CS4400 Database Project
Fall Semester 2012

Purpose of the Project

Analyze, specify, design, implement, document and demonstrate an information system application to support a new online university student/ faculty services system called GTPort. GTport is similar to, but simpler than the existing online system for Georgia Tech (Buzzport). You are required to use the Classical Methodology for Database Development. The system should be implemented using a relational DBMS that supports standard SQL queries. Class administrators will provide you with information about how to access a college-managed MySQL server in order to implement your database and the application. The professors must approve other alternative implementations. *In no circumstances can you use a tool that automatically generates SQL or automatically maps programming objects into the database.*

Project Phases

The three phases of the project cover the following work-processes from the Classical Methodology for Database Development (see notes on T-square under resources).

Phase		Due Date
I	Analysis & Specification	October 3
II	Design	October 31
III	Implementation & Testing	December 4
	Demonstrations	Dead Week

Groups

Project groups may have 3-4 members. Please avoid having smaller groups. 5 or more members will not be acceptable.

A group may remove a member from further participation in the group when Phase I is turned in or when Phase II is turned in. A written notification must be provided to the professor at that time.

Deliverables

Slides on database design methodology will be useful for phases I and II: These slides have been posted on t-square.

Phase I (hard copy)

The deliverables include:

1. A cover page listing all members in the team with their respective sections and email addresses and Tsquare username.
2. Enhanced Entity Relationship (EER) Diagram
3. Information Flow Diagram
4. A list of logical constraints that will be enforced. Do not include any constraints that can be shown in the ER diagram, but rather semantic, business logic related constraints. **You are required to include at least five constraints, although a fully-specified system will have more than that. Constraints that can be specified directly using ER notation will not count toward the five required.**
5. Any assumptions made including explanations.

Notes:

1. The EER must capture the constraints of the system as much as possible whenever applicable, i.e. total participation, super/sub class, weak entities.
2. The design of your system must satisfy all the constraints. You are allowed to make up additional assumptions and constraints as long as they do not conflict with the specified constraints and requirements. If possible, those additional assumptions and constraints should be included in the ER diagram.

   You must turn in a hard copy of your report in class

Phase II (hard copy)

1. Cover Page
2. Copy of the ER Diagram (either from phase I (with any revisions) or from the solution provided)
3. Copy of the Information Flow Diagram from phase I (either from phase I (with any revisions) or from the solution provided)
4. Task Decomposition Diagram
5. Relational Schema Diagram (with primary and foreign keys identified, referential integrity is shown by arrows)
6. Create Table statements, including domain constraints, integrity constraints, primary keys, and foreign keys
7. SQL statements for each task (***follow the template in the phase II design methodology***)

Notes: A set of SQL statements may be required in order to complete one task. However, in such cases, the last SQL statement should show the output according to the specification. Views and nested queries may be used to support the tasks. A nested query can be broken down into views to make the query more readable.

Phase III

Prior to the demo, the TA will give a sample data set. The database has to be populated with this data set prior to the demo. **5% will be deducted from the grade otherwise.**

Implement a working application with all functionality described in this document. Your source code should be mailed to the respective TA who grades your project by the deadline.

Students in 8803ISD: Those students taking 8803ISD have two additional requirements:

1. Demonstrate at least one example of an SQL Injection attack in your application. That is, intentionally leave a vulnerability in your application and then show how it can be exploited to reveal additional information about the backend database or to make illegal changes to the database. This can be demonstrated manually using your web browser or using an SQL Injection tool (many can be found on the Web).
2. Include a 2-3 page write-up describing how the mechanics of an SQL Injection attack. Then, justify why the rest of your application is NOT vulnerable to an SQL Injection attack. In other words, you should describe the kinds of precautions that need to be taken, including coding practices and configuration of the web server and scripting engine.

Deliverables for Phase 3 are:

1. Copy of the Create Table statements from phase II (with any revisions)
2. Source code (documented) for your system (soft copy)
3. A set of working SQL statements for all project tasks
4. A functional application with embedded SQL statements that accesses your database
5. 8803ISD students must demonstrate an SQL Injection attack and include the write-up mentioned above.

Grading

The project will consist of three phases (deliverables) as well as a final demonstration to the TA. Phase I and Phase II of the project are each worth 10% credit.

Phase III (20% credit): We will use the embedded SQL feature of MySQL which allows you to embed SQL statements in a Java program or web application.

GTPort

GTPort is a simple online Georgia Tech student services website. There are primarily two types of users: administrative users and regular users. The regular users can be students or instructors. The students can register for courses, view academic records, view and update personal information. The instructors can assign grades to students for a particular course. The administrator can view certain administrative reports explained in later sections.

The following sections contain a functional description of the GTPort application along with some screen mockups. The user interfaces depicted in this project description merely serve as examples to guide your thinking. Your project's interface may look completely different and that is fine—even encouraged! For example, you might choose to split up some interfaces we have shown on a single screen into multiple screens. You might choose to use popup windows instead of refreshing the page. A complete reorganization of the user interface is acceptable as long as your application supports the same functionality as described below. You may implement the project as a traditional standalone application (e.g., using Java GUIs) or as a web application (e.g., using a web scripting language like PHP). There is no restriction on the choice of language.

ALSO PLEASE NOTE THAT CERTAIN SCREENS HAVE BEEN SHOWN ONLY FOR THE PURPOSE OF THE EER DIAGRAM AND YOU NEED NOT IMPLEMENT THOSE SCREENS FOR PHASE III. WE WOULD GIVE A LIST OF THE SCREENS THAT MUST BE INCLUDED BEFORE PHASE III

ALSO NOTE THAT REQUIREMENTS FOR THE PROJECT MAY CHANGE IN DUE COURSE OF TIME. PLEASE CONSTANTLY LOOK OUT FOR POSTINGS OF REVISIONS TO THE PROJECT.

Logging in to GTPort

Fig. 1 shows the GTPort login screen. All users are uniquely identified by his or her **Username.** A valid **Username** and **Password** combination is required to log in to the system. If the user provides an invalid login credentials, an error message should be displayed and the user should be redirected / returned to the login screen.

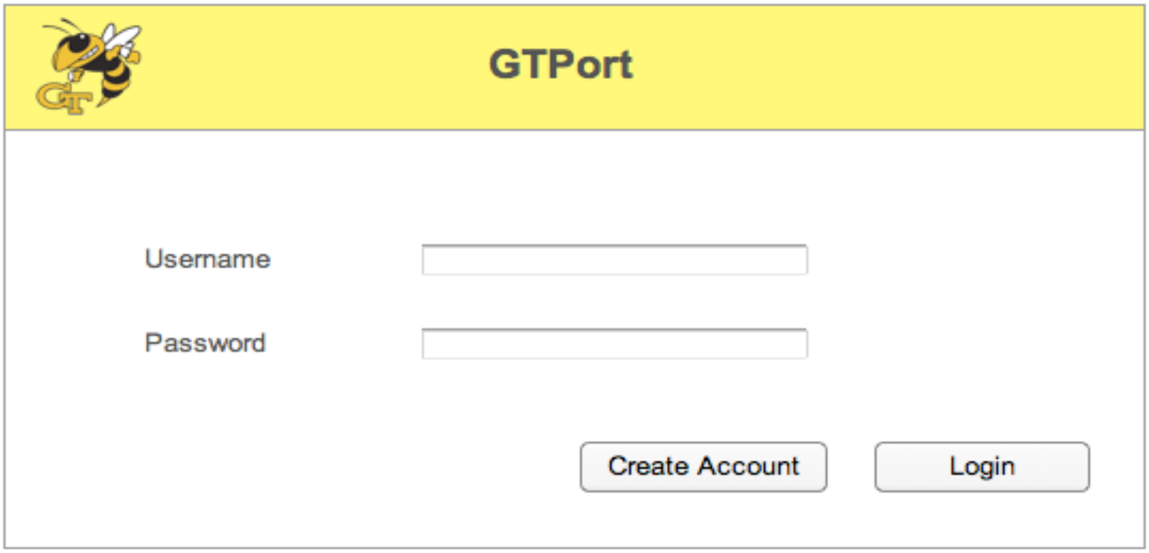


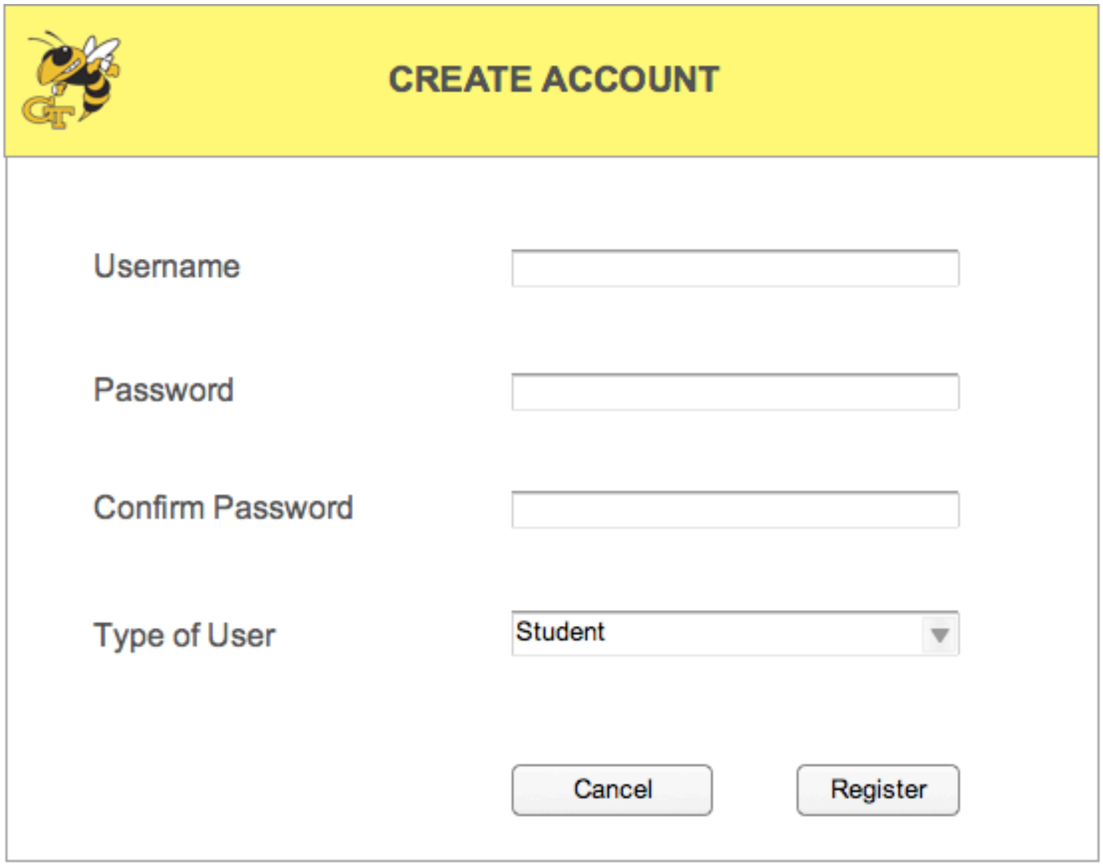
Fig.1 Login Screen

New User Registration

A new user needs to register before using the GTPort. This applies only for students and instructors and not for the administrators. We assume that the administrator already has his credentials. A *Create Account* button is provided on the login screen. Clicking this button displays the new user registration form as shown in Fig.2. Retrieving a forgotten password is an optional functionality but is not required to be implemented.

After the user clicks **Register**, the system should verify that all fields are filled in, that the **Username** has not already been registered, and that the **Password** and **Confirm Password** fields are equal. If any of these validations fail, the user should be returned to this screen to make corrections. The user should be provided with meaningful error messages so he or she knows what to correct.

Users who register through this interface do NOT have administrative privileges. The only way an administrator can be created is by the DBA behind the scene.

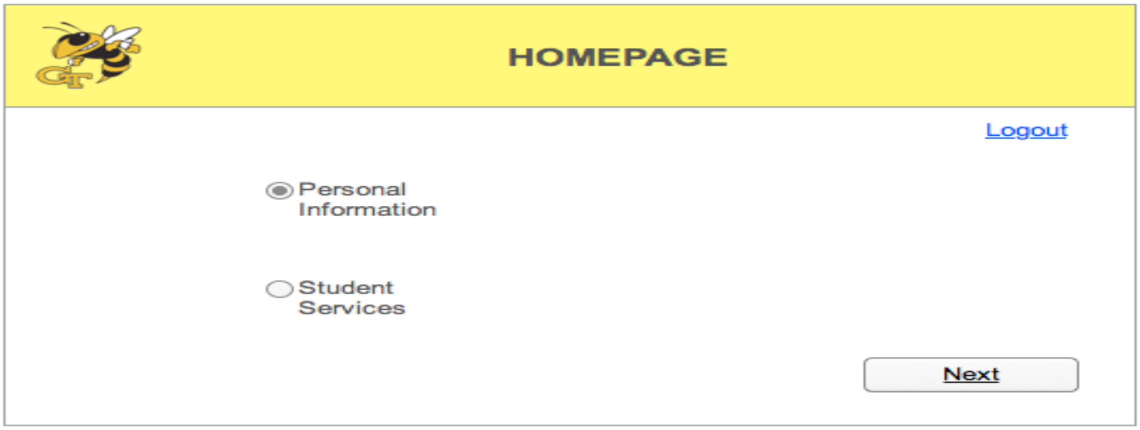


Type of User: Student

Fig 2. Create Account

Once a regular user logs into the system he/she should be presented with the following options. [Fig.3]

- Personal Information
- Student Services / Faculty Services (depends on who logged in)



● Personal Information

○ Student Services

Fig.3a Homepage for Student

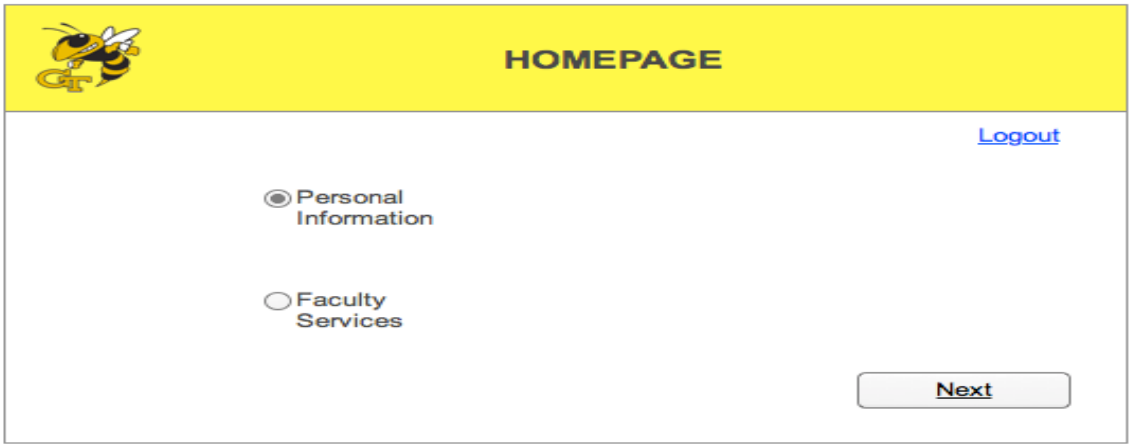


Fig 3b. Homepage for Faculty

It is mandatory for the regular users to enter their personal information before using any of the services. If the user tries to use the services provided before entering his personal information he should be shown an error message.

The Homepage presents the student with options of entering personal information or use the student services provided by the system.

When the administrator logs in he would be shown the screen below where he has an option of either adding a course (explained later) or viewing an administrative report.

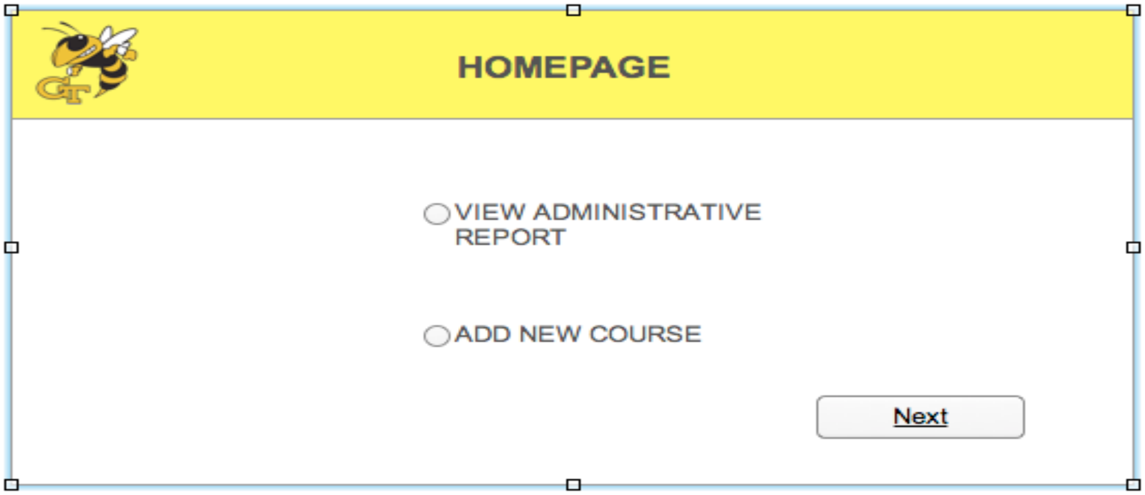


Fig 3c. Homepage for Administrator

Personal Information

On selecting the Personal Information option the student would be redirected to the Personal Information page[Fig 4a]. A student needs to enter his name , Date of Birth , gender , address , permanent address, contact number, email address, major, degree ,etc. All the above fields are mandatory. On doing so a student ID is assigned to each student which is unique.

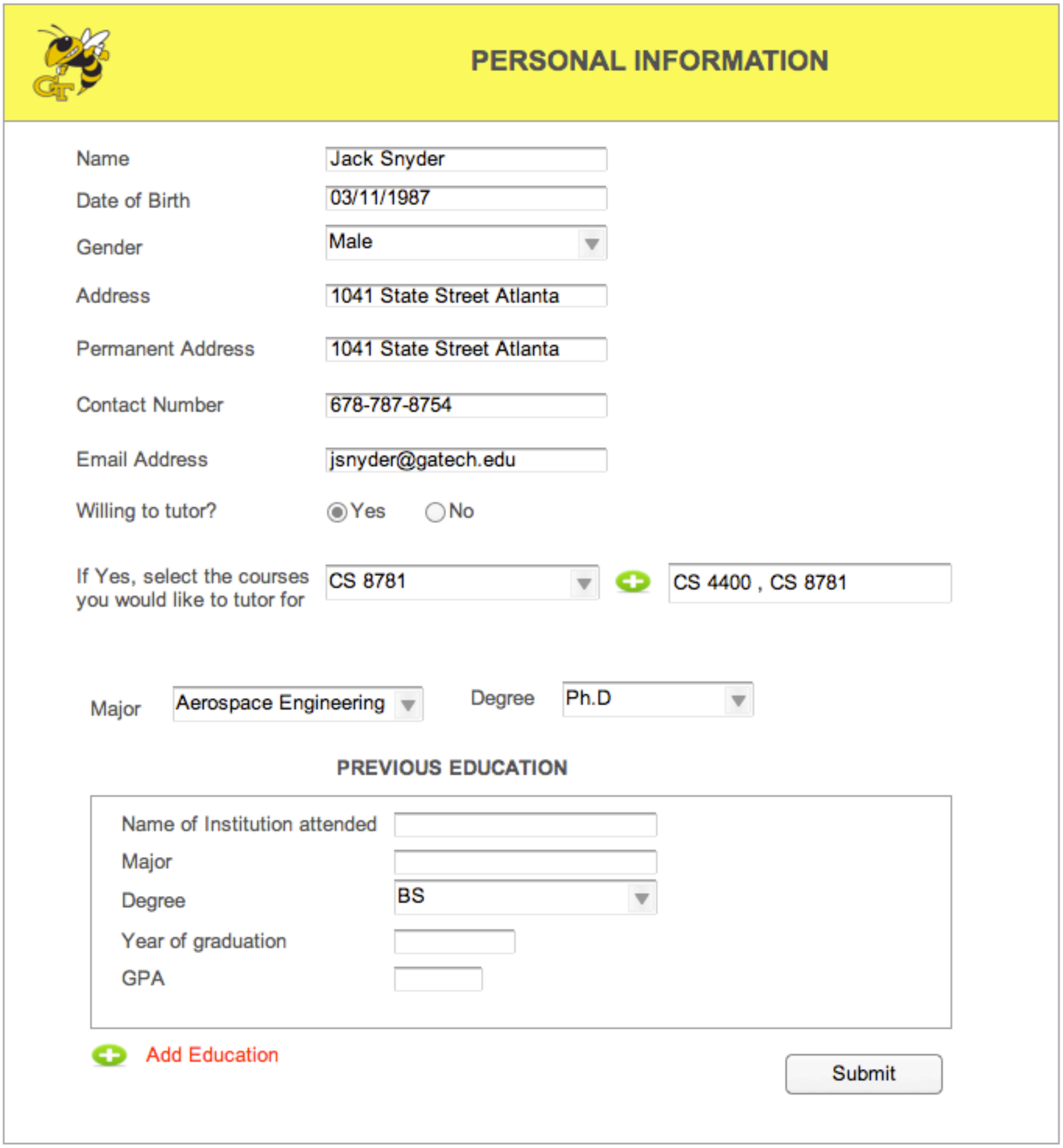
The majors offered by the institution are:

- **Aerospace Engineering**
- **Biology**
- **Biomedical Engineering**
- **Computer Science**
- **Electrical & Computer Engineering**.

There are three kind of degrees offered by the university namely

- **BS**
- **MS**
- **Ph.D**.

The student is also required to enter his previous education history. This would include the schools attended by the student in the past, year of graduation, majors studied, gpa achieved. The students should be allowed to enter a maximum of 3 previous institutions that he / she has attended. [See Fig.4a]



PERSONAL INFORMATION

Name	Jack Snyder
Date of Birth	03/11/1987
Gender	Male
Address	1041 State Street Atlanta
Permanent Address	1041 State Street Atlanta
Contact Number	678-787-8754
Email Address	jsnyder@gatech.edu
Willing to tutor?	● Yes ○ No
If Yes, select the courses you would like to tutor for	CS 8781 — CS 4400 , CS 8781

Major: Aerospace Engineering Degree: Ph.D

PREVIOUS EDUCATION

Name of Institution attended	
Major	
Degree	BS
Year of graduation	
GPA	

Fig 4a. Personal Information page for students

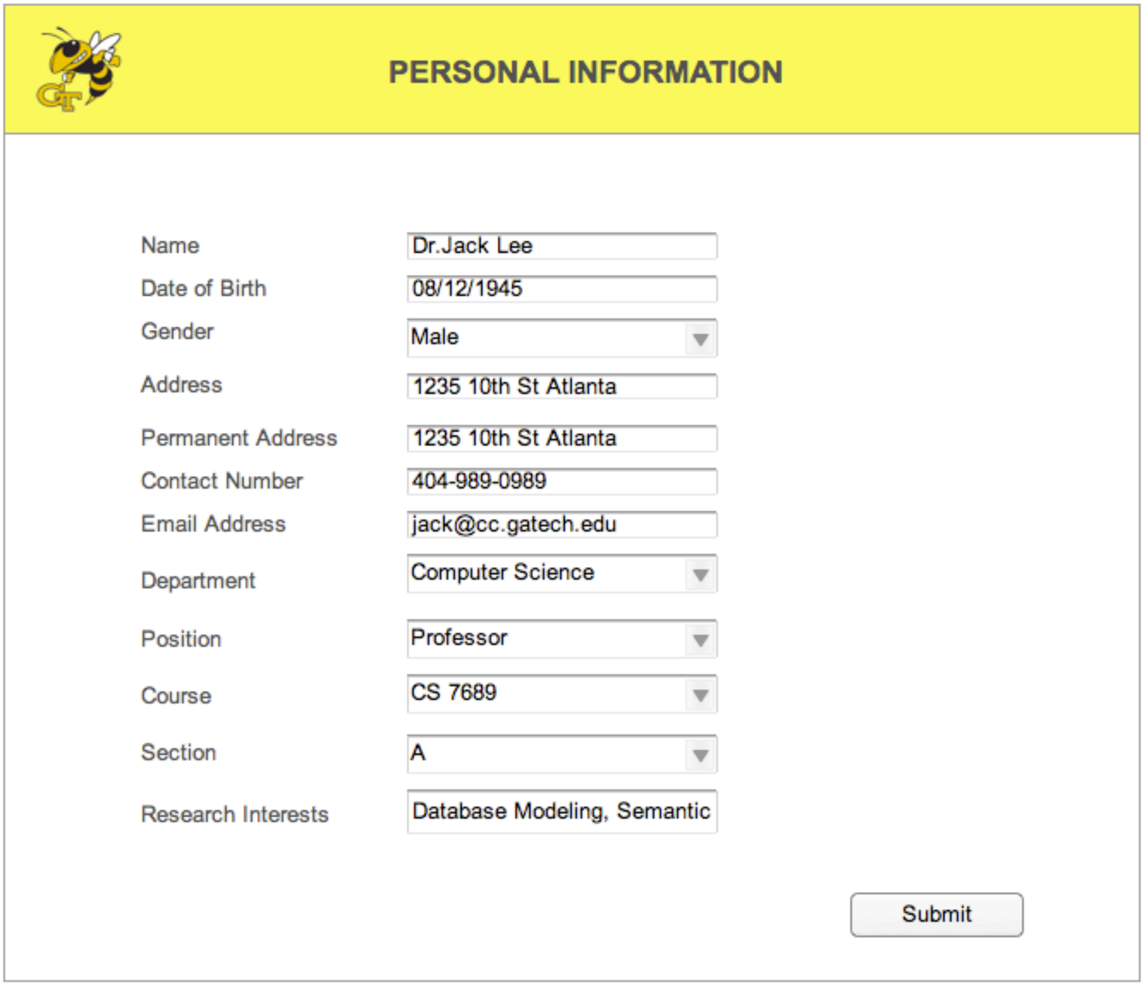
The students can also opt to tutor for courses. If they select 'Yes' then in the dropdown list only those courses should appear in which the student received atleast a grade of B in prior semesters. We do not allow students to tutor for courses they themselves did not fare well in. Providing this piece of information doesn't assign them a tutor position. They would need approval from the appropriate instructor (explained later). Once all the information is entered an ID would be generated for each student.

In case of instructors they would have to provide the system with their names, their position in the department, the department they belong to (assuming that each instructor belongs to one department only), the course they teach other than some of the general information asked from the students. [Fig 4b]. Just like students an Instructor ID is also assigned to an instructor once he provides his details on the personal information page.

Following are the possible positions of the instructors

- Professor
- Associate Professor
- Assistant Professor

The drop down list of courses in the personal information page of the faculty should list all the courses of the department he selects in the previous step. For example if the instructor belongs to the Computer Science department then the course list that he chooses from consists of all the courses offered by that department. **(Note that this is the course he offers to teach and he must teach a course and cannot teach more than one course. BUT he can teach multiple sections of the same course)**. Every instructor can also have multiple research interests.



PERSONAL INFORMATION

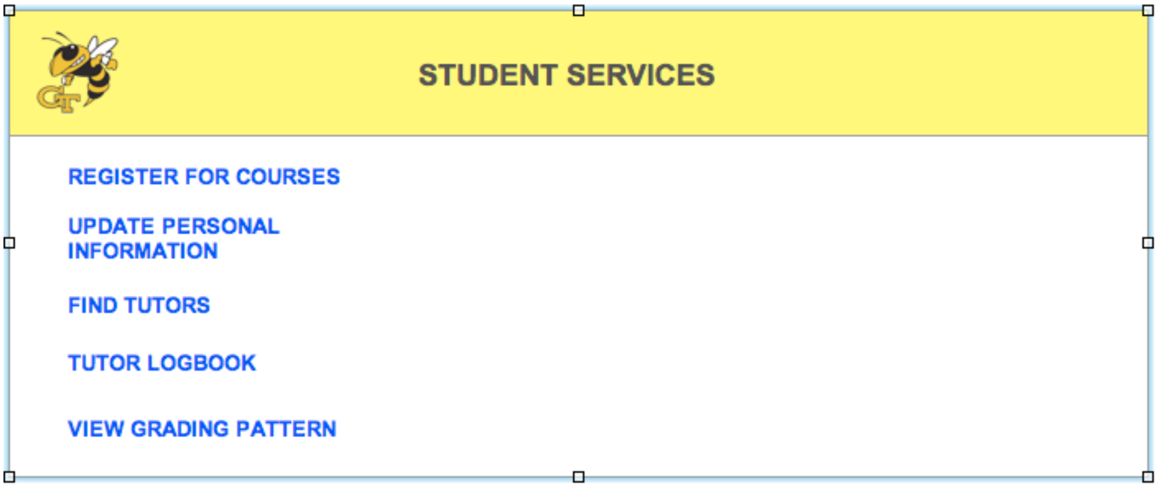
Name	Dr.Jack Lee
Date of Birth	08/12/1945
Gender	Male
Address	1235 10th St Atlanta
Permanent Address	1235 10th St Atlanta
Contact Number	404-989-0989
Email Address	jack@cc.gatech.edu
Department	Computer Science
Position	Professor
Course	CS 7689
Section	A
Research Interests	Database Modeling, Semantic

Fig 4b. Personal Information page for Faculty

Student Services

On selecting the Student Services option the student would be redirected to the Student Services page. Here the student can register for courses for the coming semester , view/update his personal information, search for tutors in a particular area and view the grading pattern report.

The main page would look like the one below.



STUDENT SERVICES

REGISTER FOR COURSES

UPDATE PERSONAL INFORMATION

FIND TUTORS

TUTOR LOGBOOK

VIEW GRADING PATTERN

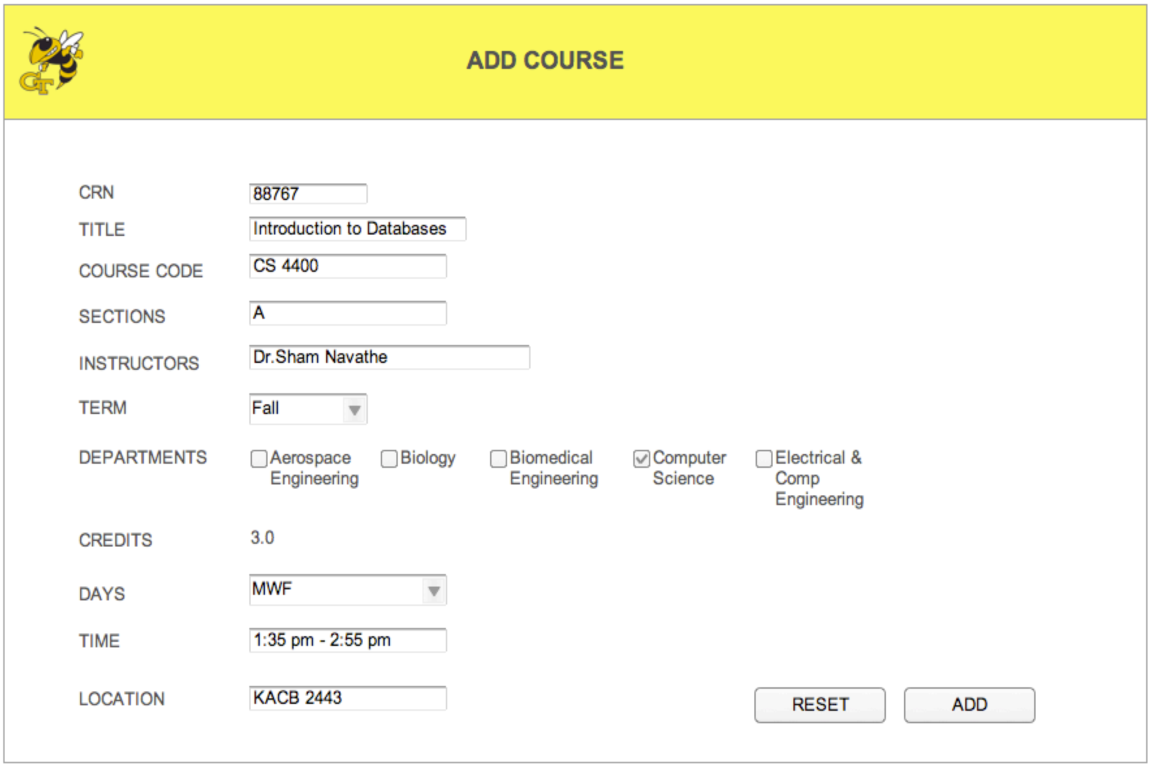
Fig. 5 Student Services Page

Please note that the ‘Tutor Logbook’ option would only be enabled if the student who is logged in is also a tutor. Otherwise this option would be disabled by default.

Course Registration

There are 5 departments in the university, information about which has been mentioned above. A department has a name and a Dept Id. Each department offers multiple courses and a course may be cross listed across multiple departments. Other than having a title which would be unique a course will have at least one course code(since courses can be cross listed across departments). For eg. A course called ‘Advanced Bioinformatics’ can have a course code of BIOL 7453 and CS 8782 . Each course may have multiple sections denoted by letters A, B,C, etc. and at least one instructor and it may be offered in the Fall or Spring only but not both. Each section would have a unique CRN. A student can register for multiple courses but can only register for courses offered by his/ her department unless the course is cross listed. He / She can choose the mode of grading for the course which can be ‘Registered’, ‘Audit’ or ‘Pass/Fail’. Information about the day and time the course would be taught along with location would also be stored. Assume all courses are 3.0 credit courses. Only the administrator is authorized to enter information about the courses. Note that if there are multiple sections of a course he can enter information about each section one at a time.[Fig 6]

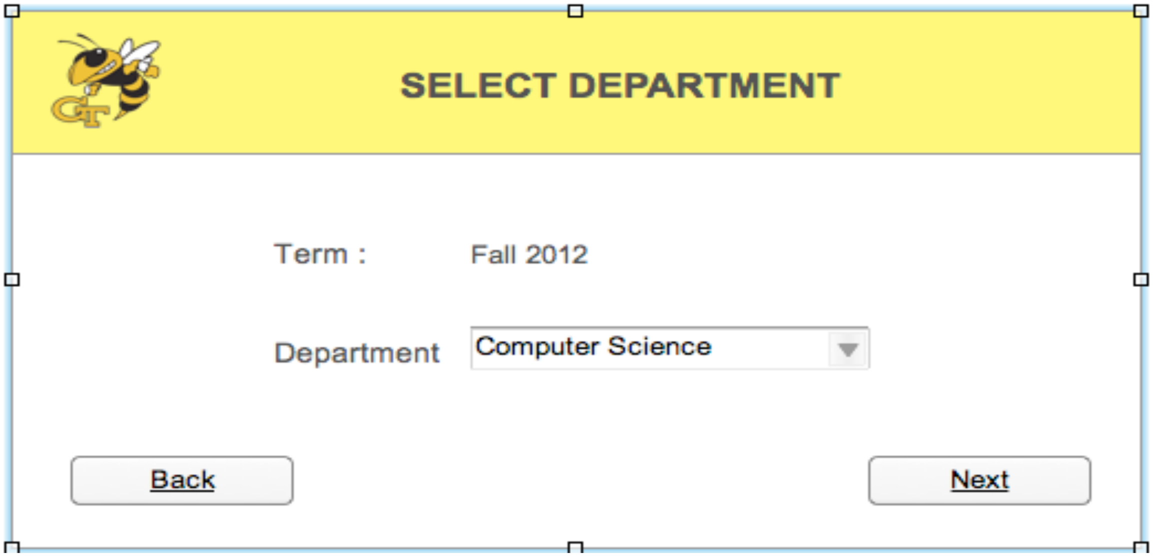
Once a student clicks on the REGISTER FOR COURSES link he would be redirected to the page where he can select his department to view the courses offered in the current term [Fig 7]. On doing so the page should show the list of courses being offered by the department [Fig 8.] . The student needs to check the course he wants to register for.



ADD COURSE

CRN	88767
TITLE	Introduction to Databases
COURSE CODE	CS 4400
SECTIONS	A
INSTRUCTORS	Dr.Sham Navathe
TERM	Fall
DEPARTMENTS	☐ Aerospace Engineering ☐ Biology ☐ Biomedical Engineering ☑ Computer Science ☐ Electrical & Comp Engineering
CREDITS	3.0
DAYS	MWF
TIME	1:35 pm - 2:55 pm
LOCATION	KACB 2443

Fig 6. Add Course

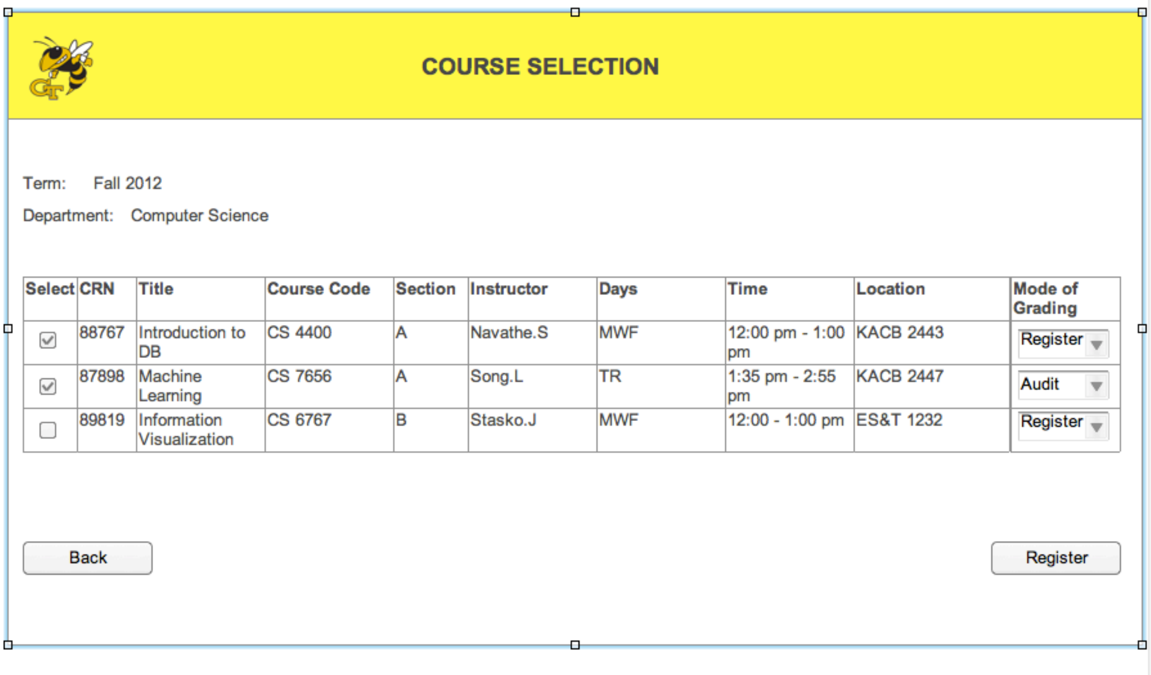


SELECT DEPARTMENT

Term : Fall 2012

Department Computer Science

Fig. 7 Department Selection



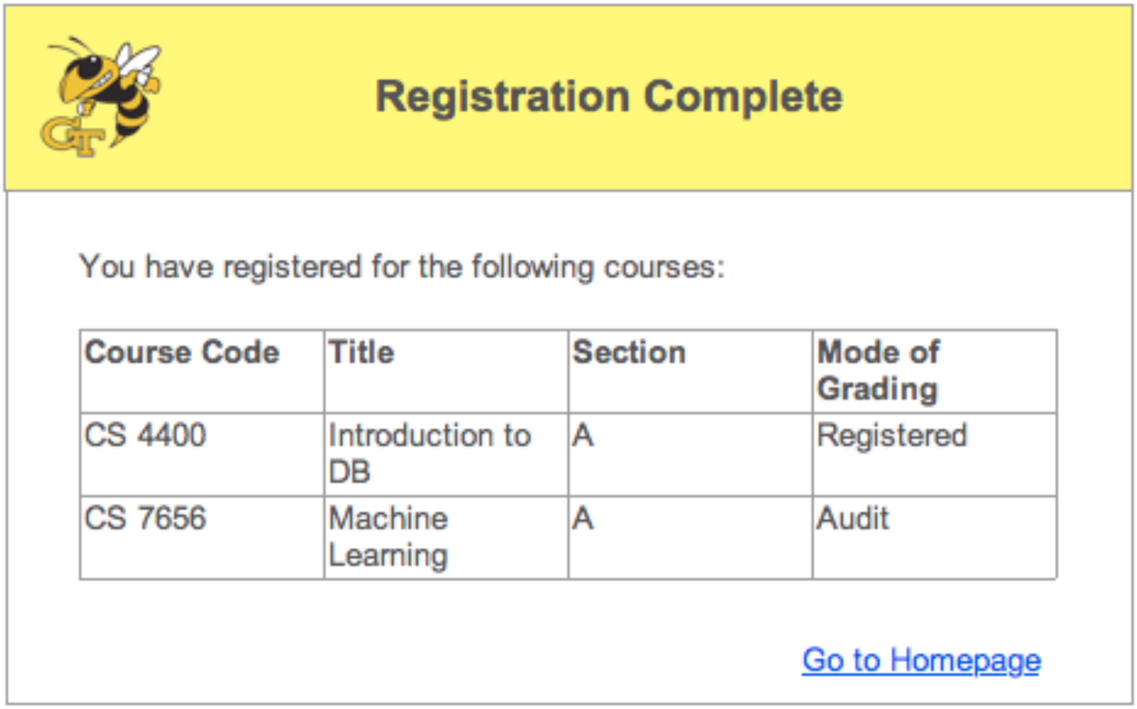
COURSE SELECTION

Term: Fall 2012

Department: Computer Science

Select	CRN	Title	Course Code	Section	Instructor	Days	Time	Location	Mode of Grading
☑	88767	Introduction to DB	CS 4400	A	Navathe.S	MWF	12:00 pm - 1:00 pm	KACB 2443	Register ▾
☑	87898	Machine Learning	CS 7656	A	Song.L	TR	1:35 pm - 2:55 pm	KACB 2447	Audit ▾
□	89819	Information Visualization	CS 6767	B	Stasko.J	MWF	12:00 - 1:00 pm	ES&T 1232	Register ▾

Fig.8 Courses View



Registration Complete

You have registered for the following courses:

Course Code	Title	Section	Mode of Grading
CS 4400	Introduction to DB	A	Registered
CS 7656	Machine Learning	A	Audit

Fig.9 View after Registration

Faculty Services

When an instructor logs into the system he can view/ update his personal information and enter grades for the students that registered for his course.[Fig 10.] The following are the grades which can assigned:

- A – Outstanding
- B – Very Good
- C – Good
- D – Average
- F – Fail

For audit courses students won't be assigned any grade since they only require class participation. For Pass/ Fail courses a grade of 'S' can be assigned for satisfactory performance and 'U' for unsatisfactory.

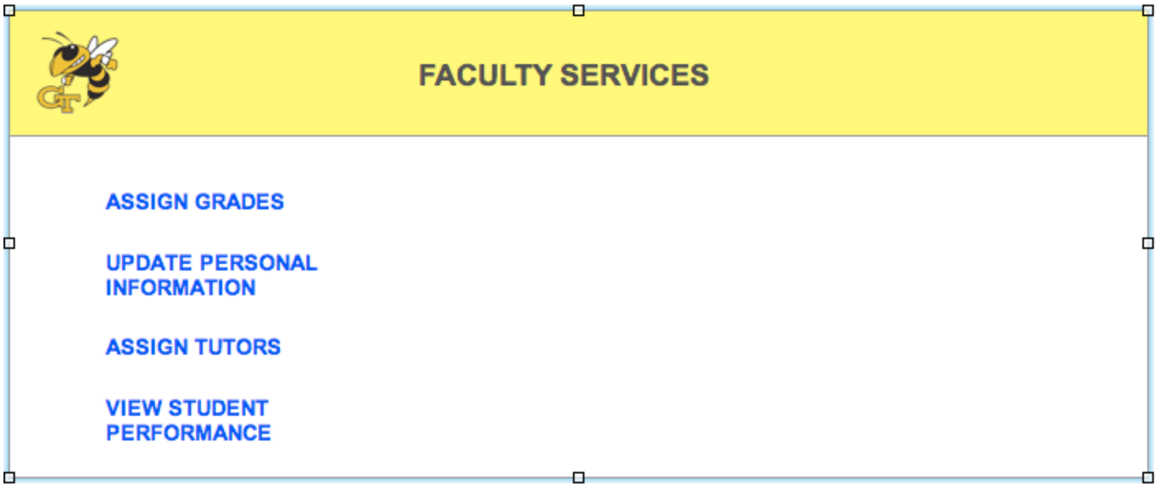
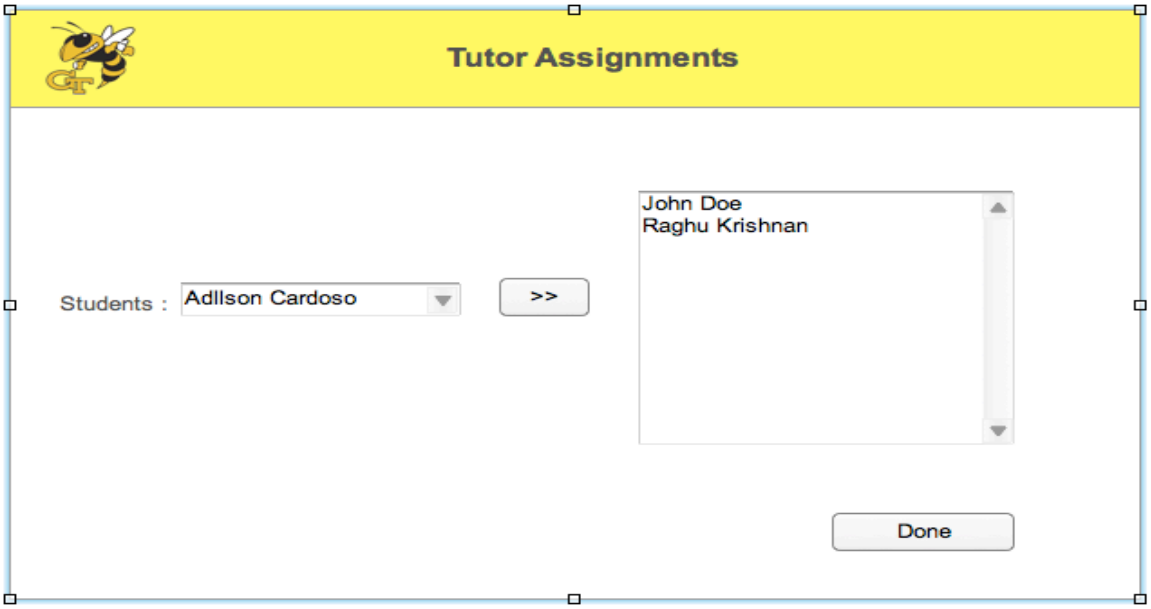


Fig.10 Faculty Services Page

The instructor can click on 'Assign Tutors' to view the list of students who have opted to tutor the course they teach. From the list they can approve the students that they think can be potential tutors. See Fig 11.



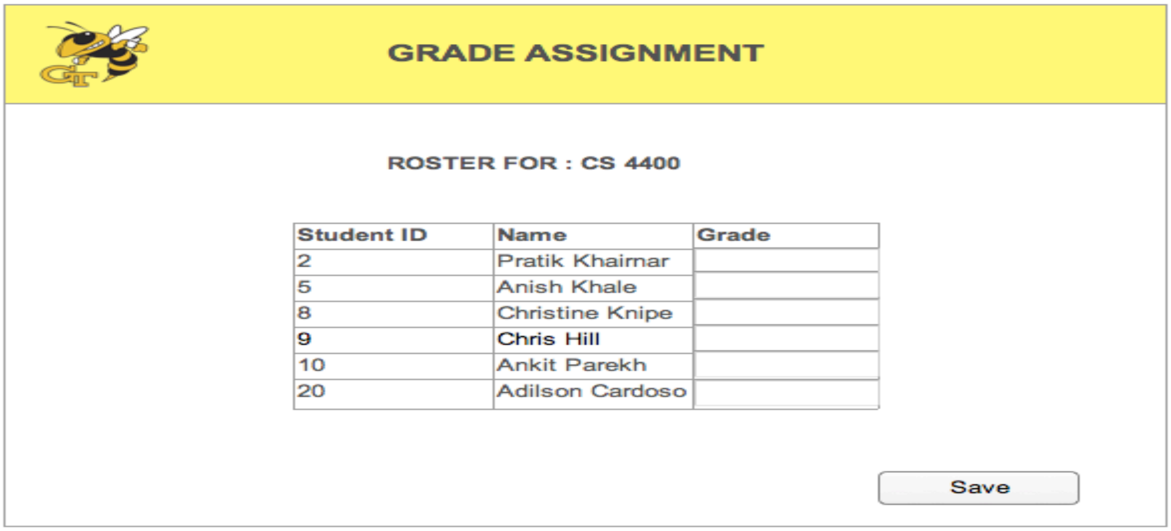
Tutor Assignments

Students : Adllson Cardoso

John Doe
Raghu Krishnan

Fig 11. Tutor Assignments

Since we are assuming that an instructor teaches only one course in a particular term on clicking ‘Assign Grades’ a roster of students is displayed who can then be graded [Fig.12]. On saving the grades the user would be prompted that he has successfully entered grades for his course and would be redirected to the Faculty services page.



GRADE ASSIGNMENT

ROSTER FOR : CS 4400

Student ID	Name	Grade
2	Pratik Khairnar	
5	Anish Khale	
8	Christine Knipe	
9	Chris Hill	
10	Ankit Parekh	
20	Adilson Cardoso	

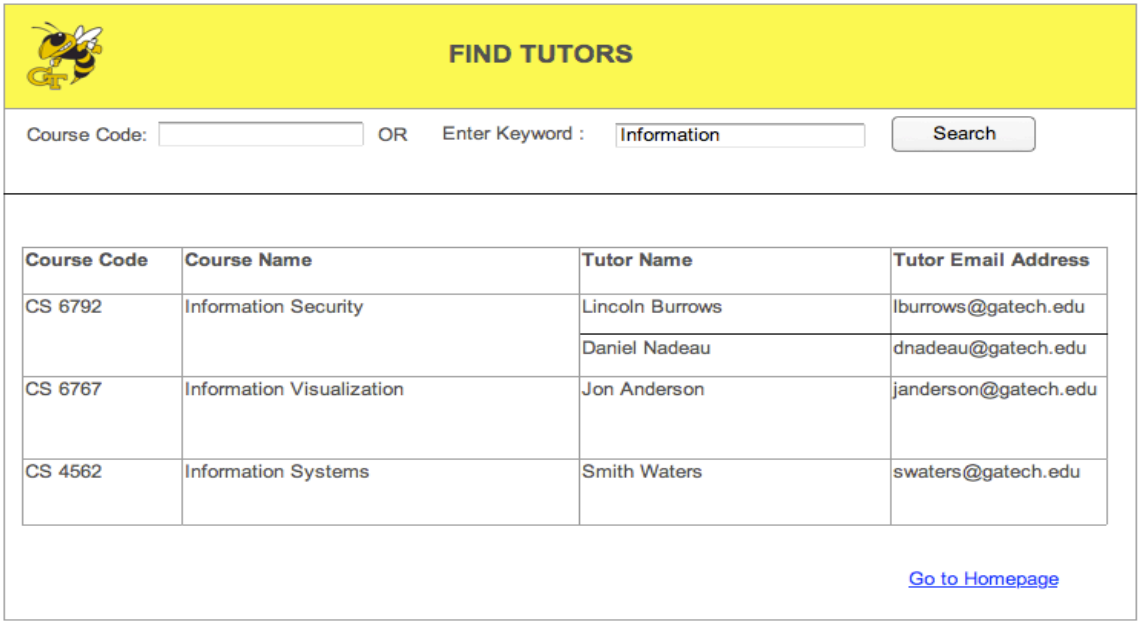
Fig. 12 Grade Assignments

View / Update Personal Information

Both students and instructors can view and update their personal information which they were required to enter when they logged into the system for the first time. To update they would be redirected to the appropriate personal information page and can edit and save the new information.

Search for Tutors

The system also provides a tutor finding service for students who are facing difficulty in a particular course work. This service is a keyword based search service. A student can either enter the course code for the particular course he needs tutors for or he can enter any keyword which should be a part of the course name. This search would return him names and email addresses of all the tutors of that particular course. If the keyword search matches multiple courses, then you are required to group the tutor information by the courses. [Fig. 14].



FIND TUTORS

Course Code: OR Enter Keyword : Information

Course Code	Course Name	Tutor Name	Tutor Email Address
CS 6792	Information Security	Lincoln Burrows	lburrows@gatech.edu
		Daniel Nadeau	dnadeau@gatech.edu
CS 6767	Information Visualization	Jon Anderson	janderson@gatech.edu
CS 4562	Information Systems	Smith Waters	swaters@gatech.edu

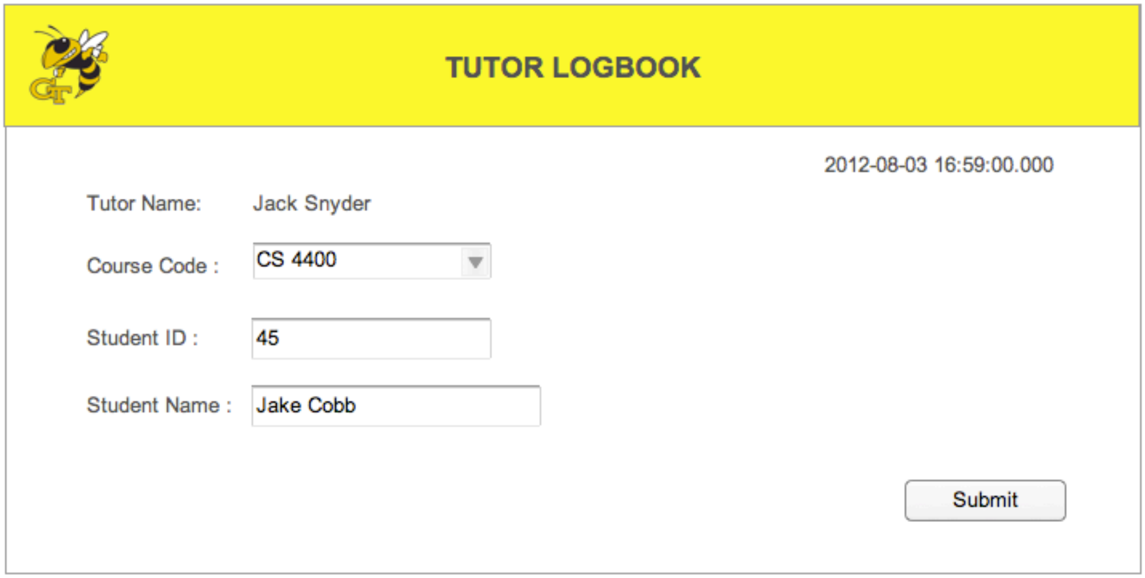
Fig. 14 Tutor Search Results

Section on Logbook maintained between the Tutor and Student.

Student Tutor Logbook

The students and the tutors maintain a logbook where they record their consultation visit whenever they meet. This would help keep a record of how many students required tutoring and whether tutoring helped them getting a good grade. The tutors have access to this logbook. When a tutor clicks on

'TUTOR LOGBOOK' option on the student services page then the screen shown in Fig 15 would pop up.



TUTOR LOGBOOK

2012-08-03 16:59:00.000

Tutor Name:	Jack Snyder
Course Code :	CS 4400
Student ID :	45
Student Name :	Jake Cobb

Fig. 15 Tutor Logbook

Since the tutor is the one who is logged , the system should automatically pull up his name. The dropdown list of courses should have courses that the tutor has been approved to tutor for. The student ID needs to be entered by the tutor and that student should be registered for that course for the current term . If the student ID entered is of a student not registered for the term then an error message should be displayed. Once the ID is entered the 'student name' textbox should auto populate.. The system should automatically pull up the date and time and display at the top of the screen. Note that you need to keep track of every visit a particular student has with a tutor for a particular course.

REPORTS

There are three kinds of reports which are required :-

- Administrative Report
- Faculty Report
- Student Report

When writing SQL statements for these reports, use as few SQL statements as possible. In some cases, you might need to use a separate SQL statement to calculate totals. The important thing to remember is to let the database do the work. Don’t simply pull in all the needed information and do the grouping/aggregation using programming language constructs. They query processor can do it much more efficiently that you can by hand. In other words, you want the output returned from your SQL query to be as close as possible to the information in the final report.

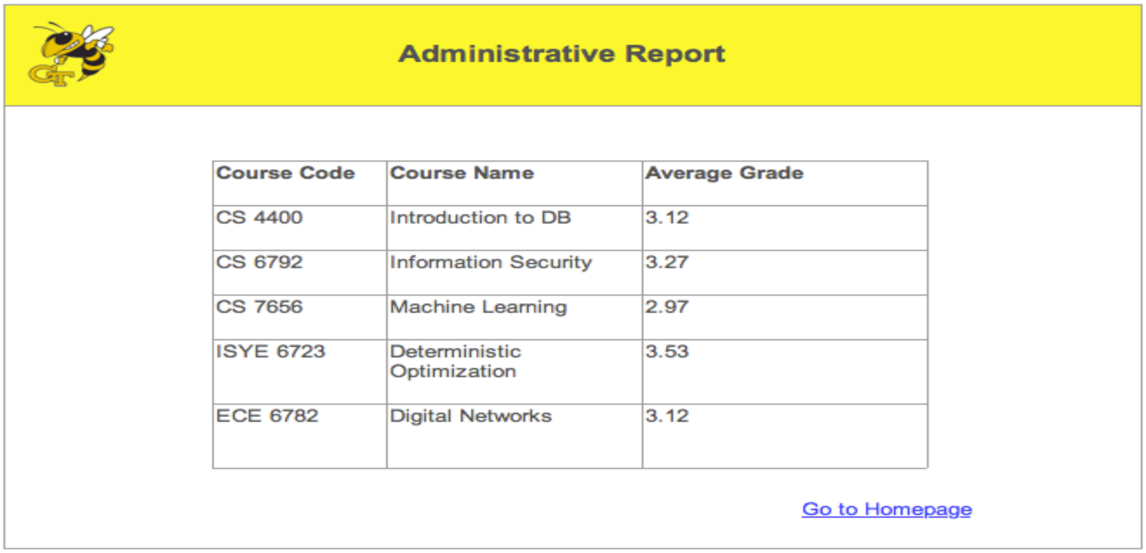
For these reports we would provide you with data that includes individual grades in courses obtained by students in the previous semesters.

Note: Each letter grade has a weight / value assigned to it. (Look the table below)

Grade	**Weight**
A	4
B	3
C	2
D	1
F	0

A) Administrative Report

An administrator would like to see what was the average grade obtained by students in the previous as well as current semester belonging to a particular course.. Figure 16 shows an example of such a report.

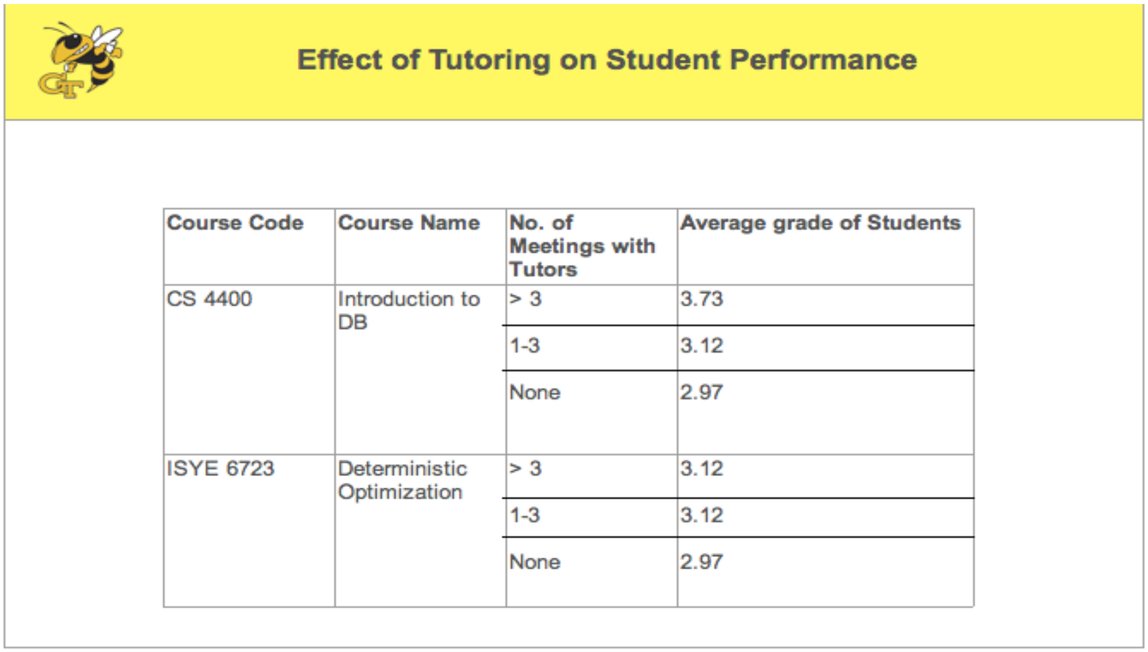


Course Code	Course Name	Average Grade
CS 4400	Introduction to DB	3.12
CS 6792	Information Security	3.27
CS 7656	Machine Learning	2.97
ISYE 6723	Deterministic Optimization	3.53
ECE 6782	Digital Networks	3.12

Fig. 16 Administrative Report

B) Effect of Tutoring Program on Student Performance

The faculty would like to see if there was any affect on the performance of students after they took assistance from tutors. They would like to see the average grade of students for a particular course who met with tutors more than 3 times , 1 – 3 times (1 included, 3 not included) and those who never saw a tutor.



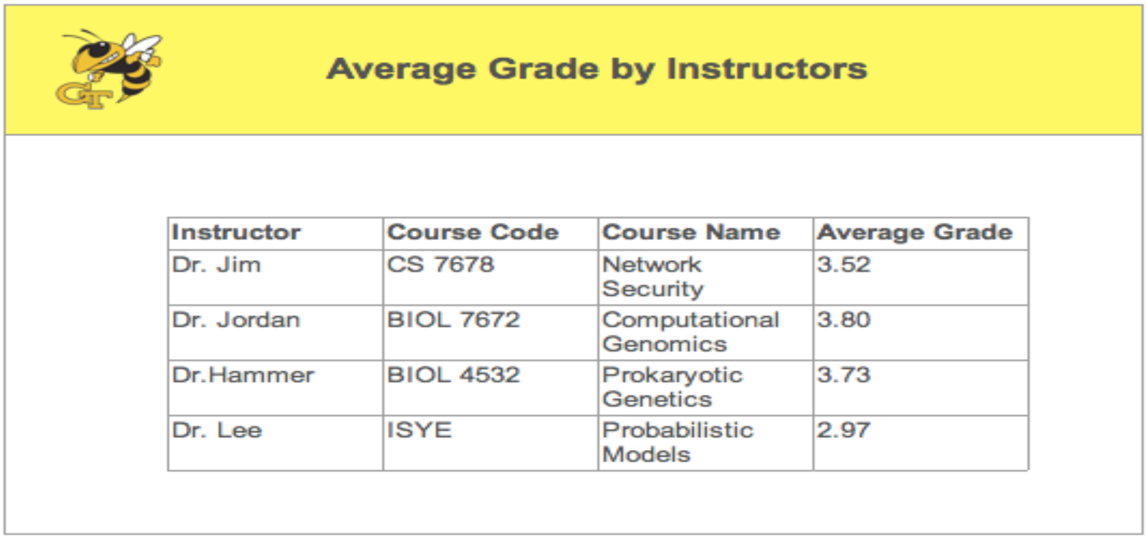
Effect of Tutoring on Student Performance

Course Code	Course Name	No. of Meetings with Tutors	Average grade of Students
CS 4400	Introduction to DB	> 3	3.73
		1-3	3.12
		None	2.97
ISYE 6723	Deterministic Optimization	> 3	3.12
		1-3	3.12
		None	2.97

Fig 17. Faculty Report

C) Grading Pattern of Instructors

This is a report which a student can view to see the grading pattern of an instructor irrespective of the department the instructor belongs to. This report will include the instructor name, course code, course name and average grade assigned.



Average Grade by Instructors

Instructor	Course Code	Course Name	Average Grade
Dr. Jim	CS 7678	Network Security	3.52
Dr. Jordan	BIOL 7672	Computational Genomics	3.80
Dr.Hammer	BIOL 4532	Prokaryotic Genetics	3.73
Dr. Lee	ISYE	Probabilistic Models	2.97

Fig.18 Student Report

END OF DESCRIPTION OF PROJECT

Document Version Info.

Version	Notes	Date
1.0	Original Version	09/05/2012
1.1	1. Task Decomposition Diagram moved from Deliverables of Phase I to Phase II. 1. The create account page has been modified 2.Research Interests field added to the Instructor's Personal information page 3. Add Course screen updated 4. Course Registration Section has been modified 5. A field called 'Degree' has been added to the education history panel	09/19/2012

	of the Personal information page for students. 6. Administrative Report has been modified. 7. Tutor Logbook screen has been modified	