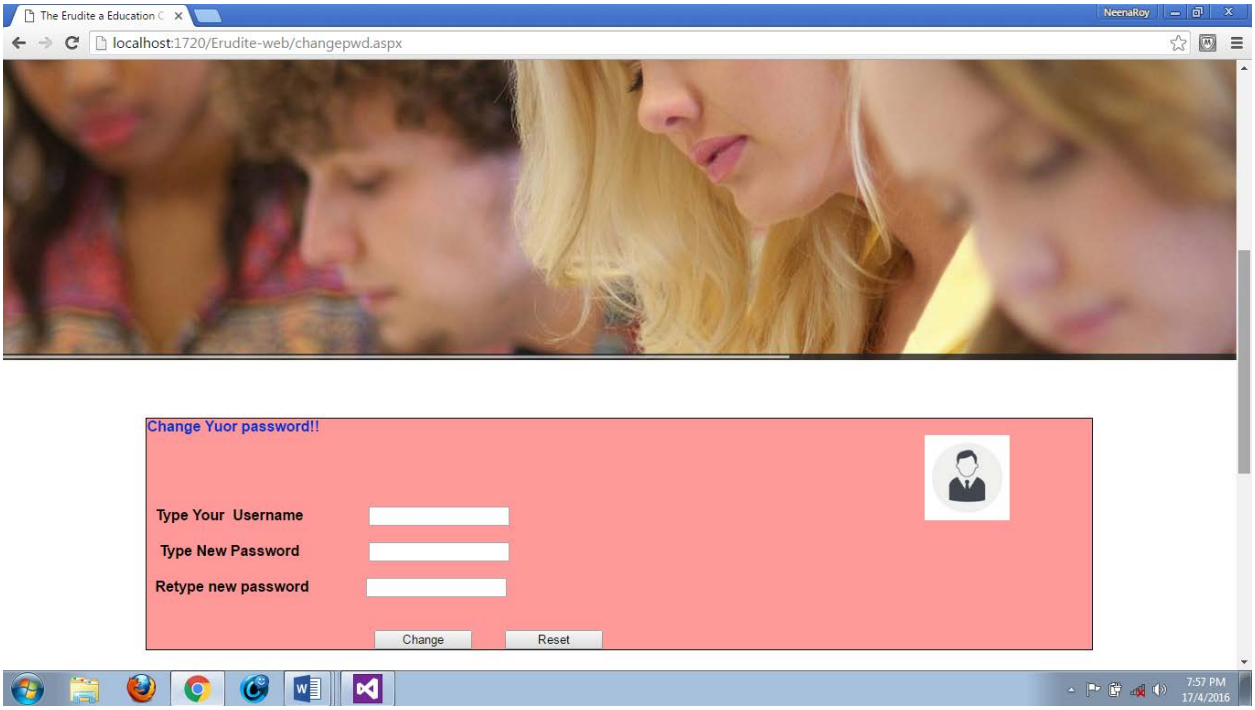
Login form



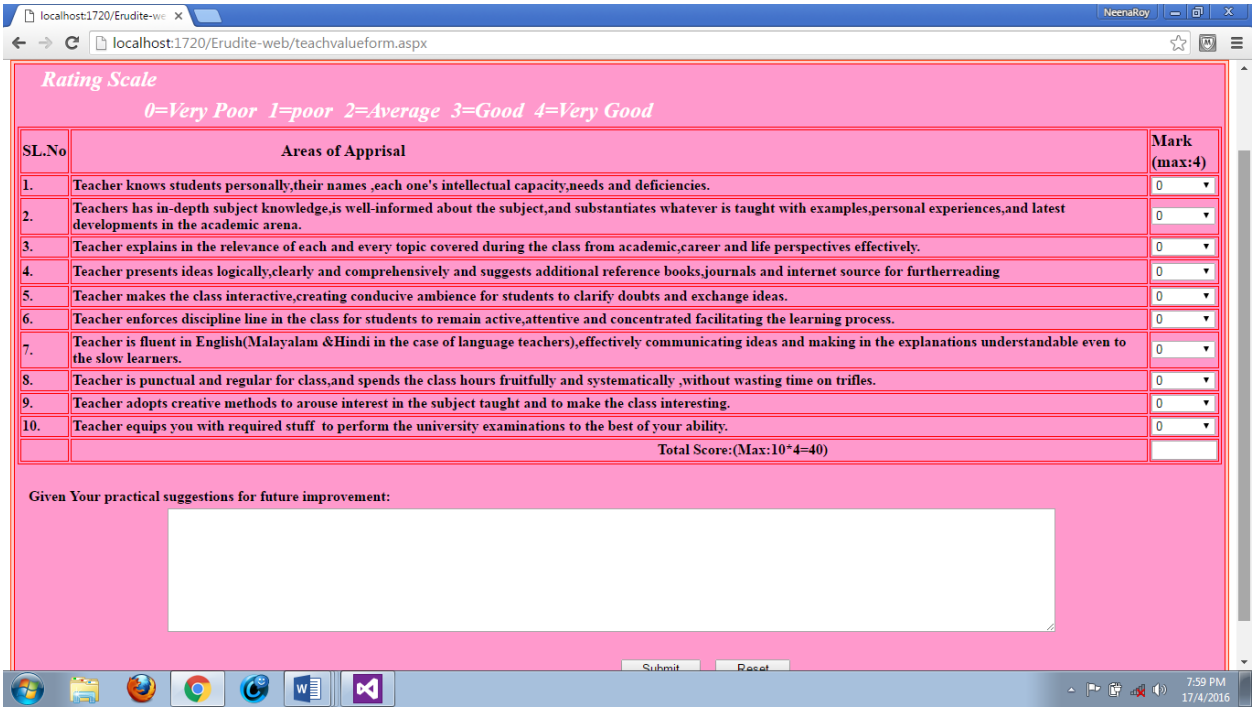
Faculty registration



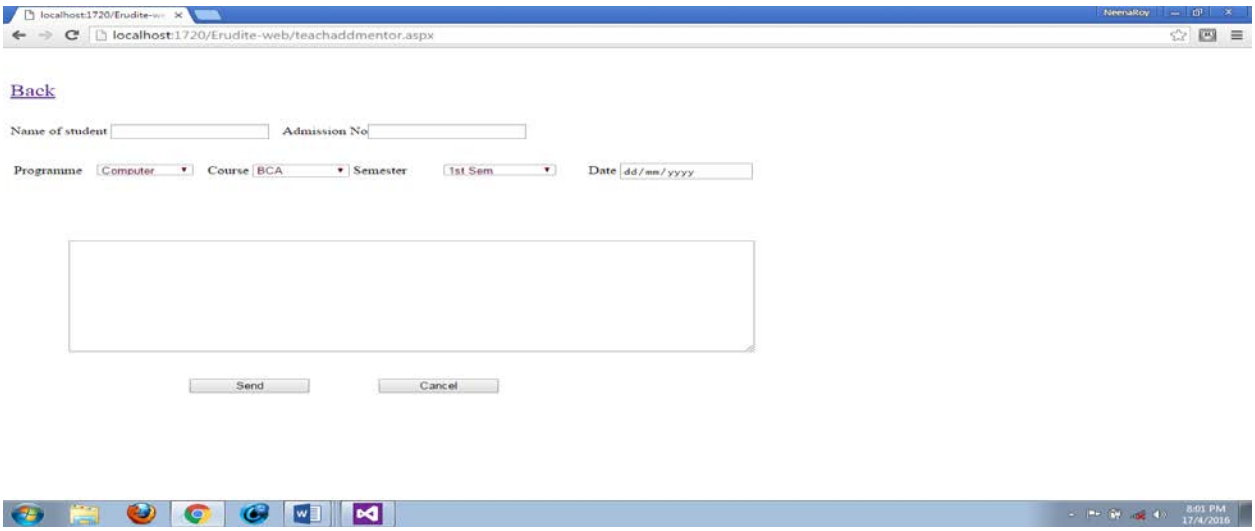
Change password



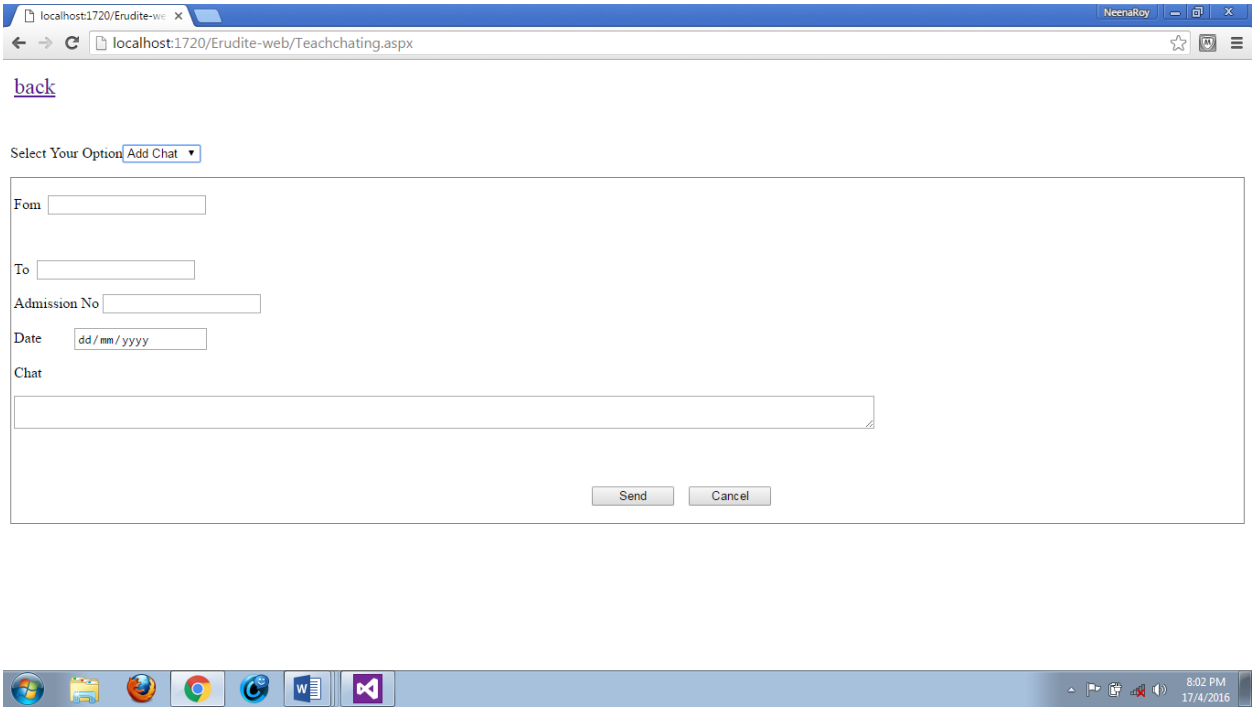
Add feedback



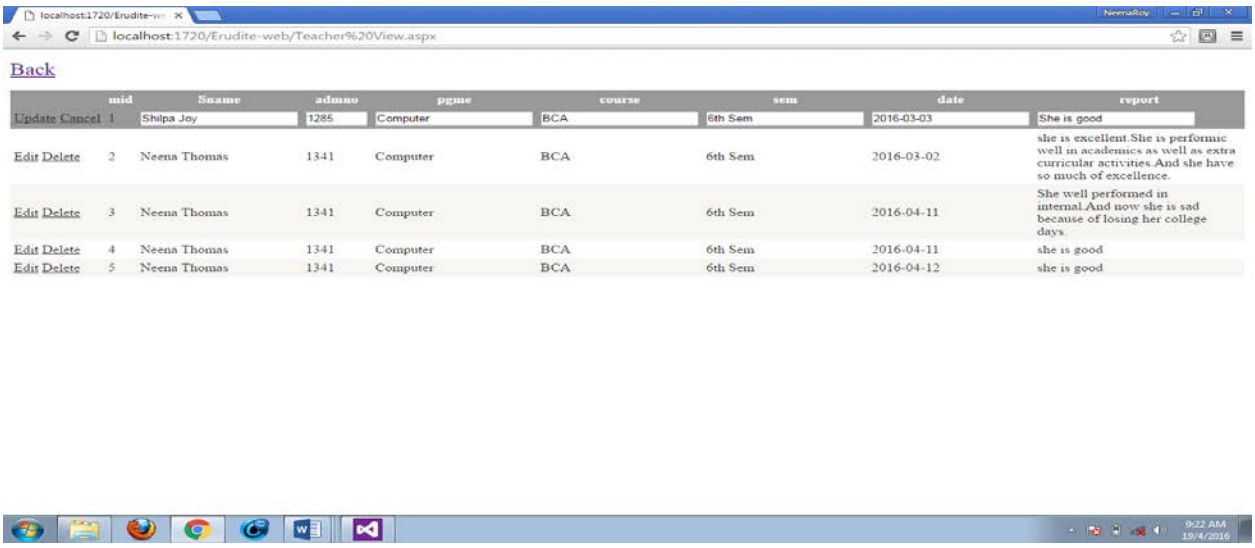
Add mentoring report



Add chat



View/Edit Mentoring reports



View Daily feedback