UBC Graduate Information System (GIS)

Project Design

University of British Columbia Okanagan

COSC 304 – Fall 2009

Version: 1.4
Date: 12/02/2009

Project Team

Kyle Kotowick ()

Andrew Campbell ()

Document Control Sheet

Date	Version Number	Author	Revision Description
11/02/2009	1.0	Kotowick	Initial Draft
11/03/2009	1.1	Kotowick, Campbell	Improved formatting, Human Interface section added
11/17/2009	1.2	Kotowick	Changes made to reflect suggestions from first marking (new relations, modified ER and DDL)
11/22/2009	1.3	Kotowick	Latest version of DDL included, new ER diagram. Added "Progress Report" section. Updated all other sections to reflect modifications.
12/02/2009	1.4	Kotowick, Campbell	Updated DDL, added final documentation (new executive summary, walkthrough, and high-level and individual file descriptions), removed "Interim Report / Current Progress" section.

Table of Contents

1. Executive Summary

The UBC Graduate Information System is a database / web browser combination system designed to keep track of any and all students that attend UBC Okanagan (can be expanded to Vancouver as well), including information such as contact information, academic records, and previous/current employment. The requirement for such a system has arisen from the need for a consolidated location where students and faculty alike can keep track of important information that may be useful in the future. The system is designed such that administrators will be responsible for adding new students and academic information (courses taken, degrees accomplished, etc.) and each student will be responsible for maintaining his/her own contact and employment information. Each student also has an individual "comment board" where that student and all faculty users can post comments.

The system operates on the basic principle that there are 3 privilege levels available to users: student, faculty, and administrator. All students are able to view all of their own information and may modify their employment records (both current and previous) and contact information (a set of 5 basic non-removable contact information types with the option to add "extra" types). Faculty may view (but not modify) all information for any student and administrators may add new users/students and modify any information for any student. The system uses a Microsoft SQL Server database on a university server to operate. This can be easily modified to be moved on to a specialized platform.

Enhanced features include the ability for faculty and administrators to search for students by student number, view lists of students in specific courses or degrees, and add new degrees or courses available to be taken. Furthermore, the system is designed to be highly secure and does not allow any users to access any features or functions without first signing in with a legitimate account. Due to the high likelihood of future modifications, the system has been constructed with a highly encapsulated, modular design.

The user interface of the software has been designed for ease of use in finding pertinent information and modifying/removing it.

2. Domain Assumptions

2.1 Users

- Users have a unique ID, along with a password used to log into the system.
- Users have three distinct permission settings: student, faculty, administrator.
- Students will be able to modify all personal data (jobs, contact info).
- Faculty will be able to see all data for all students.
- Administrators will be able to modify all data for all users (course grades, contact info, etc.).

2.2 Students

- Students have a unique student ID.
- Students have gender.
- Students have dates of enrolment and birth.
- Student first and last names are stored.

2.3 Courses

- All courses that students have taken are recorded.
- Course information includes course number, course name, grade, and pass/fail.

2.4 Graduation

- Every completion of a degree is recorded for each student (students may have multiple degrees, such as Masters or Ph.D's).
- Graduation information contains graduation date, GPA, degree, and whether or not it was an honours degree.

2.5 Jobs

- Students can list multiple jobs, current or past
- Jobs contain info on company, position, start and end dates, and salary range.

2.6 Contact Info

- Students can list multiple pieces of contact information, each as a string.

2.7 Comments

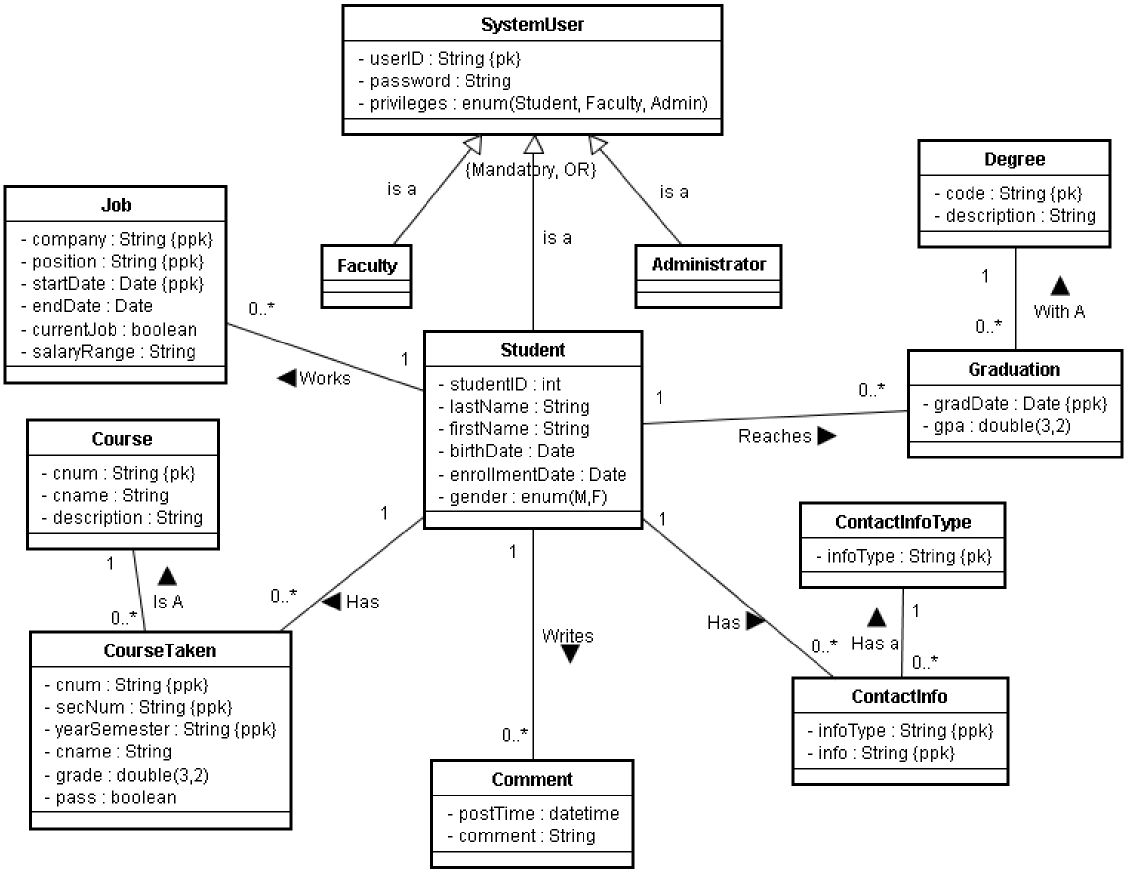
- Students and faculty can post comments over time on a student's page.

2.8 Degrees

- Maintains a list of all degrees available.

3. Entity Description

3.1 Entity Relationship Model



3.2 Entity Attributes

3.2.1 SystemUser

Attribute	Description
userID {pk}	A unique user-created ID that users sign in with
password	A user-created passphrase that is used to verify logins
privileges	A 3-option attribute that determines the access users have

3.2.2 Student

Attribute	*Description*
studentID	A unique integer used to identify each student
lastName	Student's last name
firstName	Student's first name
birthdate	Student's date of birth
enrollmentDate	Student's date of enrollment
gender	Student's gender (M/F)

3.2.3 CourseTaken

Attribute	*Description*
cnum {ppk}	Course number
cname	Course name
secNum {ppk}	Section number
yearSemester {ppk}	Year and semester course was taken (e.g. 2009W or 2008S)
grade	Student's grade upon completion of course
pass	Boolean value determining whether student passed course

3.2.4 Course

Attribute	*Description*
cnum {pk}	Course number (e.g. COSC304)
cname	Course name (e.g. Introduction to Databases)
description	Course description

3.2.5 Graduation

Attribute	*Description*
degree {ppk}	Degree the student graduated with
gradDate	Date the student graduated
GPA	Student's grade point average upon graduation

3.2.6 Job

Attribute	*Description*
company {ppk}	Company the student worked for
position {ppk}	Position the student held in the company
startDate {ppk}	Date the student began this job
endDate	Date the student ended job (NULL if current job)
currentJob	Boolean value determining whether student currently holds this position
salaryRange	Optional field for student to mention salary range for job

3.2.7 ContactInfo

Attribute	*Description*
info {ppk}	Piece of contact information for student
infoType {ppk}	Description of the type of contact info (e.g. homePhone, personalEmail, etc.)

3.2.8 ContactInfoType

Attribute	*Description*
infoType {pk}	A type of contact information (phone, email, etc.)

3.2.9 Degree

Attribute	***Description***
code {pk}	Code of the degree (e.g. B.Sc.COSC), can also include honours (e.g. B.Sc.COSC(Honours))
description	Description of the degree

3.2.10 Comment

Attribute	***Description***
postTime {ppk}	Time the comment was posted
comment	The comment that the student posts

3.3 Relational Schema

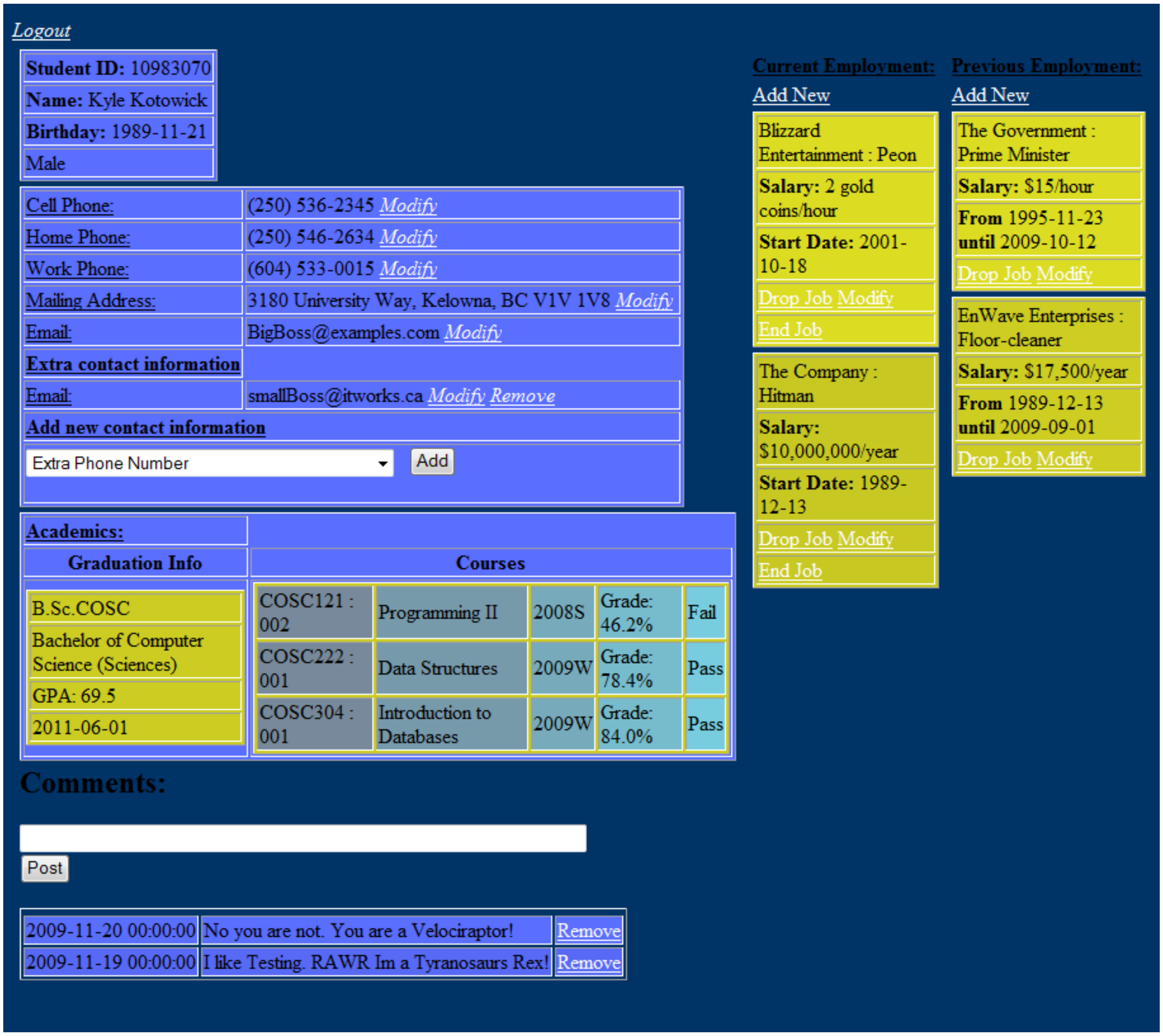
```
DROP TABLE ContactInfo;
DROP TABLE Comment;
DROP TABLE CourseTaken;
DROP TABLE Graduation;
DROP TABLE Degree;
DROP TABLE Job;
DROP TABLE ContactInfoType;
DROP TABLE Course;
DROP TABLE Student;
DROP TABLE SystemUser;

CREATE TABLE SystemUser(
 userID VARCHAR(50) NOT NULL,
 Privileges INTEGER NOT NULL,
 password VARCHAR(12) NOT NULL,
 PRIMARY KEY(UserId)
);

CREATE TABLE Student(
 studentID VARCHAR(50) NOT NULL,
 lastName VARCHAR(50) NOT NULL,
 firstName VARCHAR(50) NOT NULL,
 birthDate DATETIME NOT NULL,
 enrollmentDate DATETIME NOT NULL,
 Gender Bit NOT NULL,
 PRIMARY KEY (studentID),
 FOREIGN KEY (studentID) REFERENCES SystemUser(userID) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Job(
 studentID VARCHAR(50) NOT NULL,
 company VARCHAR(50) NOT NULL,
 position VARCHAR(50) NOT NULL,
 salaryRange VARCHAR(50),
 startDate DATETIME NOT NULL,
 endDate DATETIME,
 PRIMARY KEY (studentID, company, position, startDate),
 FOREIGN KEY (studentID) REFERENCES Student(studentID) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE ContactInfoType(
 infoType VARCHAR(50) NOT NULL,
 PRIMARY KEY(infoType)
);

CREATE TABLE ContactInfo(
 studentID VARCHAR(50) NOT NULL,
 infoType VARCHAR(50) NOT NULL,
 info VARCHAR(250) NOT NULL,
 PRIMARY KEY(studentID, info, infoType),
 FOREIGN KEY(studentID) REFERENCES Student(studentID) ON DELETE
CASCADE ON UPDATE CASCADE,
 FOREIGN KEY(infoType) REFERENCES ContactInfoType(infoType) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Course(
 cnum CHAR(7) NOT NULL,
 cname VARCHAR(100),
 description VARCHAR(500),
 PRIMARY KEY(cnum),
);

CREATE TABLE CourseTaken(
 studentID VARCHAR(50) NOT NULL,
 cnum CHAR(7) NOT NULL,
 secNum VARCHAR(3),
 yearSemester VARCHAR(5),
 grade DECIMAL(4,1),
 pass BIT,
 PRIMARY KEY (studentID,cnum, secNum, yearSemester),
 FOREIGN KEY (cnum) REFERENCES Course(cnum) ON DELETE CASCADE ON UPDATE CASCADE,
 FOREIGN KEY (studentID) REFERENCES Student(studentID) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Degree(
 code VARCHAR(20) NOT NULL,
 description VARCHAR(1000),
 PRIMARY KEY (code)
);

CREATE TABLE Graduation(
 studentID VARCHAR(50) NOT NULL,
 graDate DATETIME NOT NULL,
 gpa DECIMAL(4,1),
 degree VARCHAR(20) NOT NULL,
 PRIMARY KEY (studentID, graDate),
 FOREIGN KEY (studentID) REFERENCES Student(studentID) ON DELETE
CASCADE ON UPDATE CASCADE,
 FOREIGN KEY (degree) REFERENCES Degree(code) ON DELETE CASCADE ON
UPDATE CASCADE
```

```
);

CREATE TABLE Comment(
 studentID VARCHAR(50) NOT NULL,
 postTime DATETIME NOT NULL,
 comment VARCHAR(500) NOT NULL,
 PRIMARY KEY (studentID, postTime),
 FOREIGN KEY (studentID) REFERENCES Student(studentID) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TRIGGER updateGPADegree
ON Graduation
AFTER INSERT
AS
DECLARE @sID VARCHAR(8)
SELECT @sID = (SELECT studentID FROM inserted)
DECLARE @degree VARCHAR(20)
SELECT @degree = (SELECT degree FROM inserted)
PRINT @sID
PRINT @degree
UPDATE Graduation SET gpa=(SELECT AVG(grade) FROM CourseTaken WHERE
studentID=@sID)
WHERE studentID=@sID AND degree=@degree;

CREATE TRIGGER updateGPAaddCourse
ON CourseTaken
AFTER INSERT
AS
DECLARE @sID VARCHAR(8)
SELECT @sID = (SELECT studentID FROM inserted)
PRINT @sID
UPDATE Graduation SET gpa=(SELECT AVG(grade) FROM CourseTaken WHERE
studentID=@sID)
WHERE studentID=@sID;

CREATE TRIGGER updateGPAremoveCourse
ON CourseTaken
AFTER DELETE
AS
DECLARE @sID VARCHAR(8)
SELECT @sID = (SELECT studentID FROM deleted)
PRINT @sID
UPDATE Graduation SET gpa=(SELECT AVG(grade) FROM CourseTaken WHERE
studentID=@sID)
WHERE studentID=@sID;
```

4. Human Interface

4.1 Web Interface

4.1.1 Student Interface



Logout

Student ID: 10983070
Name: Kyle Kotowick
Birthday: 1989-11-21
Male

Cell Phone:	(250) 536-2345 *Modify*
Home Phone:	(250) 546-2634 *Modify*
Work Phone:	(604) 533-0015 *Modify*
Mailing Address:	3180 University Way, Kelowna, BC V1V 1V8 *Modify*
Email:	BigBoss@examples.com *Modify*
Extra contact information	
Email:	smallBoss@itworks.ca *Modify* *Remove*
Add new contact information	
Extra Phone Number ▾ Add	

Academics:

Graduation Info	**Courses**				
B.Sc.COSC	COSC121 : 002	Programming II	2008S	Grade: 46.2%	Fail
Bachelor of Computer Science (Sciences)	COSC222 : 001	Data Structures	2009W	Grade: 78.4%	Pass
GPA: 69.5	COSC304 : 001	Introduction to Databases	2009W	Grade: 84.0%	Pass
2011-06-01					

Current Employment:

Add New

Blizzard Entertainment : Peon
Salary: 2 gold coins/hour
Start Date: 2001-10-18
Drop Job Modify
End Job

The Company : Hitman
Salary: $10,000,000/year
Start Date: 1989-12-13
Drop Job Modify
End Job

Previous Employment:

Add New

The Government : Prime Minister
Salary: $15/hour
From 1995-11-23 **until** 2009-10-12
Drop Job Modify

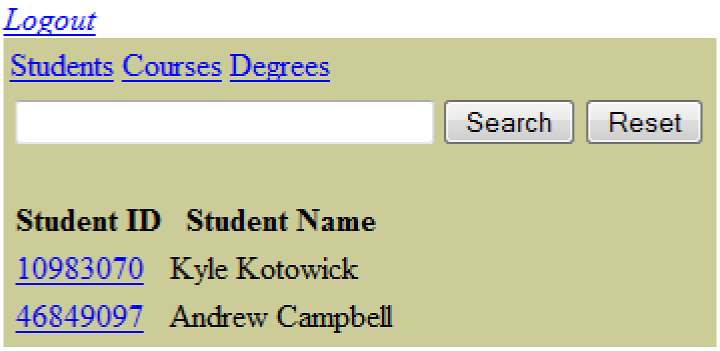
EnWave Enterprises : Floor-cleaner
Salary: $17,500/year
From 1989-12-13 **until** 2009-09-01
Drop Job Modify

Comments:

Post

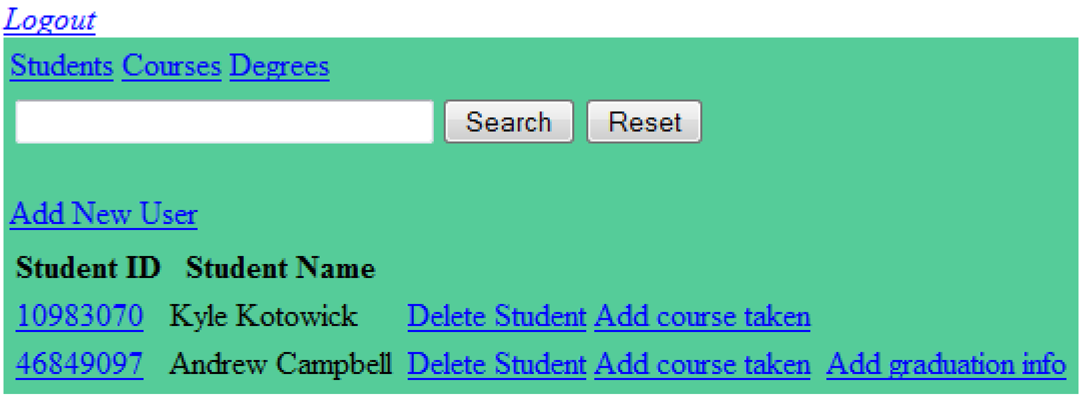
2009-11-20 00:00:00	No you are not. You are a Velociraptor!	Remove
2009-11-19 00:00:00	I like Testing. RAWR Im a Tyranosaurs Rex!	Remove

4.1.2 Faculty Interface



Student ID	Student Name
10983070	Kyle Kotowick
46849097	Andrew Campbell

4.1.3 Administrator Interface



Student ID	Student Name			
10983070	Kyle Kotowick	Delete Student	Add course taken	
46849097	Andrew Campbell	Delete Student	Add course taken	Add graduation info

4.2 Planned Features

4.2.1 Student

- Store vital data: name, student ID, jobs, academic records(if possible)
- automatically update academic records
- enforce data privacy

4.2.2 Faculty

- Parse database based on different attributes: courses, students, companies
- Search through student records

4.2.3 Administrator

- Full ability to see & modify data, where possible

5. Walkthrough

5.1 Student User

5.1.1 Login

For testing purposes, please use the following information to log in as a student:

URL: https://cs-suse-4.ok.ubc.ca/46849097/tomcat/Proj/login.jsp
Username: 10983070
Password: 10983070

5.1.2 Student Page

At this point you will see the standard view all students in the system will have. In the top left you will see basic student information, including the student's ID number, name, birthdate, and gender. Below that is a table containing all of the student's contact information. The top half of this table contains the five "basic" contact information types, which are non-removable but are modifiable (see section 5.1.1.3 for details). The bottom half of the table contains "extra" contact information types that the student has added (along with modify / remove buttons) and an option to add a new piece of information (see section 5.1.1.3).

Below the contact information table you will find the "Academics" table. This is where all graduation and course information is displayed. Here you will see degree names, GPA, graduation dates, and other data pertinent to receiving a UBC degree. Under the "courses" column, all courses that this student has taken will be displayed, including course number, course name, semester/year, grade, and whether the student passed.

Below the academics table is the "Comments" board. Here students and faculty can post comments over time. Each comment is displayed with a "remove" button beside it. See section 5.1.1.4 for details on posting and removing comments.

On the right of the page you will find the "Employment" information. This is where all current and past jobs are displayed (ordered by descending start/end date) with information on start dates, end dates (for past employment), salary, employer, and position. Each job has an option to modify/delete it, current jobs have an option to add an end date, and options are provided to add current or past jobs. See section 5.1.1.5 for details on modifying/deleting/adding jobs.

5.1.3 Contact Information

The GIS has an integrated system that allows users to modify all of their contact information and add new pieces of contact information. To test this feature, first click on “modify” beside the cell phone number. Notice that the new page shows the old information. Change this to a new number or click “reset” to set the value back to the old one. Click “submit”. Notice that the “cell phone” field has now changed to the new value. This action works for all contact information, including the “extra” types.

Now select a type of new contact information from the drop-down menu and click “add”. Type in the value you would like. Notice that a new field has been added under the “Extra Contact Information” table with your new value in it. Now, click the “remove” button beside it. Notice that the field and value are then removed.

5.1.4 Comments

All students have the ability to post on their own comment board, and faculty have the ability to post on any student’s comment board. To test this feature, type a new comment into the empty text box and click “post”. Notice that the new comment is added to the top of the list (they are sorted by descending post time). Now, click the “remove” button beside it. Also, notice how the background colors of older comments become darker as more comments are added. This is designed to provide a visual representation of the relative age of comments.

5.1.5 Employment Records

All students have the ability to add/modify/remove jobs for past and current employment. To test this function, first click on “Add New” under the “current employment” heading. Fill out the information boxes, select a start date, and click “add job”. Notice that the new job has been added to the “current employment” column (location may vary depending on the start date you selected).

Now, click the “modify” button in the job you created. Notice that all fields show the old values. Modify a value and click “modify”. Notice that the job now shows the new information.

Now click the “end job” button and select an end date. Click “end job”. Notice that the job has now been moved to the “Previous Employment” column.

Finally, click the “Drop Job” button for the job you have been working with. Notice that the job has now disappeared. Also notice that the background colour of the jobs become darker as the job becomes older relative to the new jobs.

5.2 Faculty User

5.2.1 Faculty Page

If you are not at the login screen, click the "logout" button in the top left corner. Now login with the following information:

URL: https://cs-suse-4.ok.ubc.ca/46849097/tomcat/Proj/login.jsp
Username: prof
Password: prof

At this point you will see a screen that has three options at the top, "Students", "Courses", and "Degrees". These options allow you to choose the items to display. Clicking "Students" (default) will display a searchable list of all registered students. To test, first try a search (preferably for a student ID that you can see in the list). Next, click a student's ID. This will bring you to the standard student page, with several modifications. First, notice that there are no modification options for contact information, comments, or jobs. Second, notice that the degrees and courses listed are now hyperlinks. These links will take you to a page showing all students registered in those degrees or courses.

Click on the "Home" link in the upper left corner. Now, click on the "Courses" link. This brings up a page that shows all courses listed in the database. Clicking "show all students" will show a list of all students registered in that course. Next, click on the "Degrees" link. This displays a page that shows all degrees listed in the database. Clicking "show all students" will show a list of all students registered in that degree.

5.3 Administrator User

5.3.1 Administrator Page

If you are not at the login screen, click the "logout" button in the top left corner. Now login with the following information:

URL: https://cs-suse-4.ok.ubc.ca/46849097/tomcat/Proj/login.jsp
Username: admin
Password: admin

At this point you will see a screen similar to the "Faculty" main screen, with options for "Students", "Courses", and "Degrees" at the top. These options have the same functionality as those on the faculty screen (see section 5.1.1.6). Beside each student you will see "delete student" and "add course taken" options. For students who don’t already have a degree assigned, you will see a “add graduation info” option. Clicking on a student ID will take you to a page that shows all of the student’s information in a less-graphical format (compared to the standard student page). You will also see a “add new user” link.

5.3.2 Adding Users

First ensure that you are logged in to an Administrator account (see section 5.1.1.7). To test this feature, first click on “add new user”. Fill out the information fields and select the “Student” radio button. Click “add user”. Fill out the information fields and click “add student”. Notice that the student is now in the list.

5.3.3 Adding Courses Taken

First ensure that you are logged in to an Administrator account (see section 5.1.1.7). Now, click “add course taken” beside the student you just added. Select a course from the drop-down menu and fill out the other fields. Click “Add”. You will now see the administrator’s view of the student’s page. If you scroll down to the “Courses Taken” section, you will see the course you just added in the list. Now, click the “Home” button in the top left corner.

5.3.4 Adding Graduation Information

First ensure that you are logged in to an Administrator account (see section 5.1.1.7). Click the “Add graduation info” button beside the student you added in section 5.1.1.8. This page gives you the option to allow a student to graduate with a specific degree. The GPA will be calculated using all courses listed in the “Courses Taken” for this student. Fill out the information boxes and click “Add info”. Notice that the degree is now listed under “Graduation Information” on the student page. If courses are added or removed after a student has graduated, the GPA will automatically re-calculate. Click the “Home” button in the top left corner.

5.3.5 Adding New Courses

First ensure that you are logged in to an Administrator account (see section 5.1.1.7). Now click on the “Courses” button at the top of the page. This will display a list of all courses listed in the database. Similar to the faculty page, if you click on the “Show all students” link beside a course it will list all students who have taken that course. Notice the “Add New” button above the courses. Click it. You can now add a course to the database, which will allow students to have it on their record. Fill out the information and click “add”. Now click the “Students” button at the top of the page. Click the “Add course taken” button next to a student. Notice that the new course is now available in the drop-down menu. Click the “Home” button in the top left corner.

5.3.6 Adding New Degrees

First ensure that you are logged in to an Administrator account (see section 5.1.1.7). Next, click on the “Degrees” button at the top of the page. This will display a list of all degrees listed in the database. Similar to the faculty page, if you click on the “Show all students” link beside a degree it will list all students who have graduated with that degree. Click the “Add New” button above the degrees. Select a faculty/department/school from the drop-down menu and fill out the info boxes. Click “Add Degree”. Notice that the new degree now appears in the list of degrees. Now click the “Students” button at the top of the page. Click the “Add Graduation Info” button next to a student. Notice that the new degree is now available in the drop-down menu.

6. File Descriptions

6.1 addComment.jsp

After receiving information from student.jsp, it associates the new comment with the student in question and adds it to the database. The time of the comment is taken as the server time when the page was accessed. Redirects back to student.jsp.

6.2 addContactInfo.jsp

Receives information from contactInfoPage.jsp, and either adds a new piece of contact information to the database, associated with the student who accessed the page. For security reasons, only students and administrators may access this page. Any other users who attempt to access by URL directly will be redirected away from the page. As well, since the student ID of the student adding the information is stored by the session instead of being passed along through the URL, students are unable to add information to other students. This holds true for all database additions. Redirects back to student.jsp.

6.3 addCourse.jsp

Associates a course with a student, assigns them a grade and if necessary updates their GPA.

6.4 addCourseDB.jsp

Adds a new course to the database. The information taken used is given by newCourse.jsp. A course must exist in the database, via this page, before students can be enrolled in it. Redirects to adminpage.jsp.

6.5 addDeg.jsp

Adds a degree to the database, with information taken from getNDeg.jsp. Before a student may graduate with a degree, this page must be accessed to add that degree to the database. Redirects to adminpage.jsp.

6.6 addGrad.jsp

Associates graduation information to a student. The information is recieved from getNGrad.jsp, with the exception of the GPA. The GPA is calculated server-side via a trigger, which sets the GPA to be the average grade of all courses taken by the student. Redirects to adminpage.jsp.

6.7 addJob.jsp

Taking either information from getOJob.jsp or getNJob.jsp, adds job information to the database.

6.8 addStud.jsp

Takes information from getNStud.jsp, and adds a new student to the database. Redirects to sPageForAdmin.jsp, associated with the student who has been newly created.

6.9 addUser.jsp

After receiving information from getNUser.jsp, it add the new user to the database. If the new user has either prof or admin privileges, it redirects to adminpage.jsp. If the new user is a student, the page redirects to getNStud.jsp, to receive the basic student information.

6.10 adminpage.jsp

Home page for those with administrative privileges. Allows for searching through students, courses, and degrees, adding new users, adding academic information to students, and adding courses/degrees.

6.11 contactInfoPage.jsp

Takes information pertaining to contact information from the user and sends the information to addContactInfo.jsp.

6.12 delComm

Deletes a comment.

6.13 delCourse

Deletes one of a student's courses.

6.14 delGrad

Deletes graduation information from a student.

6.15 delStu.jsp

Deletes a student from the database.

6.16 dropCourse.jsp

Deletes a course from the database.

6.17 dropDeg.jsp

Deletes a degree from the database.

6.18 dropJob.jsp

Removes a job entirely from the database.

6.19 gEndDate.jsp

For transforming a current job to a past job, takes the end date from the user and sends it to loseJob.jsp.

6.20 getNCourse.jsp

Queries information from the user to add course information to a student. Sends the information to addCourse.jsp.

6.21 getNDeg.jsp

Queries the user for information on a new degree. Sends the information to addDeg.jsp.

6.22 getNGrad.jsp

Queries the user for graduation information for a student. The information is send to addGrad.jsp to add the information to the database. This page is only accessible by administrators.

6.23 getNJob.jsp

Queries the user for Job information, sends the information to addJob.jsp. This is assumed to be a current job, and the user cannot set an end date.

6.24 getNStud.jsp

Only accessed via redirect from addUser.jsp, queries for new student information and redirects to addStud.jsp.

6.25 getNUser.jsp

Gets basic information for a new user, and sends that information to addUser.jsp.

6.26 getOJob.jsp

Queries the user for Job information, sends the information to addJob.jsp. This is assumed to be an old job, and the user must set an end date.

6.27 login.jsp

Login page. Initial page seen by the user, only page seen by non-users.

6.28 logout.jsp

Ends the session for the user. Sets all session variables to null, redirects to login.jsp.

6.29 loseJob.jsp

Updates the Job table to reflect the fact that a job now has an end date.

6.30 modCJob.jsp

Retrieves the new information about a job in which the student is currently employed. Sends the information to mutJob.jsp.

6.31 modOJob.jsp

Retrieves the new information about a job which has an end date. Sends the information to mutJob.jsp.

6.32 mutJob.jsp

Recieves information from modOJob.jsp or modCJob.jsp, and updates the Job table to reflect the changes.

6.33newCourse.jsp

Queries the user for the course name, number and description of a new course to add to the database. Redirects the information to addCourseDB.jsp. This page is only accessible by administrators.

6.34 oy.jsp

Simple page berating those attempting access pages without proper privileges.

6.35 Profpage.jsp

Home page for professors. Allows one to search through students, courses and degrees.

6.36 removeContactInfo.jsp

Deletes given piece of contact information from the database.

6.37 setStudentToSee.jsp

Sets an internal session variable for which student to view when accessing student.jsp.

6.38 setStudentToSeeAdmin.jsp

Sets an internal session variable for which student to view when accessing sPageForAdmin.jsp.

6.39 sPageForAdmin.jsp

An alternative layout to the student page for administrators. It is simpler and easier to parse, although it takes much more vertical space.

6.40 student.jsp

Home Page for the student. Shows all information pertaining to the student, and allows modification for all non-academic information.

6.41 validate.jsp

Checks to see if login information from login.jsp is valid. If it is, sets session variables permissions and validUser, and redirects appropriately. If the information is not valid, redirects to login.jsp.

7. Known Limitations

7.1 Graduating with multiple degrees

At this stage, the GIS is incapable of handling multiple degrees per student (each with associated courses). Currently, students may have either no degree (currently student or dropped out) or one completed degree that calculates its GPA from all courses that student has taken. This limitation can be relatively easily overcome by associating courses with specific degrees (simple modification to CourseTaken relation) and allowing multiple degrees to be taken concurrently.

7.2 Form Input Problems

For users who are keen on messing up the system, it is possible to post comments and/or contact information that contains HTML code. This will cause malfunctions and some errors that cannot be fixed without resetting the database relations.

7.3 Empty Forms

There is a known issue with submitting forms (when adding courses, degrees, etc.) that have empty text fields. Depending on what is being added, this may result in an SQL server error.