DESIGN AND IMPLEMENTATION OF BRAC UNIVERSITY ONLINE ADVISING APPLICATION

Hasan Ferdouse Sharif (ID: 06101004)
Nadeem Hussain (ID: 07341004)
Saniun Saber Chowdhury (ID: 05301002)

Department of Computer Science and Engineering
School of Engineering and Computer Science

April 2010

BRAC University, Dhaka, Bangladesh

DESIGN AND IMPLEMENTATION OF BRAC UNIVERSITY ONLINE

ADVISING APPLICATION

A Thesis

Submitted to the Department of Computer Science and Engineering

of

BRAC University

by

Hasan Ferdouse Sharif (ID: 06101004)

Nadeem Hussain (ID: 07341004)

Saniun Saber Chowdhury (ID: 05301002)

In Partial Fulfillment of the

Requirements for the Degree

of

Bachelor of Science in Computer Science and Engineering

April 2010

DECLARATION

I hereby declare that this thesis is based on the results found by ourself. Materials of work found by other researchers are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of
Supervisor

Signatures of
Authors

ACKNOWLEDGMENTS

Special thanks to Abdussamad Ahmed Muntahi, Saad Matin Abdullah and Sarwar Alam for taking time off their busy schedules in order to assist in the development of this work.

ABSTRACT

Student advising is a task that needs to be done at the beginning of every semester by all students who wish to register for courses in the upcoming semester. This is a tedious job and requires all advisors to be available during the advising period. As the number of students are progressively increasing year by year, the software used to advise the students for a semester need to be efficient andminimally time consuming for the advisors. Students need to have full access to all required information before meeting an advisor. In order to accomplish this, we have developed a fully web based advising system using ASP.NET, C# and SQL database. The students and faculty members who have been assigned as advisors will benefit from the use of this software. Being a web based software, students can finalise all the required advising decisions from the comfort and time of their homes instead of using the allotted window of advising in the university. Advisors only need to log onto their accounts, search for a student and confirm their advising if all courses are added to the advising sheet. Advisors also have the freedom to add or delete courses from the student's advising sheet before confirming upon requirement.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

1.1 Existing System

In the current system, database entry starts when a new student is enrolled after admission. When a student registers for the admission test, he is issued an applicant ID. This ID is then used as the primary key to identify the student throughout his endeavors with BRAC University. Along with the application, admissions officers enter the applicant’s personal information and educational history into the database.

After the applicant passes the admission test and pays the full amount of the admission fees to the university account, the student is issued a student ID according to the department and program that he registers for in that semester. Although, this ID is unique to each student, it may be subject to change if he decides to change programs at any point of time during his time with BRAC University.

The student then needs to fill out a detailed student information form and submit it to the registrar’s office along with his registration of courses for the registering semester. As the registrar’s office receives this document, the student’s information is then updated in the database and a full student profile is created.

This is the beginning of the flow of the student’s information in the current database. The student is then expected to move on with the advising process and select his courses through his assigned advisor.

1.1.1 Pre-Advising and Advising

Student Pre-Advising

For the pre-advising phase, a student logs into the website and is asked to enter his/her ID and password to log in to the system. Without logging in, the user can see class schedules, exam schedules, list of pre-requisite courses and seat status of the offered courses. When the student logs in to the system he/she is shown their grade sheet. The student can then view/modify his/her profile and choose to register in a maximum of four student clubs and forums. While logged in, the student can browse through a list of courses offered in that semester. Each student is allotted a maximum number of courses to register in a semester, based on his current CGPA.

Once, the student completes his pre advising, the selected courses are booked and the data is stored in the main database. The student then needs to wait for his assigned advising day in order to confirm the courses with his advisor and complete his registration for the selected courses.

During advising, the student physically meets his advisor and requests registration confirmation for his selected courses. Before, confirmation however, the advisor needs to check whether the student meets all the requirements needed to register for those courses. The requirements include not having any clash in class timings or exam schedules with the selected courses and completing all the pre-requisite courses if applicable. If the student fails to meet the requirements, then he is bound to change his selection of courses until he meets all the requirements. After ensuring that courses have been selected with all their respective requirements fulfilled, the advisor hands the student with a signed printed document of the advising record.

The student then calculates the total amount of fee he needs to pay in order to complete his registration. After the financial matters are settled, the student receives a receipt from the bank. The payment slip along with the advising form is taken to the Accounts section who verifies the payment done and keeps a portion of the advising form. The student then needs to submit this to the registrar’s office.

Information collected at the accounts department and the registrar’s office is then updated onto separate databases used by the two departments. These two management departments are kept completely in the dark with each other about student enrollment information. The database is only synchronized when members of the two departments schedule a physical meeting and match their records.

Problems in Student Advising Process

- Students are allotted a small specific time to complete their pre-advising.
- System is slow and irresponsive.
- Hectic to add and modify courses.
- Paper based matching of databases is used.
- No information about the course fees on the advising slip
- Class routine is not shown on the advising slip
- Class room or course facilitator information is unavailable

The Advisor’s End

Teachers/Advisors are provided with desktop software which allows them to view student records, history, pre-requisite status and course records to undergo the task of student advising. During student advising, the advisor checks the advising form to check if the student meets the required course requirement and checks if there is any clash in the class schedule and that the student has completed all the required pre-requisite courses. If the student fails to meet a certain requirement, the advisor can change that specific course so the students meets the requirements and that there is no routine clash. After verifying all this, the advisor takes a printout of the advising form and signs it.

Problems at the Advisor’s End

- The present software has high loading time
- The interface is not user friendly
- Searching is slow
- Class routine is not located in a convenient location
- Routine elements are cramped in small spaces

1.1.2 Administration

Registrar’s Office

The registrar’s office is in charge of the entire university management system. The employees over there are responsible for adding, modifying or deleting student, course and teacher related data to and from the system. They ensure that proper records are kept. They are also responsible for allotting the number of sections per course and seats per section. The office also creates the class routine for the entire university.

When a student submits their advising form to the registrar’s office, the course registration information is entered into the database and enrolment in the different courses is confirmed.

Problems at the Registrar’s Office

- Database synchronization is done manually
- Multiple entries for same data
- No Report generation
- No proper backup technique is used
- Database dumping is used for creating backup
- Old data is hard to retrieve
- Registrar does not have the most updated list of successfully registered students
- Database reaches congruency near the middle of the semester

Accounts

The accounts department has independent software and database. The database used has no link with the main university database. All the information is entered separately here again.

When a student submits the advising form along with the payment slip to the accounts office, the officer looks up the amount of fee that he has to pay per credit. He then calculates the total amount payable and then matches it with the amount paid. Once payment is clarified, the officer keeps a portion of the advising form for later use and returns the rest of it to the student. The information in the advising form is then manually entered into the account database and then verified with the registrar's office via a scheduled meeting. Paperwork is used to match all the relevant data.

Problems at the Accounts Department

- Register and accounts have different database
- Same data needs to be manually entered multiple times so that both party has the data
- Data is also manually verified amongst the multiple databases to ensure data integrity
- Very time consuming
- If a student changes a course after advising then the whole process of advising is repeated making it very tiresome and causing a big delay in confirming the final list of enrolled students in different courses
- No information about the course fees in the advising form, which causes confusion about the exact fee payable to the university
- Multiple edit phases for same data

1.2 Proposed System

After considering that the current system is partially desktop based and partially web based we have come to the conclusion that the new system to be developed should be fully web based. By making the new system web based we can ensure that anyone with proper rights can access the system from any computer. This would ensure that any manipulations made to the database at any point of time are effective immediately.

Barriers to this solution might have been the lack of availability of internet and the slow speed provided by the ISPs in our country. But now that ISPs are readily available and offering high speed internet at low prices, choosing to make a web based solution is a viable one.

We have also decided that the system would have one centralized database. This will ensure that all the required data is entered into the system once and there is no double entry. This would save a lot of resources. Also, if only one database is used by all the different types of users, then each type of user would easily be able to receive all his required information from one source. Since the same data will be shared throughout the system, disk space and time is conserved. Also there is no need to validate data amongst multiple systems through tedious manual paperwork.

1.3 Major Requirements

Our major requirements are as follows:

- The system should be reliable
- The system should be user friendly and have an easy to use interface
- It should avoid data repetition
- Manual entry should be minimized
- The system should be able to implement all tasks that are needed to be done in the student advising process of BRAC University.
- Searching for data must be fast, accurate, precise and easy
- Must provide data security
- Should be able to generate reports
- A proper data access method must be ensured
- A fast yet reliable and friendly data recording technique should be there
- The system should have back up data storage

CHAPTER II

DATABASE DESIGN

2.1 Background

BRAC University follows a particular pattern for storing data about the students, faculty and all course information. Ideally we wanted to work with the existing database structures with all the tables and data fields being the same as the currently implemented database. However, due to a lack of access to the current database, we had to formulate a dummy database to work on. Our dummy database is made purely through research and study of what information is needed inside a database rather than what is actually in it. Hence, there may be no similarity in the two databases.

The database design is based on our understanding of all the information needed in a university database.

2.2 Database E-R diagram

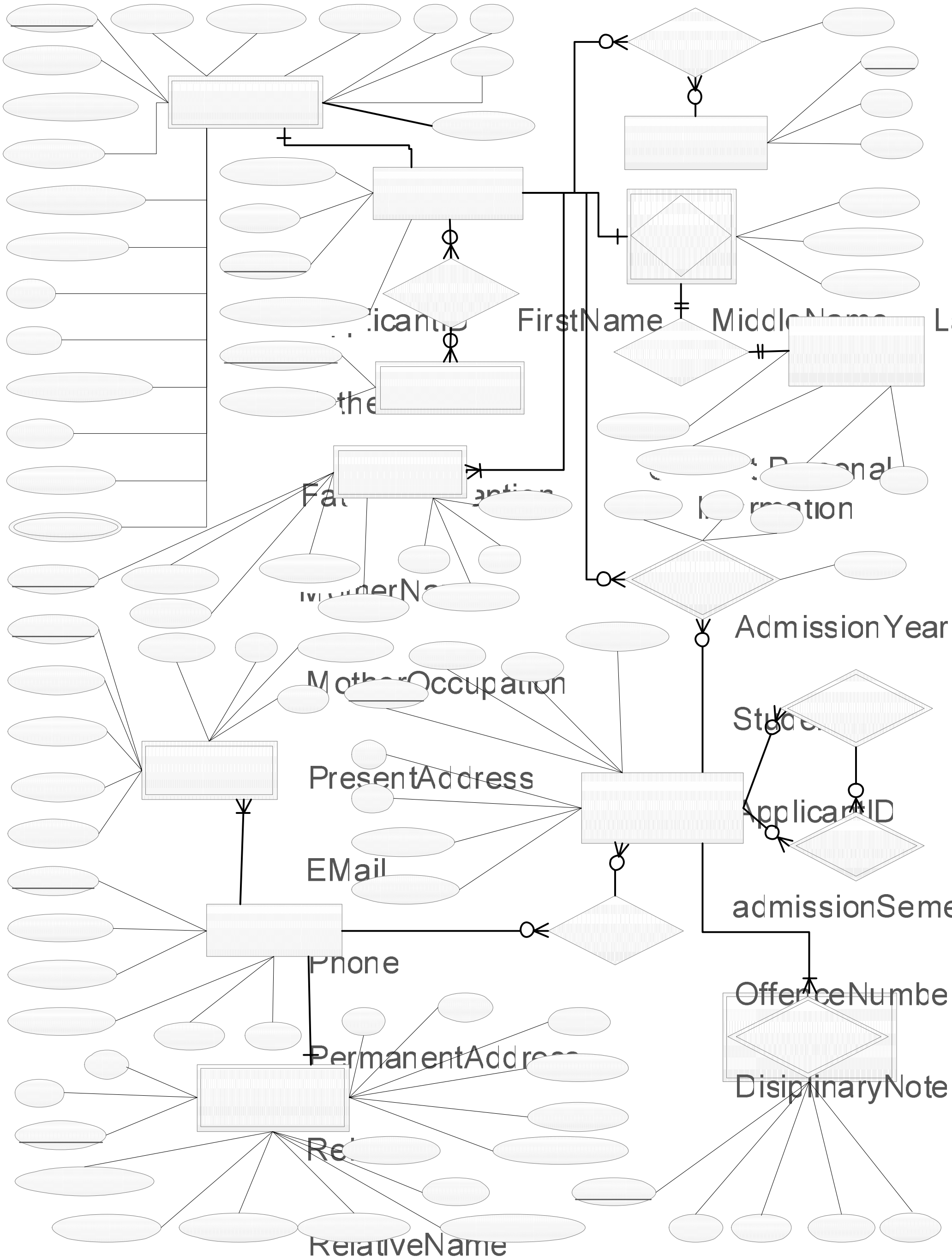


Figure 1: Entity Relationship Diagram

The above figure shows the database entity-relationship diagram. It shows all the tables in rectangles and all their fields linked to them in ovals. Each linking table is also connected in order to show the link.

2.3 Data Flow Diagrams (DFDs)

2.3.1 Context Level DFD

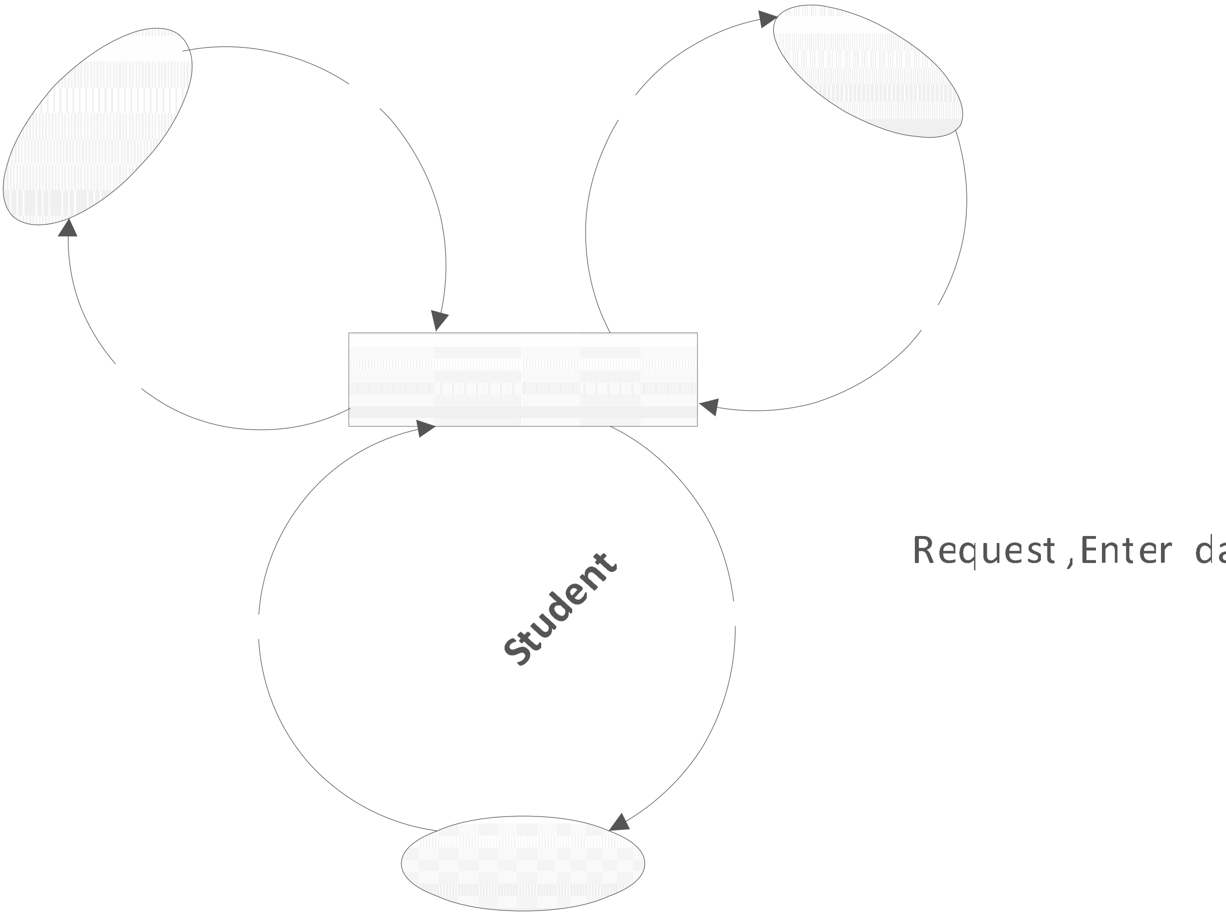


Figure 2: Context level Data Flow Diagram

In the proposed system three different types of users will interact with the database. The student will primarily view different information from the system and will be able to update their personal records. They can further choose to be a member of clubs to their liking and perform their pre-advising by choosing courses for the next semester.

The teacher will have the authority to view information regarding courses, clubs and students; however they will not be able to change those data. They will be able to update their personal records and perform different advising task.

The registrar will have full access to the system. They can view/edit/add/delete all records regarding student/ teacher/ club and courses.

2.3.2 DFD for Student

DFD Student – Level1

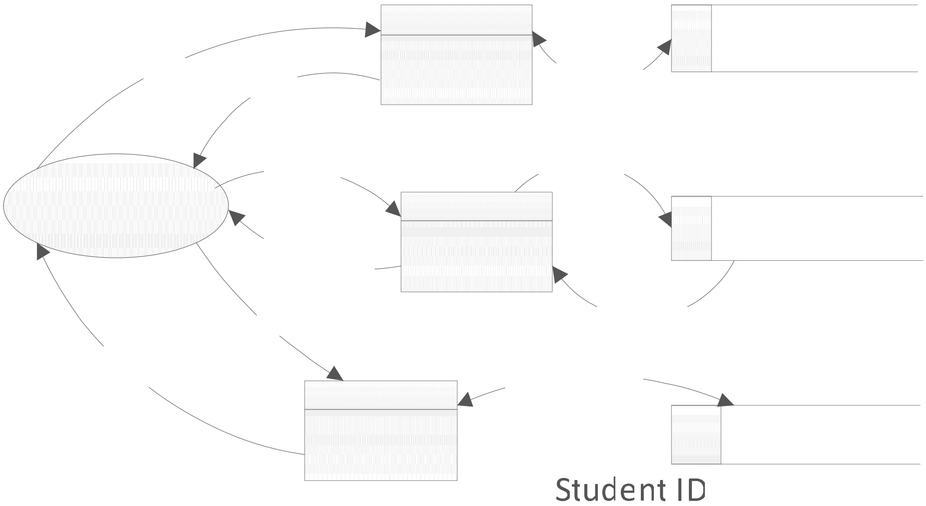


Figure 3: Level 1 Data Flow Diagram of Student

The student will send their Student Id to the system to access his/her personal information. They have to provide their Student Id and choose specific course to perform their pre- advising. For student to add club they just need to select a club and confirm it.

DFD Student – Level2: Process1

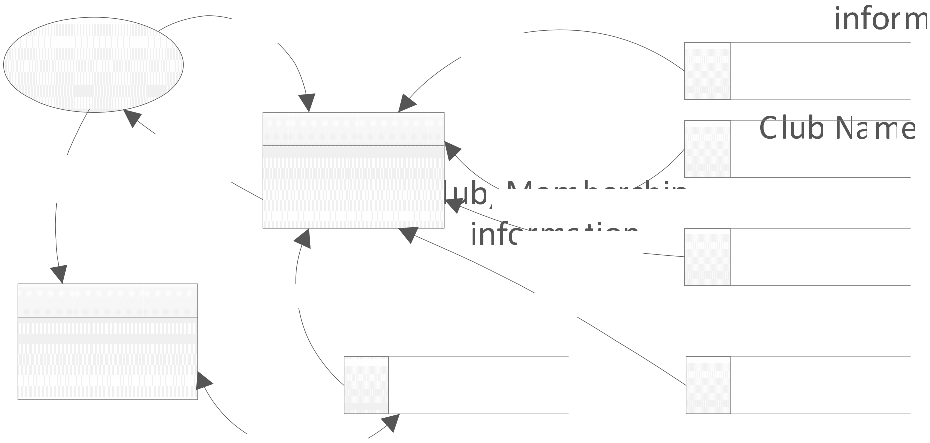


Figure 4: Level 2 Data Flow Diagram of Student for Process 1

A student will enter his/her Student ID and view their record they he will be presented with an option for editing his records. The student will be able to edit and update their basic information such as cell phone number, e-mail etc.

DFD Student – Level2: Process2

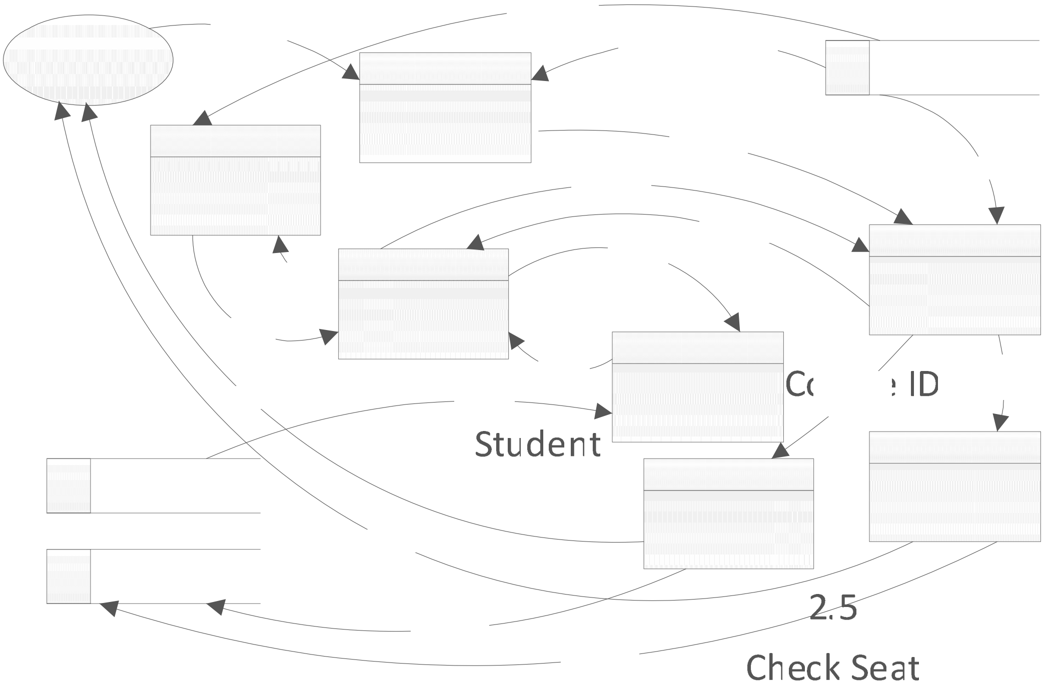


Figure 5: Level 2 Data Flow Diagram of Student for Process 2

A student will enter Course Id to check course information. This information will include course name, faculty, class and lab (conditional) routine. Once a student tries to enroll into a section of a course the system will check if he meets the course requirement and if there is clash in class routine. If all conditions are meeting then the student is enrolled in the course. The confirmation of enrolment is later done by the teacher. He will also be able to drop courses he has chosen during the time of pre-advising and add another.

DFD Student – Level2: Process3

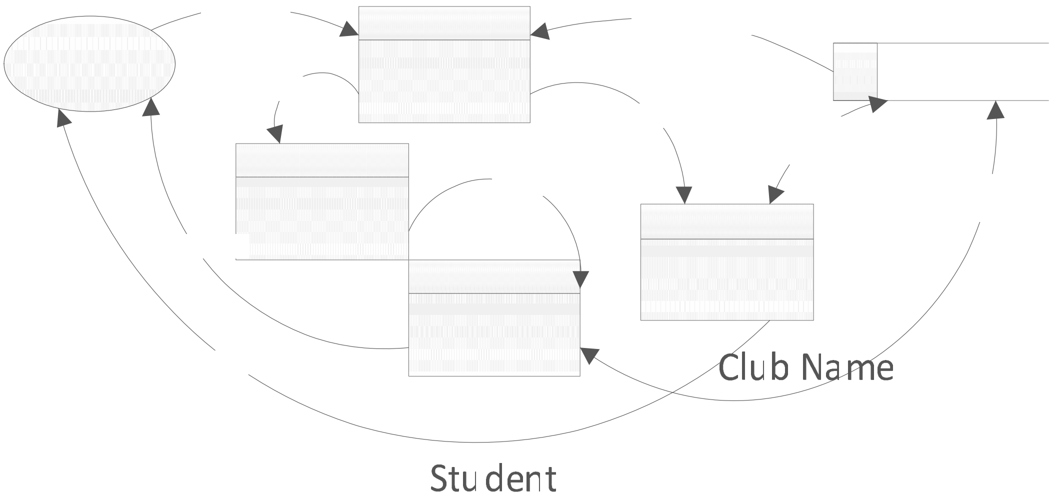


Figure 6: Level 2 Data Flow Diagram of Student for Process 3

A student will enter a Club name to view club information. If he decides to add a club, the system will check if he has already add the allotted number of club, if he has not than the system will add his ID to the club membership records. The student will also have the option to remove his name from a club.

2.3.3 DFD for Teacher

DFD Teacher – Level 1

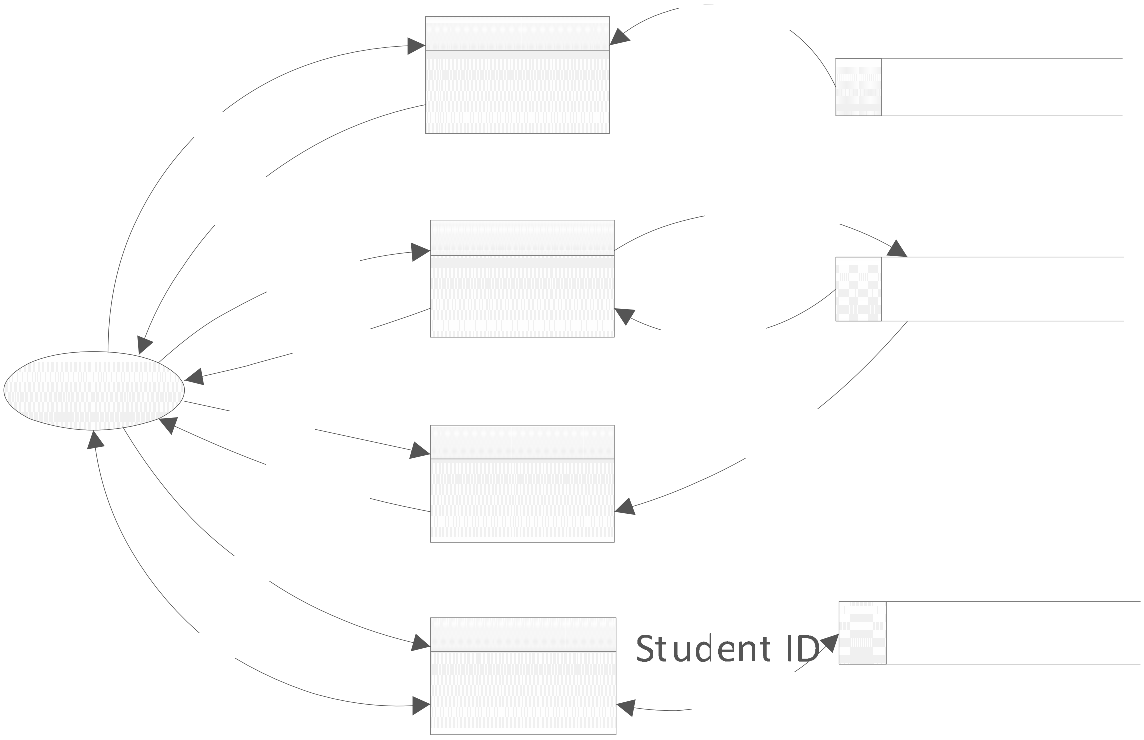


Figure 7: Level 1 Data Flow Diagram of Teacher

A teacher will enter their initials to the system to access his/her personal information and update it. They have to provide their Student Id and choose course for the student if required and confirm it. They will also be able to enroll students into courses despite the student not meeting the requirements. They will also be able to view full course information.

DFD Teacher – Level2: Process 1

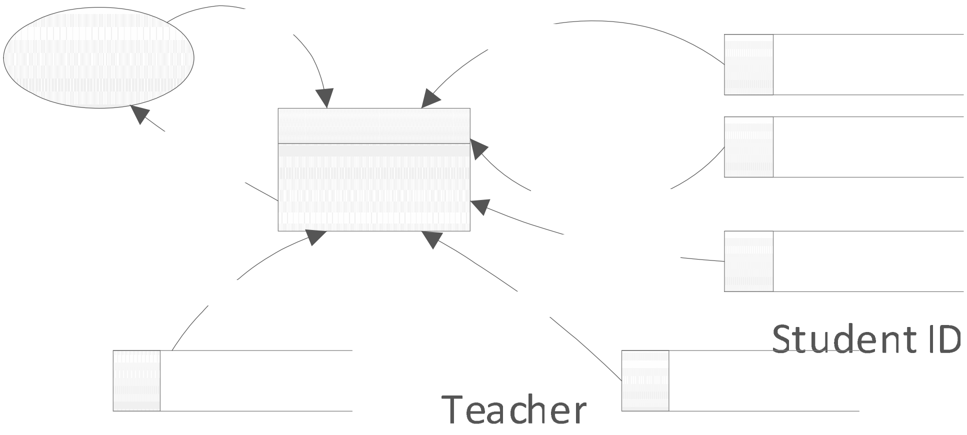


Figure 8: Level 2 Data Flow Diagram of Teacher for Process 1

A teacher will enter a Student ID and view their complete student records including enrollment history.

DFD Teacher – Level2: Process 2

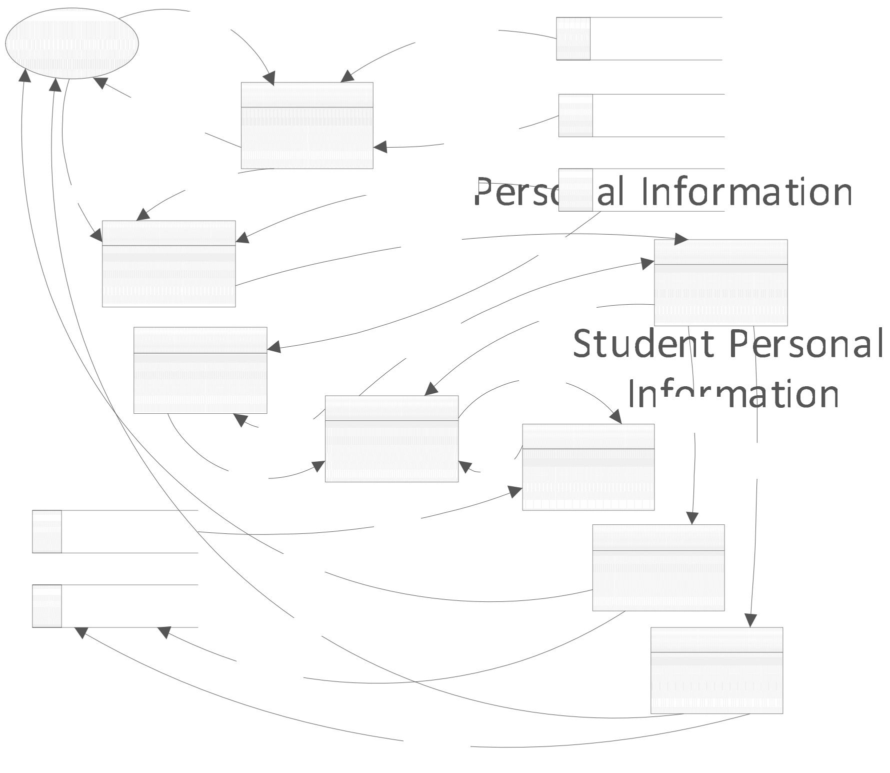


Figure 9: Level 2 Data Flow Diagram of Teacher for Process 2

A student will enter Course Id to check course information. This information will include course name, faculty, class and lab (conditional) routine. Once a student tries to enroll into a section of a course the system will check if he meets the course requirement and if there is clash in class routine. If all conditions are meeting then the student is enrolled in the course. The confirmation of enrolment is later done by the teacher. He will also be able to drop courses he has chosen and add another.

DFD Teacher – Level2: Process 3

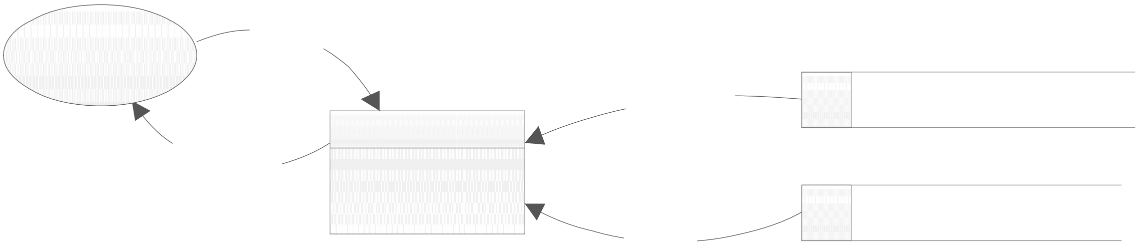


Figure 10: Level 2 Data Flow Diagram of Teacher for Process 3

A teacher will enter a Course ID and view complete course information. This information will include course faculty, timing, sections, class schedule, etc.

DFD Teacher – Level2: Process 4

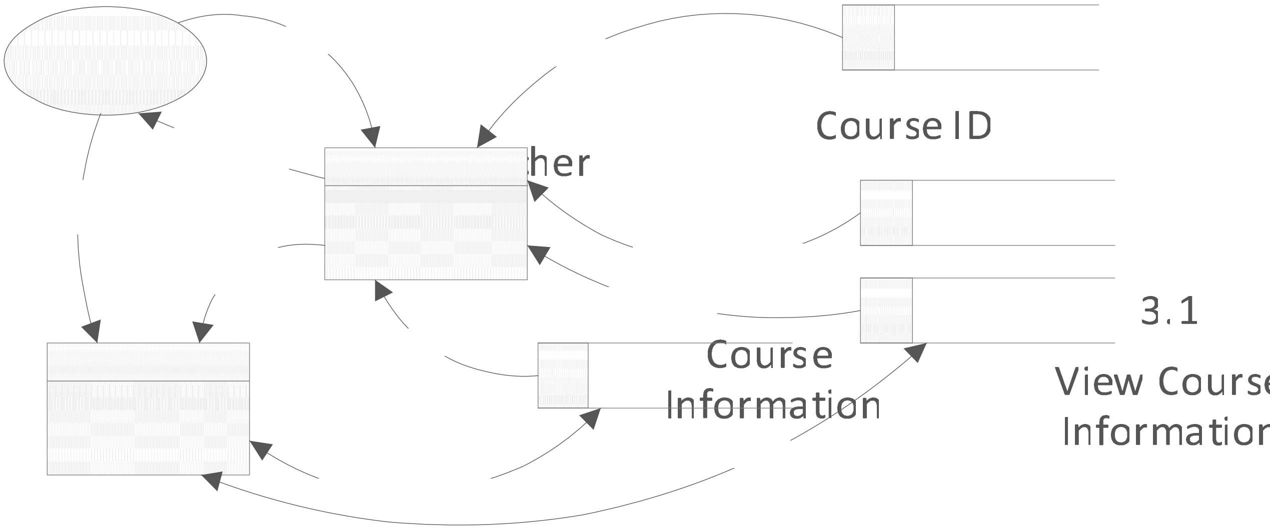


Figure 11: Level 2 Data Flow Diagram of Teacher for Process 4

A teacher will enter his/her initials and view their record they he will be presented with an option for editing his records. The teacher will be able to edit and update their basic information such as cell phone number, e-mail etc.

2.3.4 DFD for Registrar

DFD Registrar – Level1

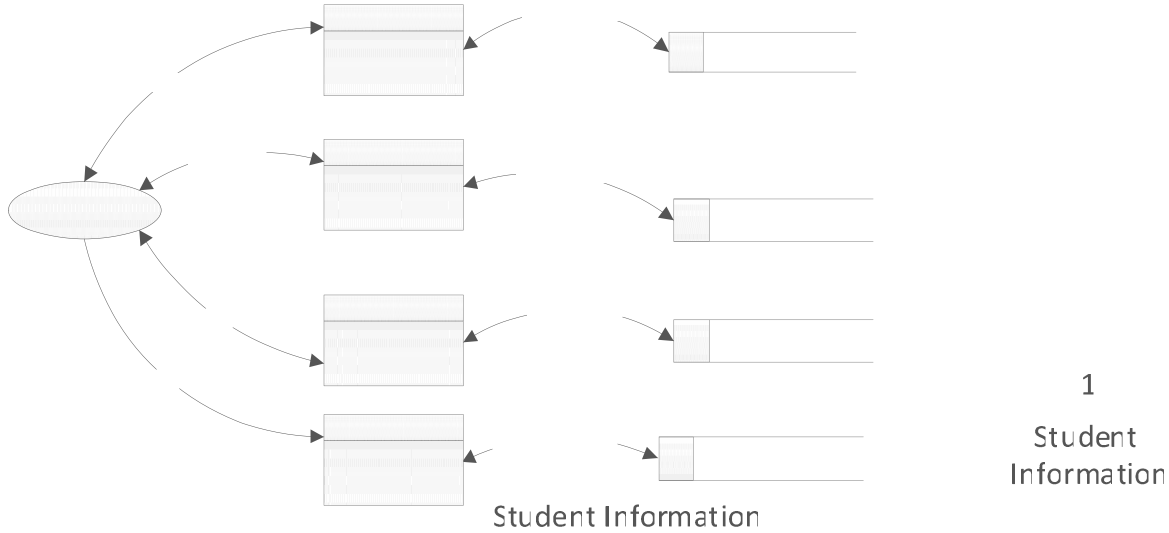


Figure 12: Level 1 Data Flow Diagram of Registrar

The registrar will have full access to the system. They can view/edit/add/delete all records regarding student, teacher, club and courses.

DFD Registrar – Level2: Process 1

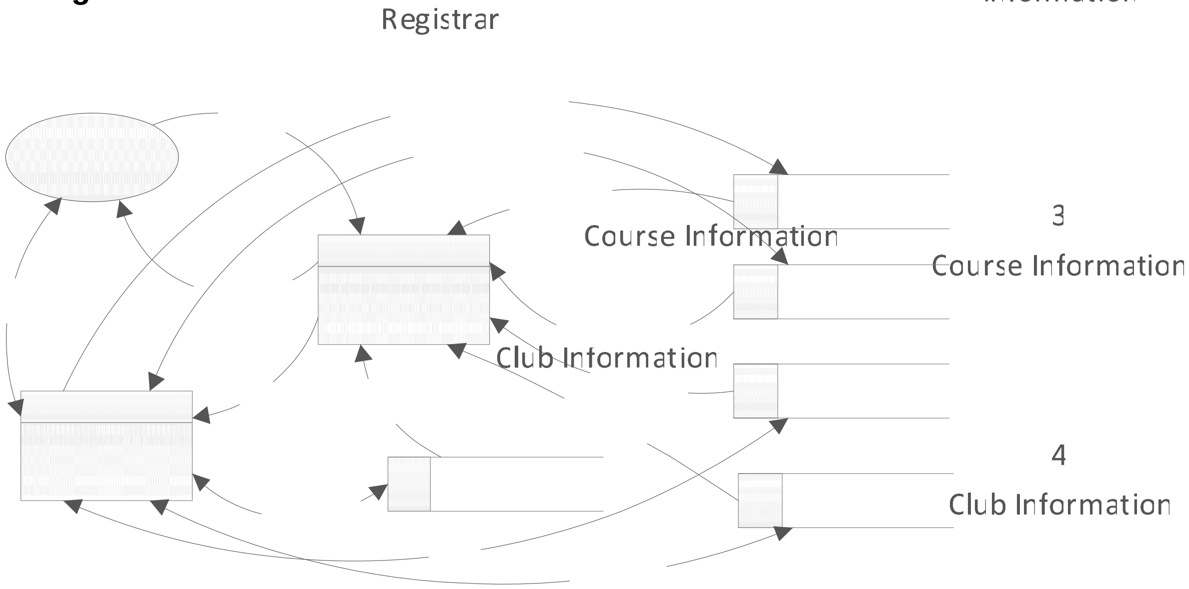


Figure 13: Level 2 Data Flow Diagram of Registrar for Process 1

The registrar will enter Student ID to view and edit existing records. They will also be allowed to delete if required or add new records during admission.

DFD Registrar – Level2: Process 2

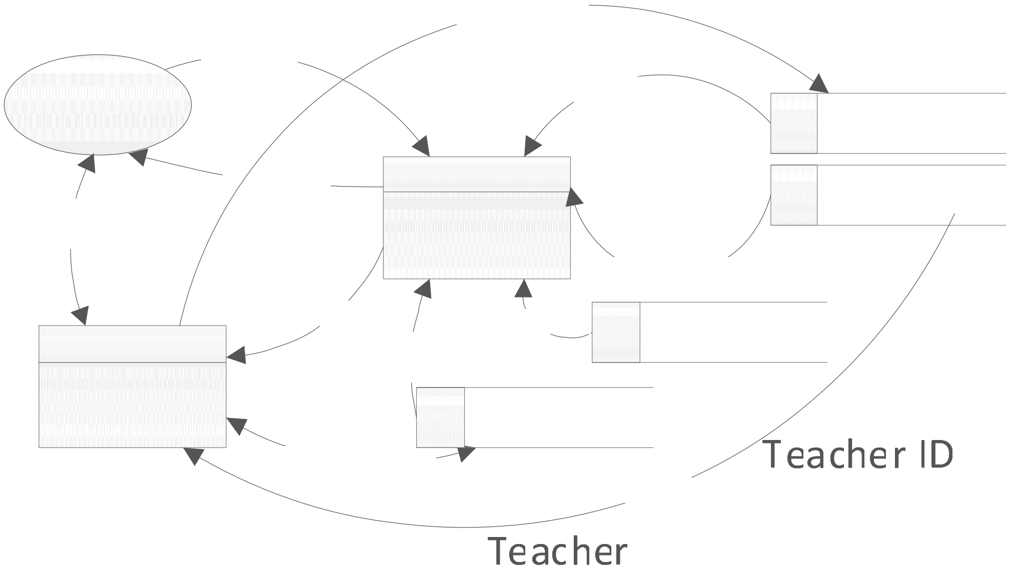


Figure 14: Level 2 Data Flow Diagram of Registrar for Process 2

The registrar will enter Student ID to view and edit existing records. They will also be allowed to delete if required or add new records of teacher upon joining.

DFD Registrar – Level2: Process 3

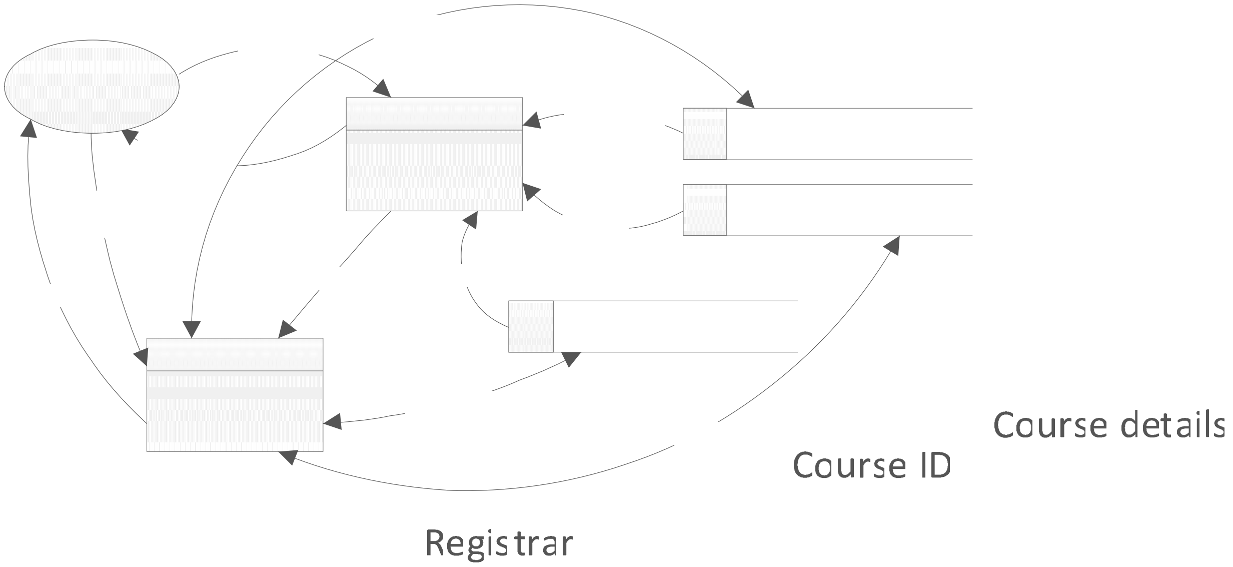


Figure 15: Level 2 Data Flow Diagram of Registrar for Process 3

The registrar will enter Course ID to view and edit existing records. They will enter which courses are to be offered and who will be the course teacher. They will also fix class routine and allot section and seat to different courses

DFD Registrar – Level2: Process 4

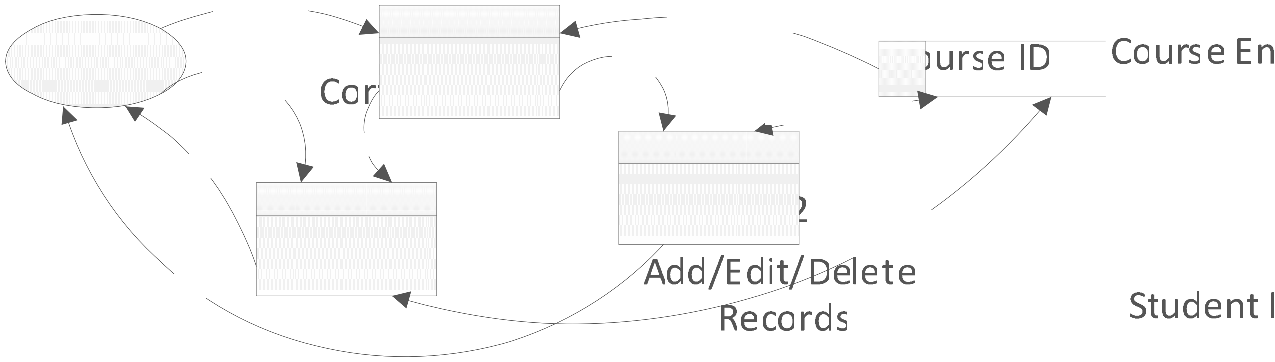


Figure 16: Level 2 Data Flow Diagram of Registrar for Process 4

The registrar will enter Club name to view and edit existing records. They will also be allowed to delete if required or add new records of club when a club is formed or disbanded.

2.4 Object classes with generalization hierarchy

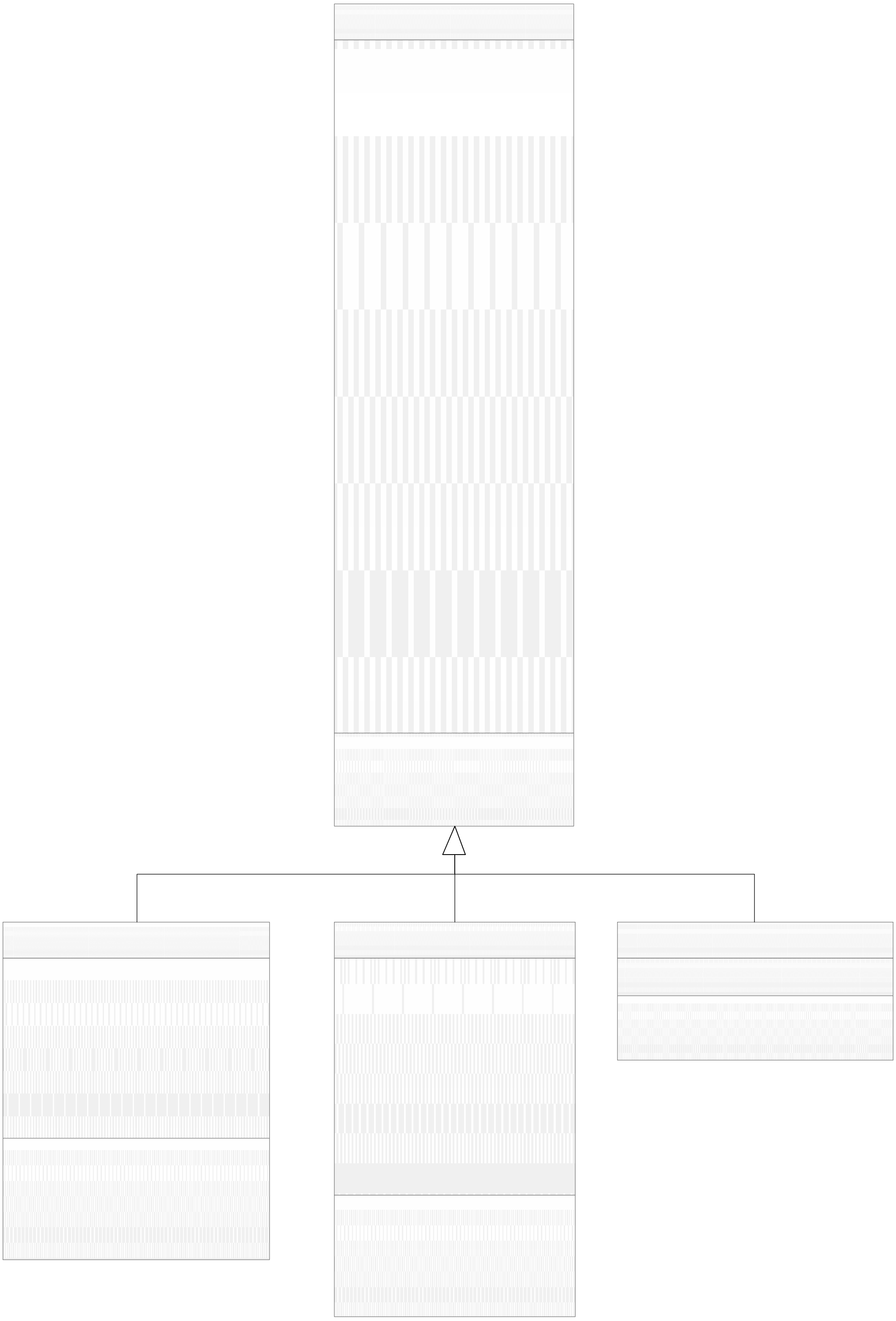


Figure 17: Object Class designs

firstNa
middle
lastNa
DOB
POB
gende
marita
father
father
mothe
mothe
phone
email
mailin
perma
relatio
relatio
relatio
degree
institu
board
passin
GPA
major

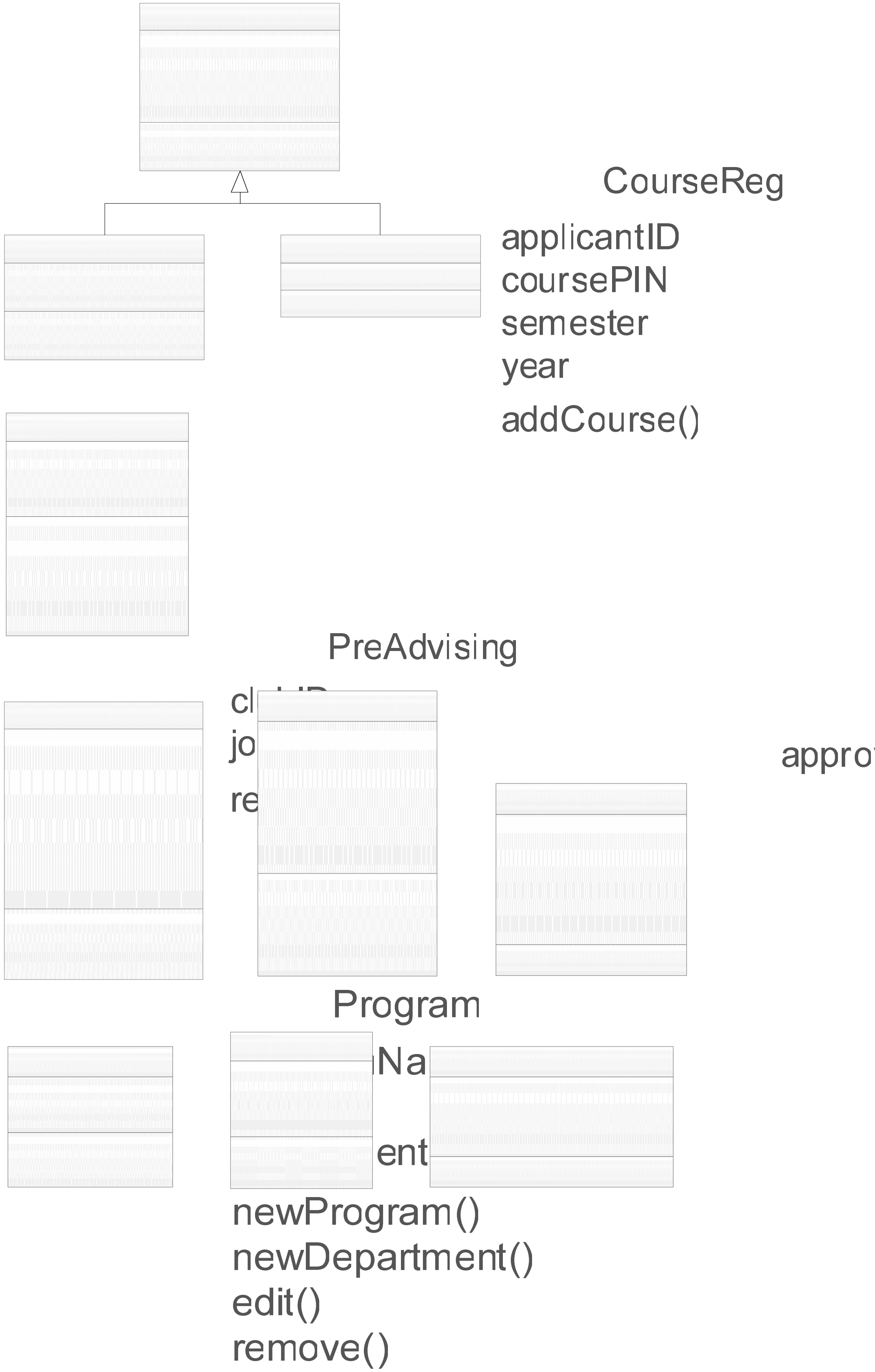

CourseReg
applicantID
coursePIN
semester
year
addCourse()
PreAdvising
cl
jo
re
appro
Program
Na
ent
newProgram()
newDepartment()
edit()
remove()

Figure 18 More Object Class designs

Figure 17 and 18 show a model of all the classes which will be used in the system designed.

2.5 Object association model

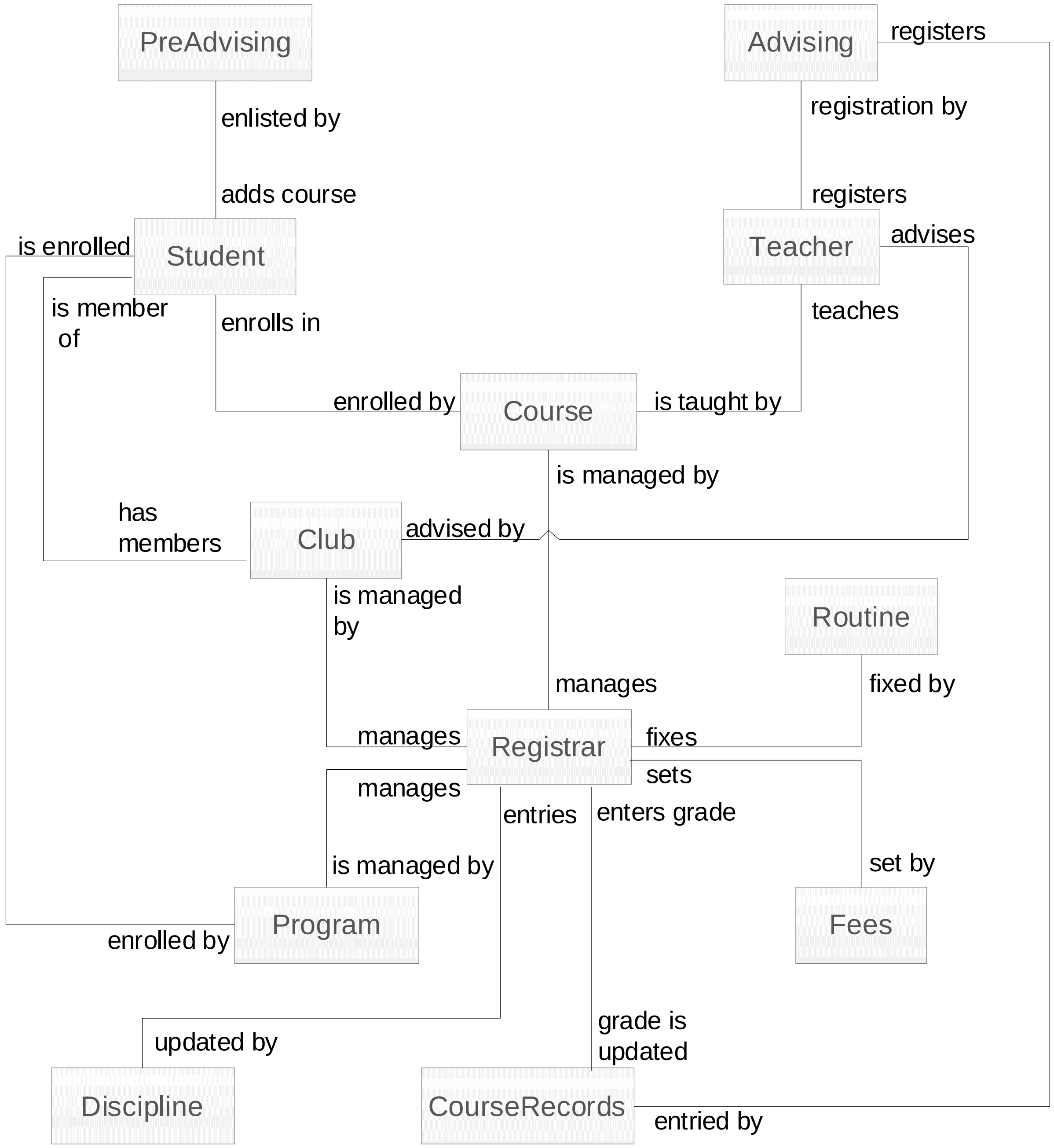


Figure 19: Object association model

The figure shows the associations of all the classes that are going to be used in the development of the system. It shows how the three main classes: Student, Teacher and Registrar will interact with the other classes and their functions.

2.6 Database tables with primary and foreign keys

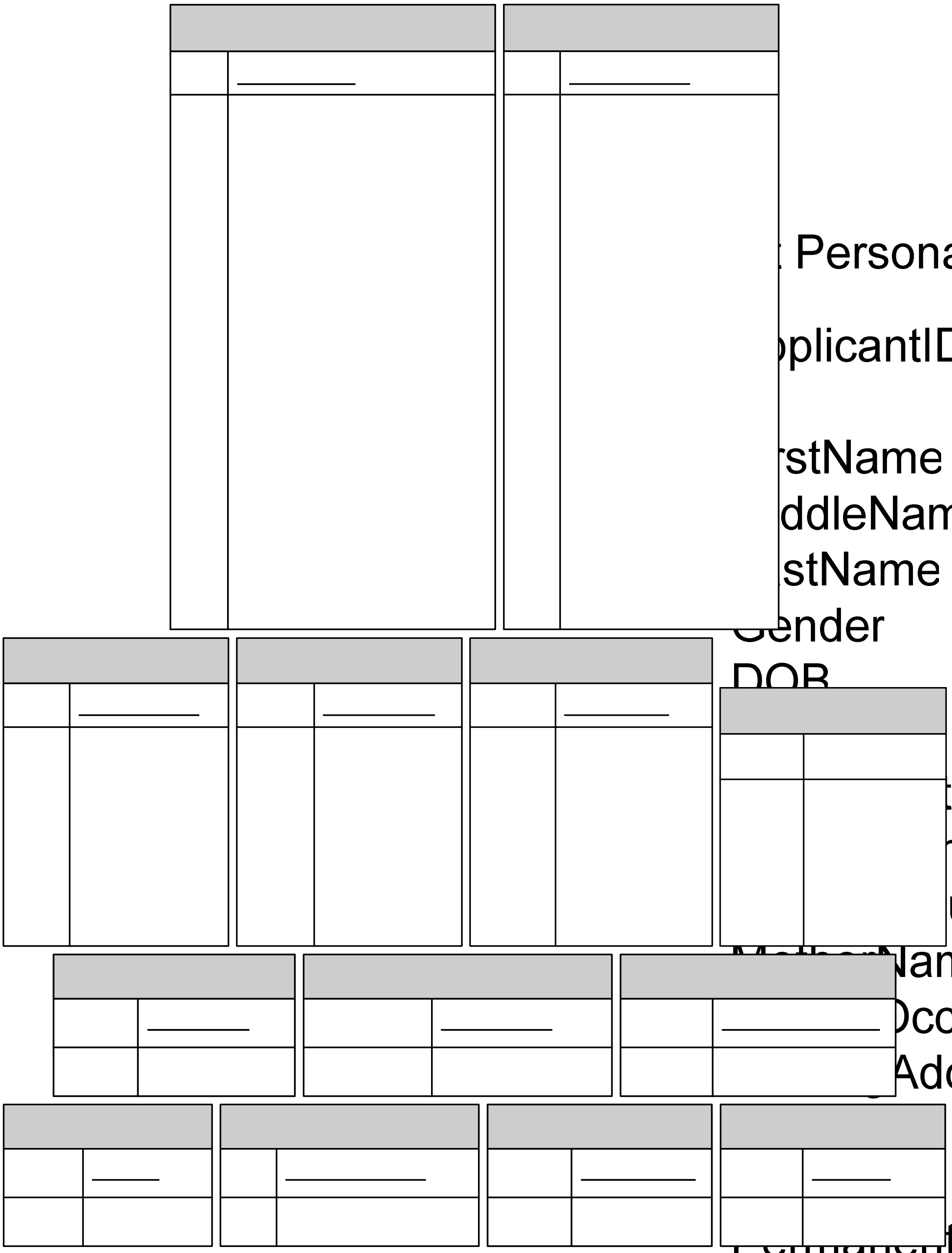
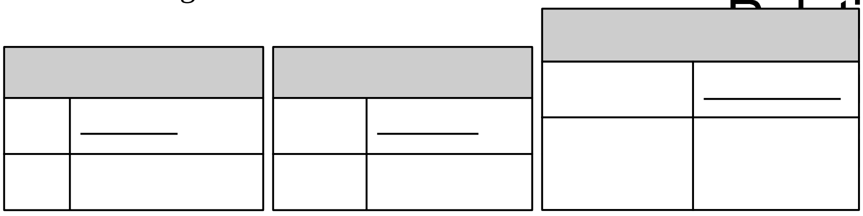
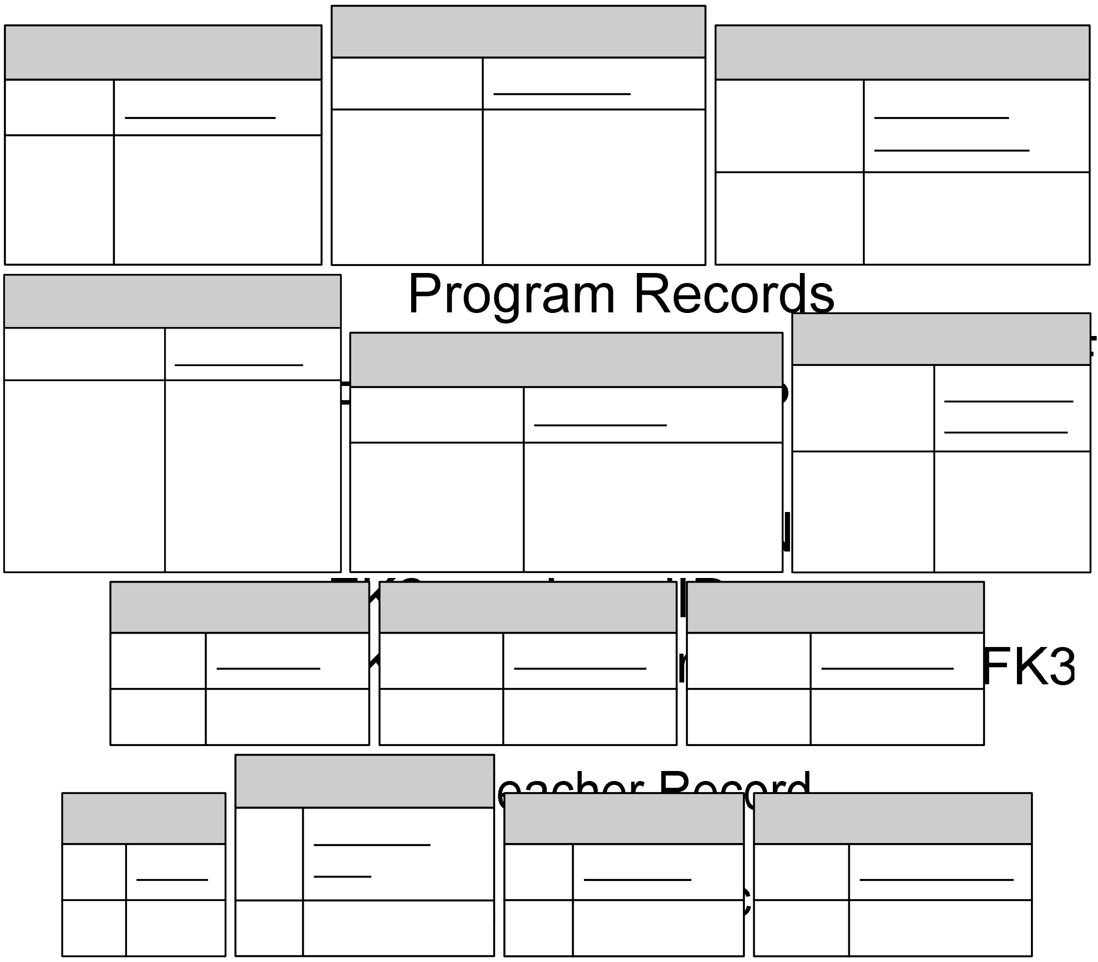


Figure 20: Database tables with PK and FK



Teacher Education		Student Education	
PK	TeacherPIN	PK,I1	ApplicantID
FK1	DegreeID	FK1	DegreeID



Studen

Stu

Figure 191: More database tables with PK and FK

	Initials
FK1,I1	DepartmentID
FK3,I2	PostID
FK2	JobID
FK6	StatusID

PK,FK1,FK2

I1

StudentStatusType

PK,I1	StatusID
I2	Status

La

PK,FK1

FK2,I1

Credit Fee

Class Day

PK Semester

PK Year

PK DayID

I1 Day

PerCreditFee

CHAPTER III

INTERFACE DESIGN

3.1 Overview

Before designing the interface we knew that it needed to be be user friendly and very easy to understand. The application is going to be fully web based and hence needs to be very fast. The main idea of replacing the current software is to ensure that students waste less time offline and that advisors take less time to complete the advising of each student. We also needed the availability of features such as grade sheet, and other background information of students available to advisors at ease.

There are certain features in the application which need to be accessible without signing in and certain features that will have different levels of functionality depending on the access rights of the signed user.

In order to fulfill our needs, we decided that all the different options should be linked with menu items. Hence, just below the header, we put a tab of all the menu items. Below that is a content pane which would change upon selection of the different menu tabs.

The different menu items inserted are:

- Home
- Personal Info
- Education History
- Grade Sheet
- Advising Sheet
- Courses Available
- Pre-requisites
- Faculty Info
- Club/Forums
- Class Schedule
- Exam Schedule
- Seat Status
- About

Each tab, upon click, loads a different page and loads the content below the menu tab.

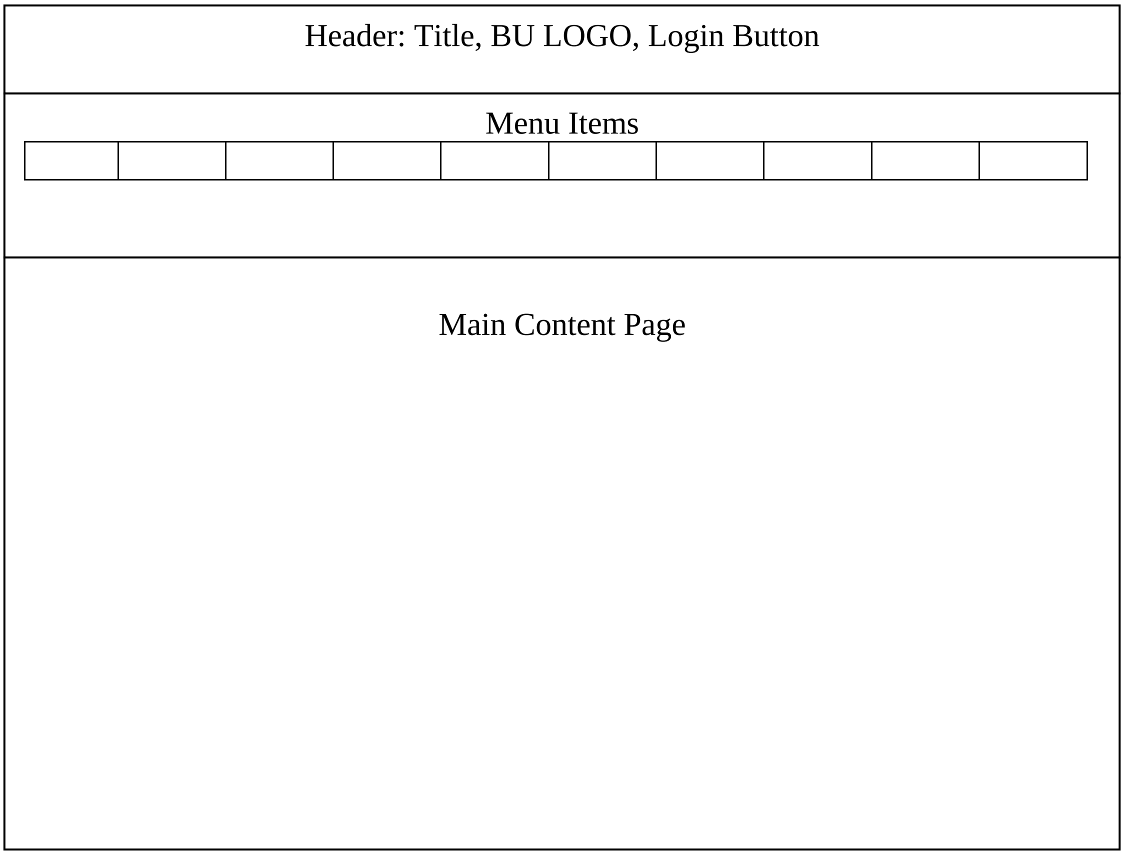
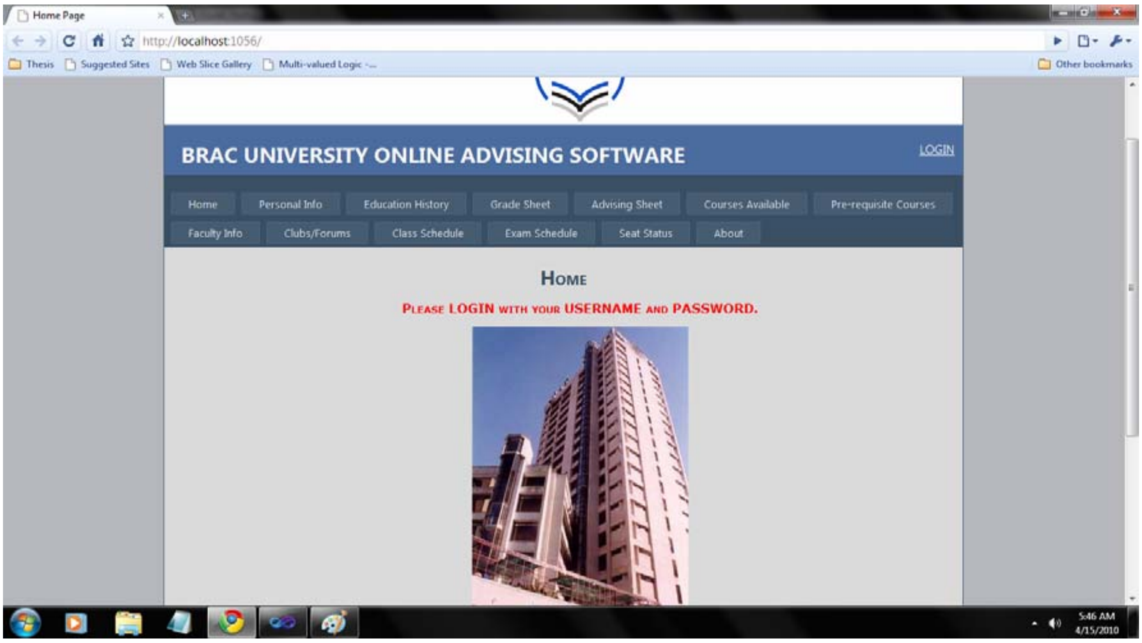


Figure22: Interface design layout

3.2 Interface Design Implementation

The interface was created on ASP.NET on a C# platform. This was ideal, because ASP.NET includes the use of all HTML tags and extra control tags which can be used to control the webpage. Further details of ASP.NET are stated on the latter chapters.



BRAC UNIVERSITY ONLINE ADVISING SOFTWARE

HOME

PLEASE LOGIN WITH YOUR USERNAME AND PASSWORD.

Figure23: Interface layout on implementation

CHAPTER IV

LANGUAGES USED FOR IMPLEMENTATION

4.1 .NET Framework

The Microsoft .NET Framework is a software framework that can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a Microsoft offering and is intended to be used by most new applications created for the Windows platform.

The framework's Base Class Library provides a large range of features including user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. The class library is used by programmers, who combine it with their own code to produce applications.

Programs written for the .NET Framework execute in a software environment that manages the program's runtime requirements. Also part of the .NET Framework, this runtime environment is known as the Common Language Runtime (CLR). The CLR provides the appearance of an application virtual machine so that programmers need not consider the capabilities of the specific CPU that will execute the program. The CLR also provides other important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

Interoperability

Because interaction between new and older applications is commonly required, the .NET Framework provides means to access functionality that is implemented in programs that execute outside the .NET environment. Access to COM components is provided in the System.Runtime.InteropServices and

System.EnterpriseServices namespaces of the framework; access to other functionality is provided using the P/Invoke feature.
Common Runtime Engine

Common Runtime Engine

The Common Language Runtime (CLR) is the virtual machine component of the .NET framework. All .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling.

Language Independence

The .NET Framework introduces a Common Type System, or CTS. The CTS specification defines all possible data types and programming constructs supported by the CLR and how they may or may not interact with each other conforming to the Common Language Infrastructure (CLI) specification. Because of this feature, the .NET Framework supports the exchange of types and object instances between libraries and applications written using any conforming .NET language.
Base Class Library

Base Class Library

The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages using the .NET Framework. The BCL provides classes which encapsulate a number of common functions, including file reading and writing, graphic rendering, database interaction, XML document manipulation and so on.

Simplified Deployment

The .NET framework includes design features and tools that help manage the installation of computer software to ensure that it does not interfere with previously installed software, and that it conforms to security requirements.

Security

The design is meant to address some of the vulnerabilities, such as buffer overflows, that have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.

Portability

The design of the .NET Framework allows it to theoretically be platform agnostic, and thus cross-platform compatible. That is, a program written to use the framework should run without change on any type of system for which the framework is implemented. While Microsoft has never implemented the full framework on any system except Microsoft Windows, the framework is engineered to be platform agnostic, and cross-platform implementations are available for other operating systems (see Silverlight and the Alternative implementations section below). Microsoft submits the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language), the C# language, and the C++/CLI language to both ECMA and the ISO, making them available as open standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms.

4.2 .NET Framework Advantages

The .NET Framework offers a number of advantages to developers. The following paragraphs describe them in detail.

4.2.1 Consistent Programming Model

Different programming languages have different approaches for doing a task. For example, accessing data with a VB 6.0 application and a VC++ application is totally different. When using different programming languages to do a task, a disparity exists among the approach developers use to perform the task. The difference in techniques comes from how different languages interact with the underlying system that applications rely on.

With .NET, for example, accessing data with a VB .NET and a C# .NET looks very similar apart from slight syntactical differences. Both the programs need to import the System.Data namespace, both the programs establish a connection with the database and both the programs run a query and display the data on a data grid. The VB 6.0 and VC++ example mentioned in the first paragraph explains that there is more than one way to do a particular task within the same language. The .NET example explains that there's a unified means of accomplishing the same task by using the .NET Class Library, a key component of the .NET Framework.

The functionality that the .NET Class Library provides is available to all .NET languages resulting in a consistent object model regardless of the programming language the developer uses.

4.2.2 Direct Support for Security

Developing an application that resides on a local machine and uses local resources is easy. Consider an application that accesses data on a remote machine or has to perform a privileged task on behalf of a no privileged user. In this scenario security is much more important as the application is accessing data from a remote machine.

With .NET, the Framework enables the developer and the system administrator to specify method level security. It uses industry-standard protocols such as TCP/IP, XML, SOAP and HTTP to facilitate distributed application communications. This makes distributed computing more secure because .NET developers cooperate with network security devices instead of working around their security limitations.

4.2.3 Simplified Development Efforts

With classic ASP, when a developer needs to present data from a database in a Web page, he is required to write the application logic (code) and presentation logic (design) in the same file. He was required to mix the ASP code with the HTML code to get the desired result.

ASP.NET and the .NET Framework simplify development by separating the application logic and presentation logic making it easier to maintain the code. You write the design code (presentation logic) and the actual code (application logic) separately eliminating the need to mix HTML code with ASP code. ASP.NET can also handle the details of maintaining the state of the controls, such as contents in a textbox, between calls to the same ASP.NET page.

Another advantage of creating applications is debugging. Visual Studio .NET and other third party providers provide several debugging tools that simplify application development. The .NET Framework simplifies debugging with support for Runtime diagnostics. Runtime diagnostics helps you to track down bugs and also helps you to determine how well an application performs. The .NET Framework provides three types of Runtime diagnostics: Event Logging, Performance Counters and Tracing.

4.2.4 Easy Application Deployment and Maintenance

The .NET Framework makes it easy to deploy applications. In the most common form, to install an application, all you need to do is copy the application along with the components it requires into a directory on the target computer. The .NET Framework handles the details of locating and loading the components an application needs, even if several versions of the same application exist on the target computer. The .NET Framework ensures that all the components the application depends on are available on the computer before the application begins to execute.

4.3 ASP

4.3.1 What is ASP?

- ASP stands for Active Server Pages
- ASP is a Microsoft Technology
- ASP is a program that runs inside IIS
- IIS stands for Internet Information Services

4.3.2 What is an ASP File?

- An ASP file is just the same as an HTML file
- An ASP file can contain text, HTML, XML, and scripts
- Scripts in an ASP file are executed on the server
- An ASP file has the file extension ".asp"

4.3.3 How Does ASP Differ from HTML?

- When a browser requests an HTML file, the server returns the file
- When a browser requests an ASP file, IIS passes the request to the ASP engine. The ASP engine reads the ASP file, line by line, and executes the scripts in the file. Finally, the ASP file is returned to the browser as plain HTML

4.3.4 What can ASP Do?

- Dynamically edit, change, or add any content of a Web page
- Respond to user queries or data submitted from HTML forms
- Access any data or databases and return the results to a browser
- Customize a Web page to make it more useful for individual users
- The advantages of using ASP instead of CGI and Perl, are those of simplicity and speed
- Provide security - since ASP code cannot be viewed from the browser
- Clever ASP programming can minimize the network traffic

4.3.5 Lifetime of Variables

A variable declared outside a procedure can be accessed and changed by any script in the ASP file. A variable declared inside a procedure is created and destroyed every time the procedure is executed. No scripts outside the procedure can access or change the variable. To declare variables accessible to more than one ASP file, declare them as session variables or application variables.

Session Variables

Session variables are used to store information about ONE single user, and are available to all pages in one application. Typically information stored in session variables are name, id, and preferences.

Application Variables

Application variables are also available to all pages in one application. Application variables are used to store information about ALL users in one specific application.

The Session object

When you are working with an application on your computer, you open it, do some changes and then you close it. This is much like a Session. The computer knows who you are. It knows when you open the application and when you close it. However, on the internet there is one problem: the web server does not know who you are and what you do, because the HTTP address doesn't maintain state.

ASP solves this problem by creating a unique cookie for each user. The cookie is sent to the user's computer and it contains information that identifies the user. This interface is called the Session object.

The Session object stores information about, or change settings for a user session.

Variables stored in a Session object hold information about one single user, and are available to all pages in one application. Common information stored in session variables are name, id, and preferences. The server creates a new Session object for each new user, and destroys the Session object when the session expires.

4.4 Advantages of Using ASP.NET

- ASP.NET drastically reduces the amount of code required to build large applications
- ASP.NET makes development simpler and easier to maintain with an event-driven, server-side programming model
- ASP.NET pages are easy to write and maintain because the source code and HTML are together
- The source code is executed on the server. The pages have lots of power and flexibility by this approach
- The source code is compiled the first time the page is requested. Execution is fast as the Web Server compiles the page the first time it is requested. The server saves the compiled version of the page for use next time the page is requested
- The HTML produced by the ASP.NET page is sent back to the browser. The application source code you write is not sent and is not easily stolen
- ASP.NET makes for easy deployment. There is no need to register components because the configuration information is built-in
- The Web server continuously monitors the pages, components and applications running on it. If it noticies memory leaks, infinite loops, other illegal software or activities, it seamlessly kills those activities and restarts itself
- ASP.NET validates information (validation controls) entered by the user without writing a single line of code
- ASP.NET easily works with ADO .NET using data-binding and page formatting features
- ASP.NET applications run faster and counters large volumes of users without performance problems

4.5 Differences between ASP.NET and Client-Side Technologies

Client-side refers to the browser and the machine running the browser. Server-side on the other hand refers to a Web server.

4.5.1 Client-Side Scripting

Javascript and VBScript and generally used for Client-side scripting. Client-side scripting executes in the browser after the page is loaded. Using client-side scripting you can add some cool features to your page. Both, HTML and the script are together in the same file and the script is download as part of the page which anyone can view. A client-side script runs only on a browser that supports scripting and specifically the scripting language that is used. Since the script is in the same file as the HTML and as it executes on the machine you use, the page may take longer time to download.

4.5.2 Server-Side Scripting

ASP.NET is purely server-side technology. ASP.NET code executes on the server before it is sent to the browser. The code that is sent back to the browser is pure HTML and not ASP.NET code. Like client-side scripting, ASP.NET code is similar in a way that it allows you to write your code alongside HTML. Unlike client-side scripting, ASP.NET code is executed on the server and not in the browser. The script that you write alongside your HTML is not sent back to the browser and that prevents others from stealing the code you developed.

4.6 C#

C# is a multi-paradigm programming language encompassing imperative, functional, generic, object-oriented (class-based), and component programming disciplines. It was developed by Microsoft within the .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270). C# is one of the programming languages designed for the Common Language Infrastructure.

4.7 What are the advantages of C# over VB.NET

- XML documentation generated from source code comments. (This is coming in VB.NET with Whidbey (the code name for the next version of Visual Studio and .NET), and there are tools which will do it with existing VB.NET code already.)
- Operator overloading - again, coming to VB.NET in Whidbey.
- Language support for unsigned types (you can use them from VB.NET, but they aren't in the language itself). Again, support for these is coming to VB.NET in Whidbey.
- The using statement, which makes unmanaged resource disposal simple.
- Explicit interface implementation, where an interface which is already implemented in a base class can be reimplemented separately in a derived class. Arguably this makes the class harder to understand, in the same way that member hiding normally does.
- Unsafe code. This allows pointer arithmetic etc, and can improve performance in some situations. However, it is not to be used lightly, as a lot of the normal safety of C# is lost (as the name implies). Note that unsafe code is still managed code, i.e. it is compiled to IL, JITted, and run within the CLR.

4.8 Java vs C#/.NET

4.8.1 Java platform Pros

- Fully cross platform APIs including support for Graphics, Windowing and Multimedia. All future APIs will be cross platform. Java has even more APIs than Windows. Look at the current proposals.
- Mature.
- LOTS AND LOTS of libraries available. Many are free and/or open source.
- High density of OO design patterns put to practise compared to C#/.NET. This makes developing complex applications much easier (at the cost of making 'my first text editor' type projects slightly more difficult).
- Well designed APIs following good OO practise.
- The Java Community Process allows people with an investment in Java have a say. New Java features can be proposed, discussed and voted for in a open democratic forum. Microsoft does NOT have anything like this which is unfortunate for anyone investing in .NET.
- Apache.
- Currently supported by multiple commercial vendors (not one).
- Although not recognised by any international standards committee, Java is an open standard governed by Sun and the JCP members (or which there are MANY). C# is an ECMA standard but without support from Microsoft, very few third parties have been willing to implement it. Sun also tends to be less secretive than Microsoft about their products. The C#/CLI ECMA submission is also just a subset of the .NET framework. It has yet to be seen whether any future APIs for .NET will ever be submitted for standardisation (it doesn't look good).
- Smallest unit of distribution is a class (.class file). This helps in application development, debugging and deployment. Having a one to one mapping between types and the file system is simpler because file systems are well understood.
- Good IDEs are available (Netbeans, Eclipse, Together). Many are free and support more features than VS.NET. Unfortunately many are slower than VS.NET.
- Swing is well designed and utilises the MVC design pattern extensively. Windows Forms pales in comparison.

- Swing is more powerful than SWF(System.Windows.Forms). For example, border styles can be plugged into any component using the strategy design pattern. In Windows Forms, components are responsible for supplying and drawing their own borders. Some support borders, some don't, and some only supply a few borders. This is because SWF is a thin wrapper around 7 year old windows controls and doesn't seem to have any improvements over WFC in J++. Infact it's lacking some features that were present in WFC like DHTML support.
- Java developers are more likely to be University graduates with a strong understanding of OOP.

Unlike Microsoft, Sun understands and embraces OOP. Microsoft prefers speed over good design (at least some of their employees seem to anyway).

4.8.2 C#/.NET Pros

- The CLR is well designed and arguably better than the JVM (pity about some of the class library)
- The CLR supports execution of both managed and unmanaged code.
- The CLR was designed to be language agnostic and has extended instructions to support non-imperative languages such as mondrian (a functional language) and mercury (a logical language). *Ironically*, Java currently supports more programming languages.
- The **best** way to write Windows desktop and web applications.
- The remoting framework is advanced and well designed.
- Good support for taking advantage of the native OS. [This isn't so good if you want to be cross-platform].
- Metadata attributes are a VERY, VERY powerful feature. Attributes allow classes to be much more expressive. Java NEEEEEDS attributes.
- SWF is faster than Swing (most of the time). Unfortunately the design is lacking when compared to Swing. My guess is it was designed with VB programmers in mind :-(. Windows Forms does have powerful support for DataBinding. If you're writing data bound rich clients, Windows Forms will get the job done quicker.
- The proposed generics support is vastly superior to the upcoming Java 1.5 implementation.
- XML documentation tags are cool. (We have JavaDoc tags to thank for promoting the idea).

- The CLI is an ECMA standard. Unfortunately, this isn't as useful as it appears since many of the .NET class libraries are missing from the standard. When Java gets new APIs, they become a standard that other vendors can implement (there are even reference implementations). This has not and most likely will not happen with .NET.
- The *System.CodeDom* namespace is cool.
- Supports ValueTypes (structs) as first class citizens. "Int" is considered an object.
- Access to native code is easy. A lot of thought has been given to the runtime marshalling infrastructure.

When overriding virtual methods you have to explicitly state that you're overriding with the *overrides* keyword.

CHAPTER V

OTHER LANGUAGE ALTERNATIVES

5.1 PHP:

PHP is a scripting language designed to fill the gap between SSI (Server Side Includes) and Perl, intended largely for the web environment. PHP has gained quite a following in recent times, and it is one of the forerunners in the Open Source software movement. Its popularity derives from its C-like syntax, its speed and its simplicity. PHP is currently divided into two major versions: PHP 4 and PHP 5, although PHP 4 is deprecated and is no longer developed or supplied with critical bug fixes. PHP 6 is currently under development.

Basically, PHP allows a static web document to become dynamic. "PHP" is a recursive acronym that stands for "PHP: Hypertext Preprocessor". PHP preprocesses (that is, PHP processes before the output is sent to the browser) hypertext documents. Because of this, the pages can change before the user sees them, based on conditions. This can be used to write something to the page, create a table with a number of rows equal to the number of times the user has visited, or integrate the web page with a web database, such as MySQL.

5.1.1 Advanced PHP

Advanced PHP includes high level programming and PHP techniques designed to make PHP even more useful and powerful.

5.1.2 Advantages of Using PHP:

- PHP is accessible
- It's available for free
- It's available with documentation in many languages
- There are many support groups, forums, and teams supporting PHP

- There is a wealth of online information regarding PHP
- It's quick to develop in PHP
- A basic PHP script can be created without a firm understanding of programming principals, compilation, and other currently important programming concepts
- PHP is loosely typed, which makes basic scripts much faster to develop with less attention to design
- Programmers of Java, PERL, BASIC, and other popular languages can find many parallels to ease transition to PHP
- PHP is flexible. Use OOP or not. Use naming convention(s) or not
- It runs on many different operating systems
- It can be optimized, even "compiled" for performance closer to that of more established compiled languages
- Open Source, readily available (you can be using it today) and dual-licensed - if you are doing non-profit work or not licensing, there is no cost.
- Very Easy to understand Syntax, some really cool features (arrays are something else!)
- Interfaces very easily with Apache/MySQL
- Server side (no need to futz with client installs - only with rendering issues if you use CSS).
- Lots of good source code out there to use and/or learn from, as well as many useful libraries for working with PDFs, graphics, etc.
- Lots of good books and on-line help (php.net is great)
- Platform agnostic, can run on Windows Linux or Mac servers. Also very scalable.
- Lots of hosting services have it ready to use, no special configuration (except if you have special security needs)
- Pretty easy to access other web-based tools through PHP (i.e. google maps, etc.)

PHP doesn’t use a lot of the system’s resources so it runs fast and doesn’t tend to slow other processes down. It is typically used as an Apache module, written in C, so it loads and executes quickly. It works well with other software and can be quite fast. PHP is also fairly stable and since it is open source, the PHP community works together to fix any bugs. The community offers technical support and continuously updates the code further expanding PHP's capabilities.

PHP offers many levels of security to prevent malicious attacks. These security levels can be adjusted in the .ini file.

Another key advantage of PHP is its connective abilities. PHP uses a modular system of extensions to interface with a variety of libraries such as graphics, XML, encryption, etc. In addition, programmers can extend PHP by writing their own extensions and compiling them into the executable or they can create their own executable and load it using PHP's dynamic loading

5.1.3 Disadvantages of PHP:

- Out of the box, PHP tends to execute more slowly than assembly, C, and other compiled languages
- PHP is loosely typed. For developers of all skill levels, this allows room for unexpected behavior due to programmer error that many other languages might not permit. [Of course, few if any languages can protect from developer error!]
- There are many ways to do one thing and many cases where a function has ambiguous handling due to legacy support or PHP development history.
  - If you want to do more than just HTML/CSS pages on the client you need to also add javascript, java or other client-side language in your output (goes for Perl and some other languages).
  - The way browsers work make handling data and coding programs more interesting (technically each page is a new run on the system, so you have to manage your variables coming in and going out and between pages. Not hard, just different.)
  - Web programming (regardless of language) is open to security flaws due to unimplemented or unknown vulnerabilities, takes a bit more caution.

5.2 SilverLight:

Microsoft Silverlight :

Microsoft Silverlight is a web application framework that provides functionalities similar to those in Adobe Flash, integrating multimedia, graphics, animations and interactivity into a single runtime environment. Initially released as a video streaming plugin, later versions brought additional interactivity features and support for CLI languages and development tools. The current version 4 was released to developers on April 12, 2010, and the end-user runtime will be released on April 15, 2010.

It is compatible with multiple web browser products used on Microsoft Windows, Linux (using Novell Moonlight), and Mac OS X operating systems. Mobile devices, starting with Windows Phone 7 and Symbian (Series 60) phones, will likely become supported in 2010. A free software implementation named Moonlight, developed by Novell in cooperation with Microsoft, is available to bring compatible functionality to Linux, FreeBSD and other open source platforms.

Silverlight provides a retained mode graphics system similar to Windows Presentation Foundation, and integrates multimedia, graphics, animations and interactivity into a single runtime environment. In Silverlight applications, user interfaces are declared in Extensible Application Markup Language (XAML) and programmed using a subset of the .NET Framework. XAML can be used for marking up the vector graphics and animations. Silverlight can also be used to create Windows Sidebar gadgets for Windows Vista.

Silverlight Architecture

The essential architecture and components of Microsoft Silverlight is not only an appealing canvas for displaying rich and interactive Web and media content to end users. It is also a powerful yet lightweight platform for developing portable, cross-platform, networked applications that integrate data and services from many sources. Furthermore, Silverlight enables you to build user interfaces that will significantly enhance the typical end user experience compared with traditional Web applications.

While Silverlight as a client-side runtime environment seems simple and compact in size, the Silverlight development platform integrates a number of features and complex technologies, making them accessible for developers. To create effective Silverlight-based applications, developers need a working knowledge of the platform architecture.

Advantages:

- Silverlight offers cross-domain support for applications
- Silverlight plugin is just a 1MB download and is supported in windows XP and above.
- Silverlight also has a browser plugin but still needs to reach at pace with flash.
- supports asynchronous calls
- Benefits of Ajax

Disadvantages:

- One issue to consider is that Silverlight is still a very new technology, with a few bugs to iron out (search SO for the silverlight tag to find a couple - list box resizing is one of the most egregious).
- Does not provide direct communication with SQL server. Instead WCF and LINQ to SQL is used as a media between the two to get/set data into the server.

5.3 Ajax:

Ajax is based on open standards so it is widely accepted and supported. Bandwidth usage of web is less as specific demands are accomplished on done on Ajax. It has full support of flash and flex. Ajax also has support for asynchronous calls

Disadvantages of Ajax

Ajax also uses JavaScript, CSS and XHTML but there are an issue of cross browser compatibility and what work on one browser might not work on another browser. This causes debugging to be very difficult and time consuming.

5.4 Flash:

Flash actually has the power of animation. The flash player used to run the Flash/ Flex apps are installed in about a billion computers. Flash is a multi-platform application that will appear the same on all browsers and platforms as long as it’s plug-in has be downloaded.

Flash should be used when the primary goal is visual effects to the users. As our project need to maintain a lot of data we have decide not to use flash.

CHAPTER VI

SECURITY FEATURES

6.1 Custom Generated Security

- User accounts are implemented to restrict the access of data.
- Each account is given account type and each account type given a specified level of access depending on the user therefore the user can only view data relevant to him/her.
- Measures are taken to control login Sessions thus preventing direct access from previously saved link.
- The database is designed to operated in such a way that it will only run in servers with specific custom setting. Therefore even if the database is somehow compromised extracting the data will be very difficult.
- All Session Variables are clearly dealt with when a user logs out therefore minimizing security risk.

LIST OF REFERENCES

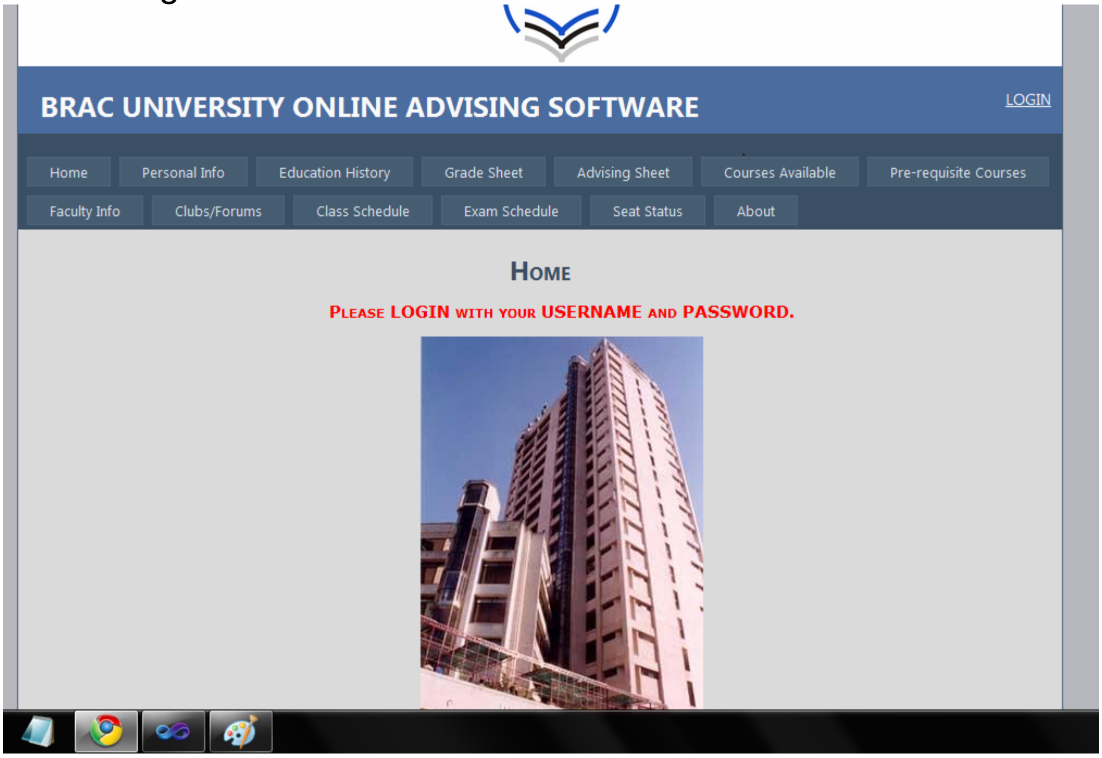
LIST OF REFERENCES

ASP.NET WEB BLOG:
http://weblogs.asp.net/scottgu/archive/2006/02/24/ASP.NET-2.0-Membership_2C00_-Roles_2C00_-Forms-Authentication_2C00_-and-Security-Resources-.aspx
OFFICIAL ASP.NET SITE: http://www.asp.net
MSDN BLOG: http://blogs.msdn.com/csharpfaq/archive/2004/03/11/87816.aspx
http://www.jonasjohn.de/snippets/csharp/sql-connection-example.htm
OFFICIAL MSDN SITES:
http://msdn.microsoft.com/en-us/library/bb386927.aspx
http://msdn.microsoft.com/enus/library/system.web.ui.webcontrols.gridview.selectedvalue(VS.100).aspx
OFFICIAL MICROSOFT SITE: http://support.microsoft.com/kb/910440
SILVERLIGHT FORUM: http://forums.silverlight.net/forums/p/29977/97518.aspx
http://www.experts-exchange.com/Programming/Languages/.NET/ASP.NET/Q_23456300.html
JAVA TUTORIALS:
http://www.java2s.com/Tutorial/CSharp/0100__String/0200__String-Compare.htm
SQL BLOG: http://blog.sqlauthority.com/2007/06/21/sql-server-retrieve-current-date-time-in-sql-server-current_timestamp-getdate-fn-now/
http://www.bigresource.com/MS_SQL--Binding-a-SqlDataSource-to-a-TextBox-or-label--Hr8puJRW.html#Ss7bPj8x
http://oreilly.com/windows/archive/csharp-regular-expressions.html

USER MANUAL

User Manual

Home Page:



BRAC UNIVERSITY ONLINE ADVISING SOFTWARE

HOME

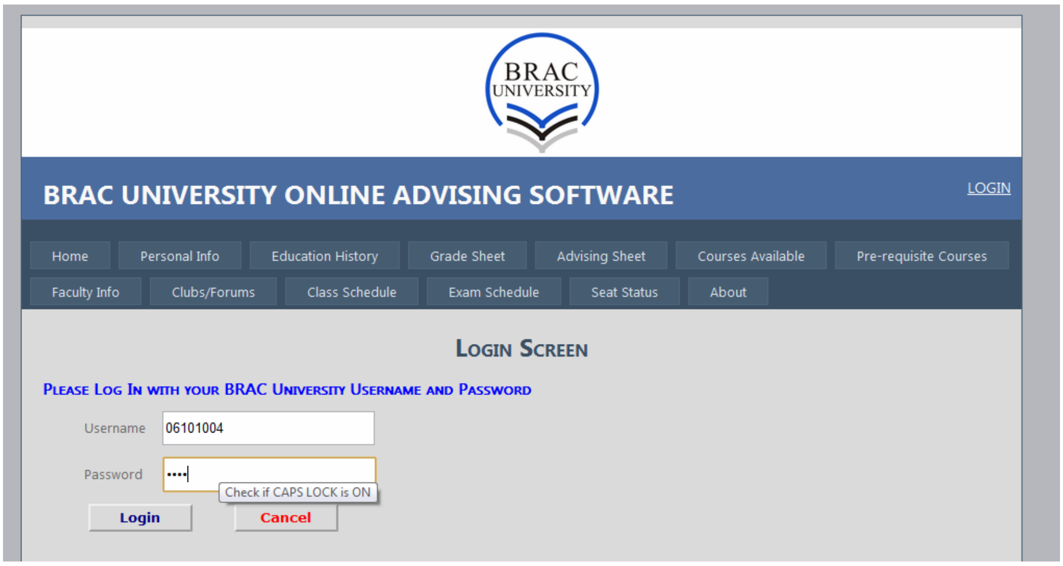
PLEASE LOGIN WITH YOUR USERNAME AND PASSWORD.

Please enter the website address in your internet browser. After browser loads, this is the page you are greeted with.

To login press the login link on the upper right corner. To view other pages select them from the menu under the heading BRAC UNIVERSITY ONLINE ADVISING SOFTWARE”. The pages on the top line are user specific pages that will only open after you have logged in. The pages on the bottom line are pages that anyone can access and view.

Login page:

This is the login page. Enter you username and password to get access to the complete website.



Login Screen

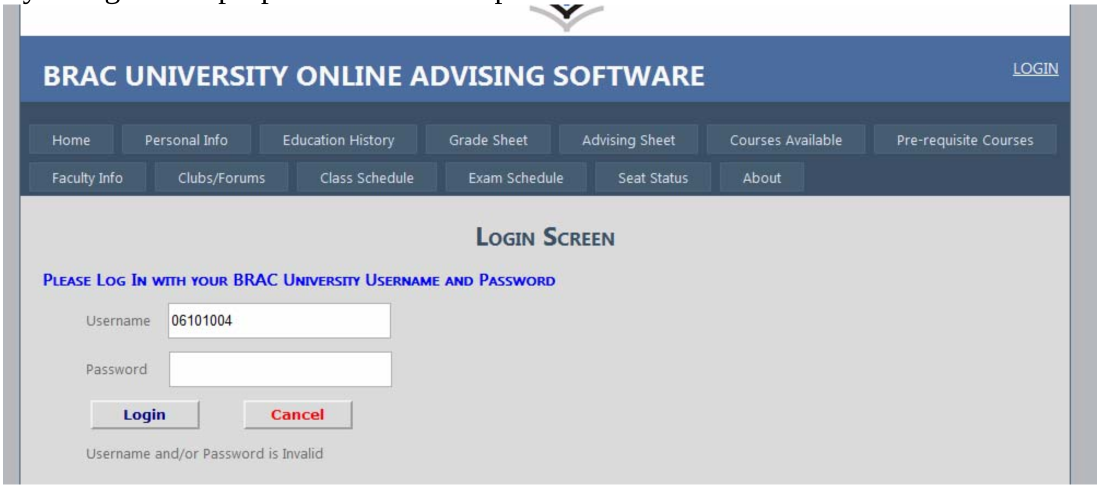
Please Log In with your BRAC University Username and Password

Username 06101004

Password ••••

Check if CAPS LOCK is ON

If wrong username/password is entered you will be given this message. You can again try to login with proper username and password.



Login Screen

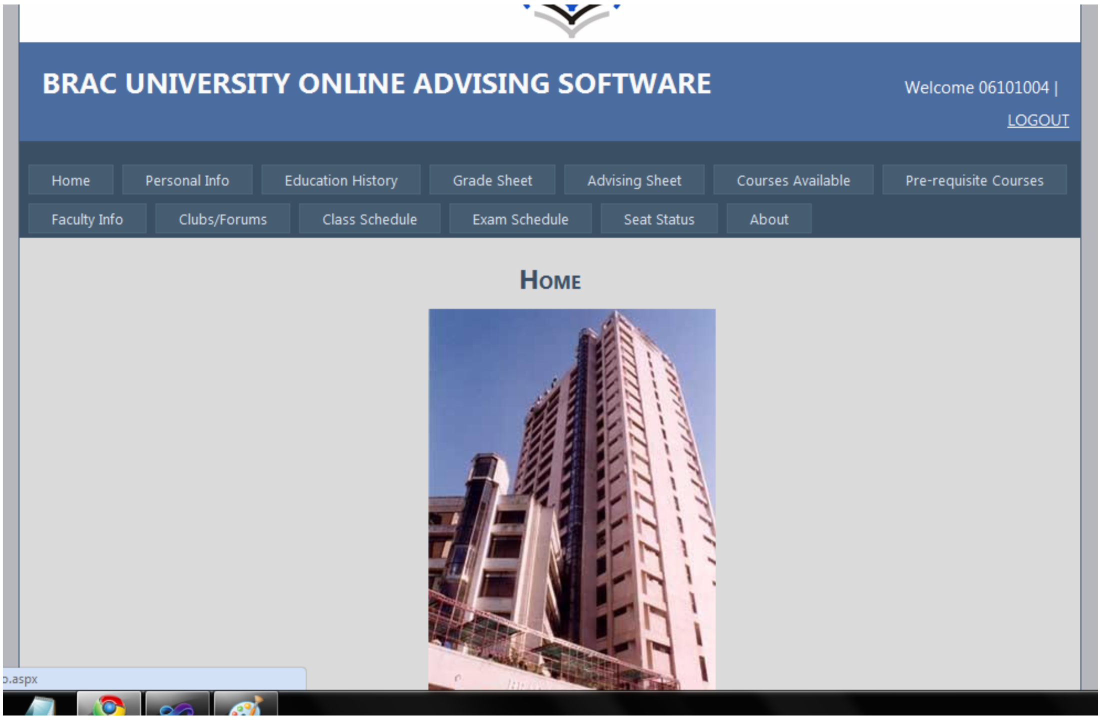
Please Log In with your BRAC University Username and Password

Username 06101004

Password

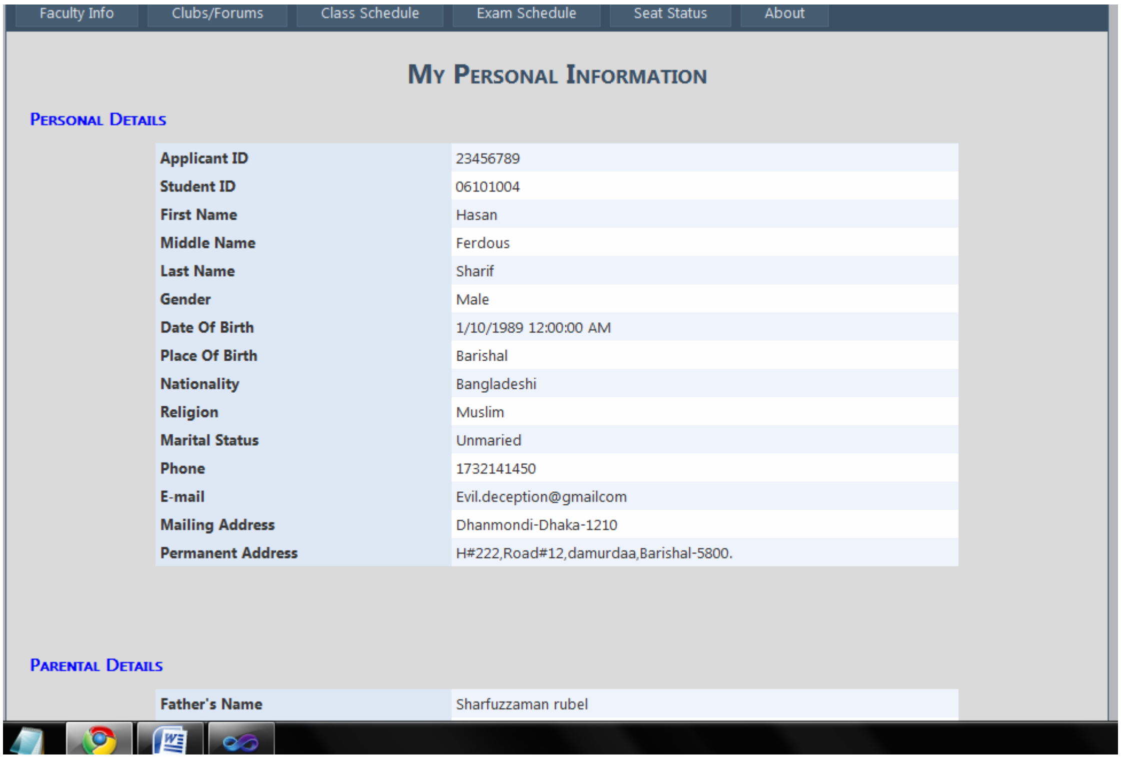
Username and/or Password is Invalid

Upon entering correct username & passsword the main page will be automatically redirected and you will not see the message "Please Login with your Username and Password". On the top-right side the Login link with be replaced by Logout link and the current User Id will be displayed. Now the user has access to view all the paages available.



Personal Info:

The personal info displays the personal information of the user who is currently logged in. For a student it will display his/her personal information. If the user is a faculty member they will see their respective personal information.



My Personal Information

Personal Details

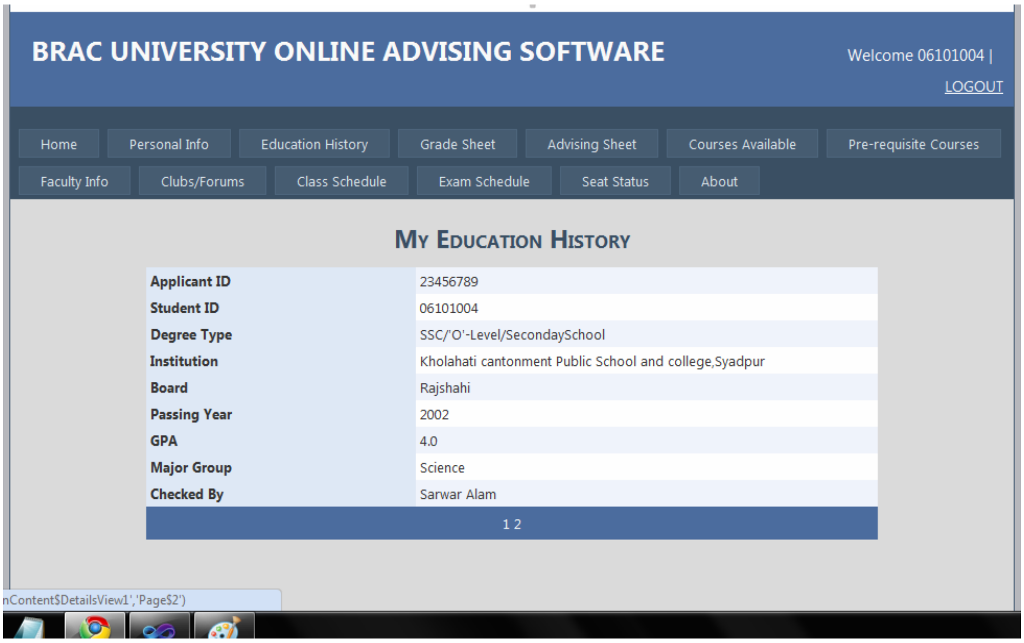
Applicant ID	23456789
Student ID	06101004
First Name	Hasan
Middle Name	Ferdous
Last Name	Sharif
Gender	Male
Date Of Birth	1/10/1989 12:00:00 AM
Place Of Birth	Barishal
Nationality	Bangladeshi
Religion	Muslim
Marital Status	Unmaried
Phone	1732141450
E-mail	Evil.deception@gmailcom
Mailing Address	Dhanmondi-Dhaka-1210
Permanent Address	H#222,Road#12,damurdaa,Barishal-5800.

Parental Details

Father's Name	Sharfuzzaman rubel

Education Information:

The Education History page displays the educational information of the user who is currently logged in. For a student it will display his/her educational information. If the user is a faculty member they will see their respective educational information. The page at a time will display only one educational record. If the user has two records he/she will have the option to move between the records using the control at the bottom



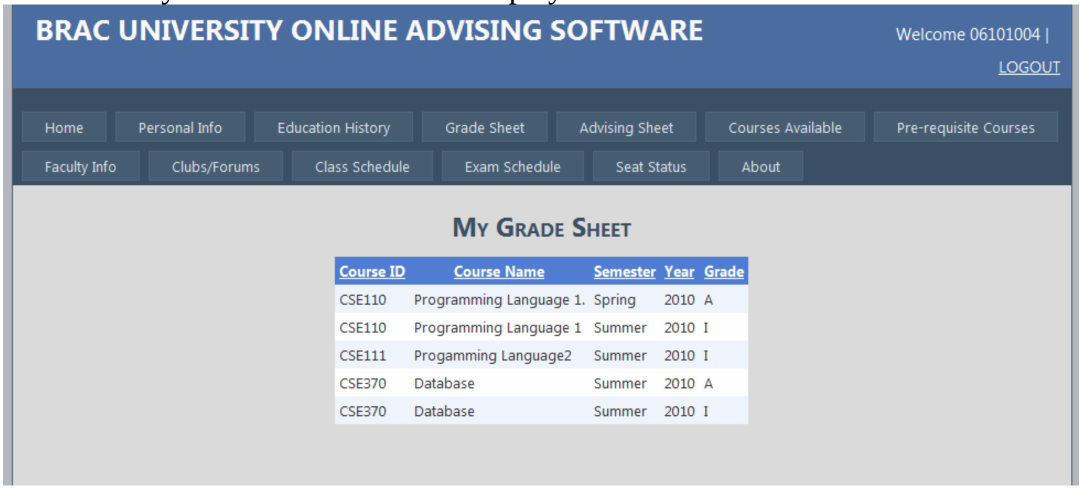
MY EDUCATION HISTORY

Applicant ID	23456789
Student ID	06101004
Degree Type	SSC/'O'-Level/SecondaySchool
Institution	Kholahati cantonment Public School and college,Syadpur
Board	Rajshahi
Passing Year	2002
GPA	4.0
Major Group	Science
Checked By	Sarwar Alam

1 2

Grade Sheet

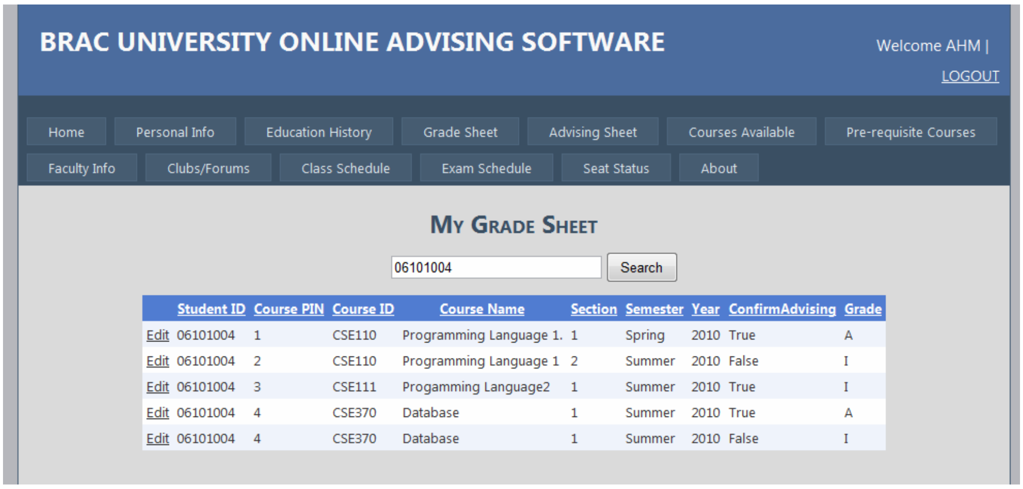
The Grade sheet displays the grade of a student. If a student is logged in than the page automatically loads his/her data and displays it.



MY GRADE SHEET

Course ID	Course Name	Semester	Year	Grade
CSE110	Programming Language 1.	Spring	2010	A
CSE110	Programming Language 1	Summer	2010	I
CSE111	Progamming Language2	Summer	2010	I
CSE370	Database	Summer	2010	A
CSE370	Database	Summer	2010	I

When a teacher is logged in the no records are pulled by default instead they are given the option to search for a specific student using their ID so that the teacher can see their grade sheet.



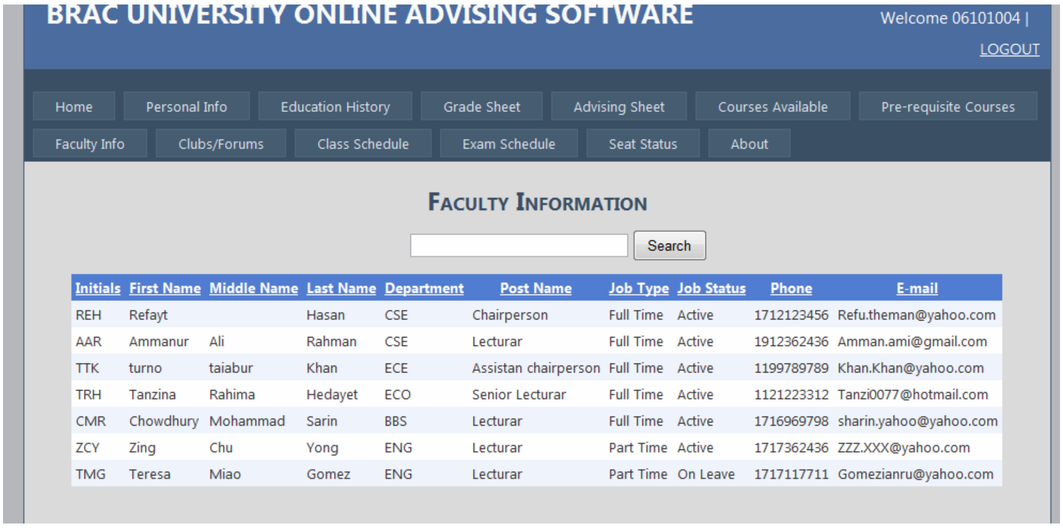
MY GRADE SHEET

06101004

	Student ID	Course PIN	Course ID	Course Name	Section	Semester	Year	ConfirmAdvising	Grade
Edit	06101004	1	CSE110	Programming Language 1.	1	Spring	2010	True	A
Edit	06101004	2	CSE110	Programming Language 1	2	Summer	2010	False	I
Edit	06101004	3	CSE111	Progamming Language2	1	Summer	2010	True	I
Edit	06101004	4	CSE370	Database	1	Summer	2010	True	A
Edit	06101004	4	CSE370	Database	1	Summer	2010	False	I

Faculty Info:

The faculty info button displays the entire faculty enlisted in BRAC University along with their initials, Departments and contact information; like-Cell number and e-mail address ,so that student can easily enquire which teacher is taking their registered course and contact them easily in time of need.

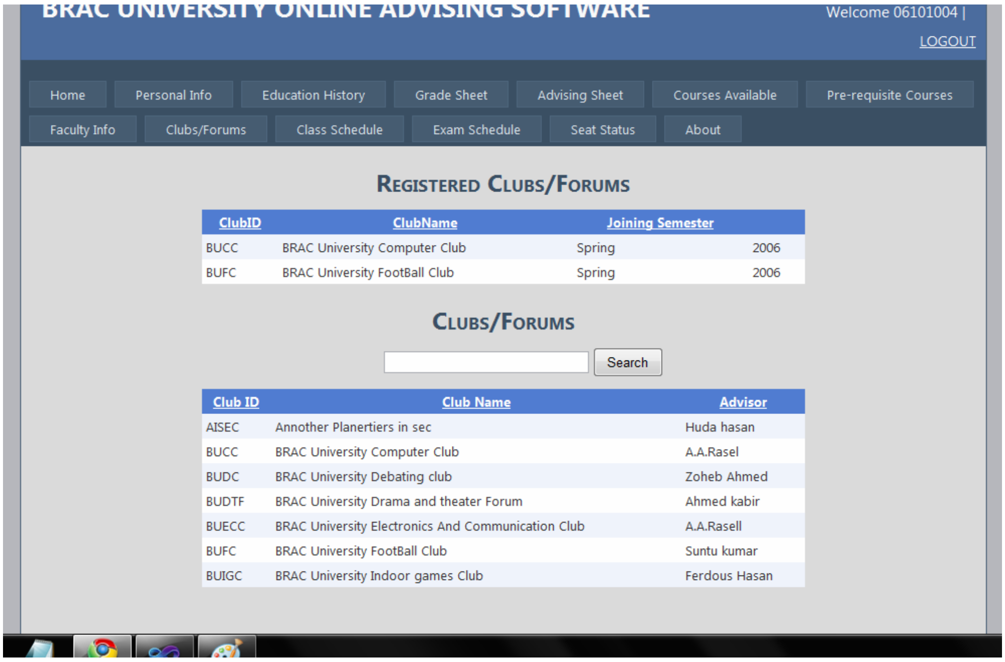


FACULTY INFORMATION

Initials	First Name	Middle Name	Last Name	Department	Post Name	Job Type	Job Status	Phone	E-mail
REH	Refayt		Hasan	CSE	Chairperson	Full Time	Active	1712123456	Refu.theman@yahoo.com
AAR	Ammanur	Ali	Rahman	CSE	Lecturar	Full Time	Active	1912362436	Amman.ami@gmail.com
TTK	turno	taiabur	Khan	ECE	Assistan chairperson	Full Time	Active	1199789789	Khan.Khan@yahoo.com
TRH	Tanzina	Rahima	Hedayet	ECO	Senior Lecturar	Full Time	Active	1121223312	Tanzi0077@hotmail.com
CMR	Chowdhury	Mohammad	Sarin	BBS	Lecturar	Full Time	Active	1716969798	sharin.yahoo@yahoo.com
ZCY	Zing	Chu	Yong	ENG	Lecturar	Part Time	Active	1717362436	ZZZ.XXX@yahoo.com
TMG	Teresa	Miao	Gomez	ENG	Lecturar	Part Time	On Leave	1717117711	Gomezianru@yahoo.com

Clubs/forums:

As displayed in the picture, the clubs/forum tab shows all the clubs enlisted in BRAC University along with a search option. And if the logged in user is a student then he/she can also see the clubs he/she is enlisted in.



REGISTERED CLUBS/FORUMS

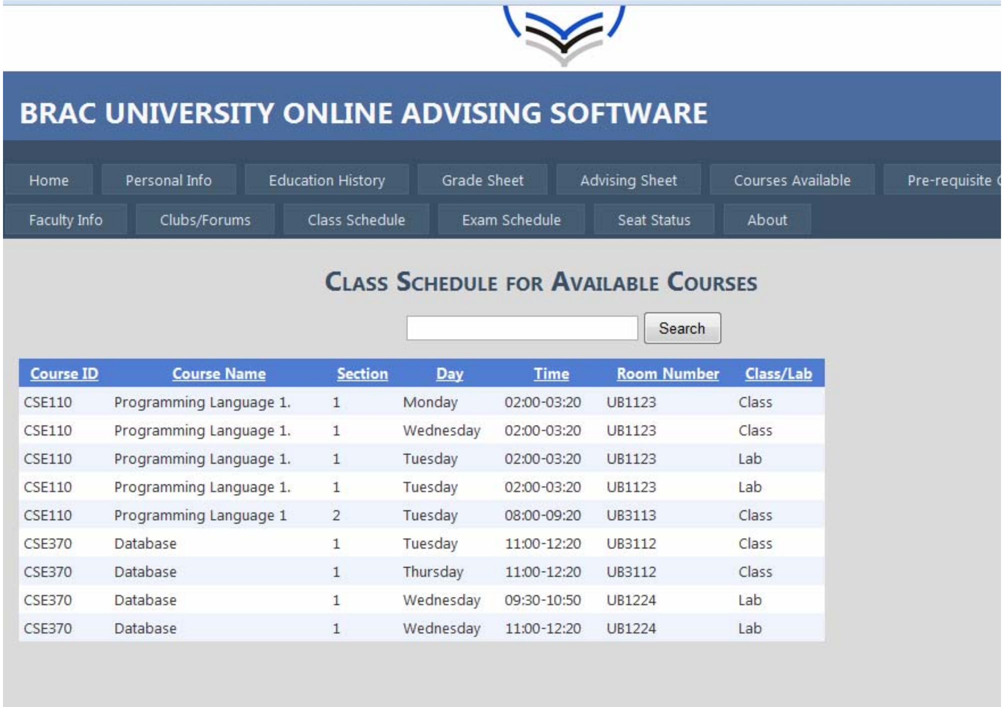
ClubID	ClubName	Joining Semester	
BUCC	BRAC University Computer Club	Spring	2006
BUFC	BRAC University FootBall Club	Spring	2006

CLUBS/FORUMS

Club ID	Club Name	Advisor
AISEC	Annother Planertiers in sec	Huda hasan
BUCC	BRAC University Computer Club	A.A.Rasel
BUDC	BRAC University Debating club	Zoheb Ahmed
BUDTF	BRAC University Drama and theater Forum	Ahmed kabir
BUECC	BRAC University Electronics And Communication Club	A.A.Rasell
BUFC	BRAC University FootBall Club	Suntu kumar
BUIGC	BRAC University Indoor games Club	Ferdous Hasan

Class Schedule:

Class schedule displays all the courses that are being offered in the current semester and there class timing with room number and class type.

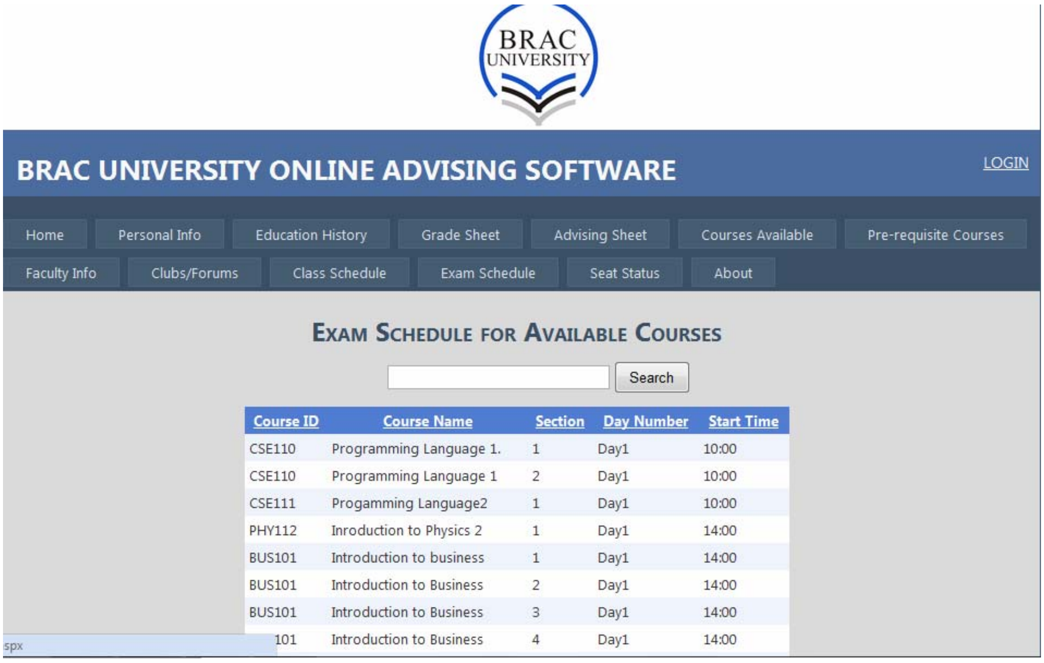


CLASS SCHEDULE FOR AVAILABLE COURSES

Course ID	Course Name	Section	Day	Time	Room Number	Class/Lab
CSE110	Programming Language 1.	1	Monday	02:00-03:20	UB1123	Class
CSE110	Programming Language 1.	1	Wednesday	02:00-03:20	UB1123	Class
CSE110	Programming Language 1.	1	Tuesday	02:00-03:20	UB1123	Lab
CSE110	Programming Language 1.	1	Tuesday	02:00-03:20	UB1123	Lab
CSE110	Programming Language 1	2	Tuesday	08:00-09:20	UB3113	Class
CSE370	Database	1	Tuesday	11:00-12:20	UB3112	Class
CSE370	Database	1	Thursday	11:00-12:20	UB3112	Class
CSE370	Database	1	Wednesday	09:30-10:50	UB1224	Lab
CSE370	Database	1	Wednesday	11:00-12:20	UB1224	Lab

Exam Schedule:

Exam schedule displays the final examinations date and time for all the course those are being offered in the current semester.



Exam Schedule for Available Courses

Course ID	Course Name	Section	Day Number	Start Time
CSE110	Programming Language 1.	1	Day1	10:00
CSE110	Programming Language 1	2	Day1	10:00
CSE111	Progamming Language2	1	Day1	10:00
PHY112	Inroduction to Physics 2	1	Day1	14:00
BUS101	Introduction to business	1	Day1	14:00
BUS101	Introduction to Business	2	Day1	14:00
BUS101	Introduction to Business	3	Day1	14:00
101	Introduction to Business	4	Day1	14:00