UBC Graduate Information System (GIS)

Project Design

University of British Columbia Okanagan

COSC 304 – Fall 2009

Version: 1.1
Date: 11/02/2009

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)

Table of Contents

1. Executive Summary

This database will support students, who may view and modify their own data, faculty, who view all student data, and administrators, who may modify all data. The student records will contain the basic identifying elements (names, birthdates, student ID) and graduation information (graduation date, GPA, degree received). Students will be able to add and update personal contact information of any text-based form to the database (such as telephone numbers, email addresses, links to social networking sites, etc.).

Attached to the student records will be a list of all courses the student has completed, as well as the grade the student received in each course. This feature will require access to UBC academic record data in order to be implemented. Finally, the student will be able to store information about all jobs held in the past and current employment. Employment information stored will include company, position, and start and end dates (the student must enter this data manually).

Access to the data will be limited to a set of permissions. Students may access and modify their own data, but are unable to access the data of other students. Alternatively, faculty will have access to all the student records, but will be unable to modify any (other than course grades). Administrators will have full system access and will be able to modify all data. These permissions will be enforced via a login page and separate relations holding accounts and permissions.

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

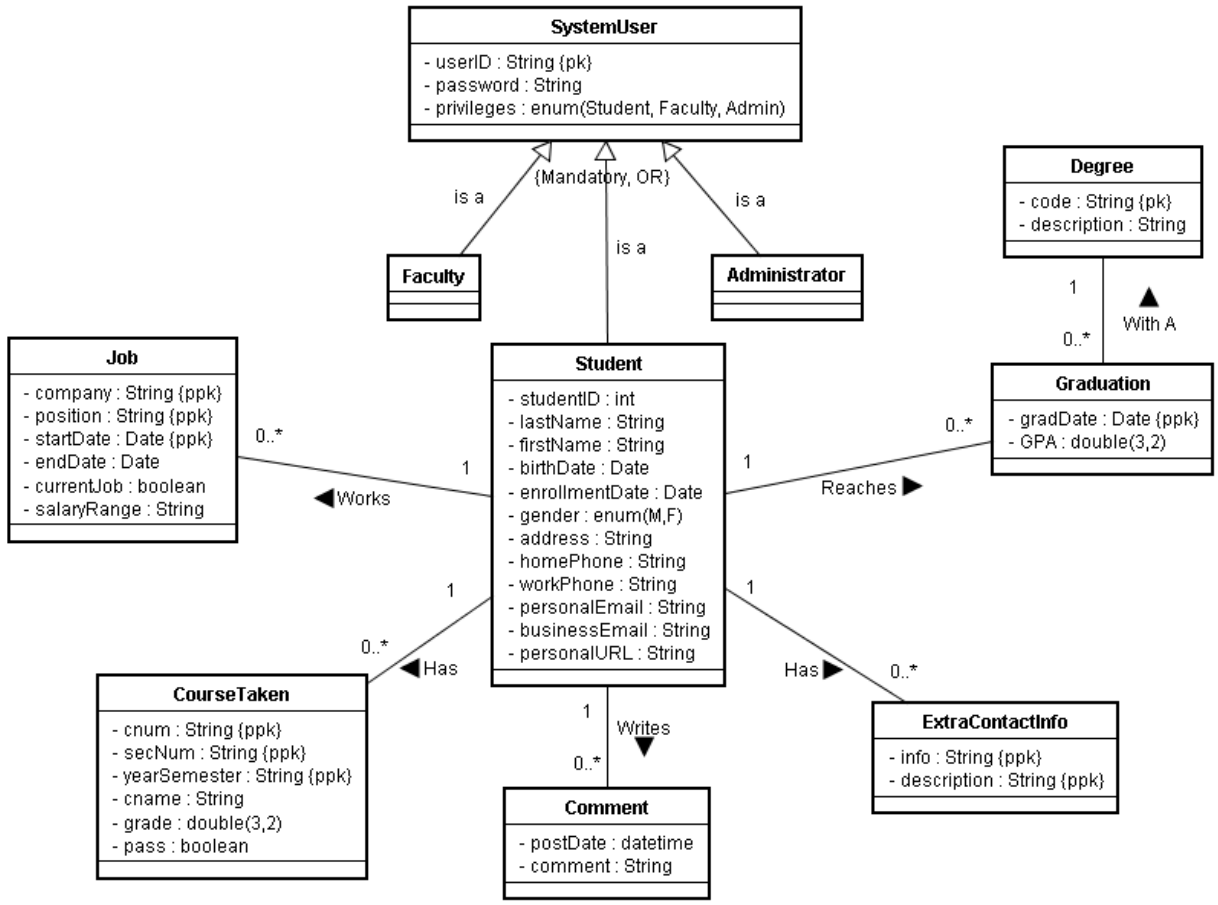
- Students can list multiple jobs.
- Jobs are either current or past.
- Jobs contain info on company, position, start and end dates, and salary range.

2.6 Contact Info

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

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)
address	Student's current address
homePhone	Student's home phone number
workPhone	Student's work phone number
personalEmail	Student's personal email address
businessEmail	Student's business email address
personalURL	Student's personal web site

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 Graduation

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

3.2.5 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

3.2.6 ExtraContactInfo

Attribute	*Description*
info {ppk}	Piece of contact information for student
description {ppk}	Description of the type of contact info

3.2.7 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

3.2.8 Comment

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

3.3 Relational Schema

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

```
);

CREATE TABLE Student(
  userID VARCHAR(50) NOT NULL,
  studentID VARCHAR(50) NOT NULL,
  lastName VARCHAR(50),
  firstName VARCHAR(50),
  birthDate DATE,
  enrollmentDate DATE,
  Gender boolean,
  address   VARCHAR(300),
  homePhone VARCHAR(20),
  workPhone VARCHAR(20),
  personalEmail VARCHAR(200),
  businessEmail VARCHAR(200),
  personalURL VARCHAR(300),
  PRIMARY KEY (userID),
  FOREIGN KEY (userID) 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,
  startDate DATE NOT NULL,
  salaryRange VARCHAR(50),
  endDate DATE,
  currentJob boolean,
  PRIMARY KEY (studentID, company, position, startDate),
  FOREIGN KEY (studentID) REFERENCES Student(studentID) ON DELETE CASCADE ON UPDATE CASCADE
);

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

CREATE TABLE CourseTaken(
  studentID VARCHAR(50) NOT NULL,
  cnum VARCHAR(7) NOT NULL,
  cname VARCHAR(50),
  secNum VARCHAR(3)
  yearSemester VARCHAR(5),    //e.g. 2009W
  grade DOUBLE(3,2),
  pass BOOLEAN,
  PRIMARY KEY (studentID,cnum, secNum, yearSemester),
  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 DATE NOT NULL,
  gpa DOUBLE(3,2),
  degree VARCHAR(20),
  PRIMARY KEY (studentID, graDate),
  FOREIGN KEY (studentID) REFERENCES Student(studentID) ON DELETE CASCADE ON UPDATE CASCADE
  FOREIGN KEY (degree) REFERENCES Degree(code) ON DELETE SET NULL 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
);
```

4. Human Interface

4.1 Web Interface

4.1.1 Student Interface

<table>
<tr><td rowspan="3">Name
Student ID
Birthday
Gender</td><td colspan="2">Current Job (Modify)</td><td>Contact Info(Modify)</td></tr>
<tr><td colspan="2" rowspan="2">Company
Position
Start Date</td><td>Phone</td></tr>
<tr><td>12345566</td></tr>
<tr><td colspan="3">Employment History</td><td>Email</td></tr>
<tr><td colspan="3" rowspan="2">Company
Position
Startdate - enddate</td><td>notmine@gmial</td></tr>
<tr><td>Websites</td></tr>
<tr><td colspan="3" rowspan="3">Company
Position
StartDate - enddate</td><td>Facebook account</td></tr>
<tr><td>webserver</td></tr>
<tr><td>Mailing Address</td></tr>
<tr><td colspan="4">123 Mailing Address</td></tr>
<tr><td colspan="4">456 other address</td></tr>
<tr><td colspan="4">Academics</td></tr>
<tr><td rowspan="4">Degree
GPA
Graduation Date
Honours</td><td colspan="3">Courses</td></tr>
<tr><td>CPSC 1620 Intro Programming</td><td>2.2(Pass)</td><td>(Fall 2009)</td></tr>
<tr><td>SOCI 1000 Intro Sociology</td><td>1.1(Fail)</td><td>(Spring 2008)</td></tr>
<tr><td>INDC 3840 Polynesian Interpretive dance</td><td>4.0(Pass)</td><td>(Summer 2008)</td></tr>
</table>

4.1.2 Faculty Interface

Show: *(Students)* courses *(Companies)* *(Degrees)*		
Search:_____ *(name)* *(Number)*		
Name	**Department**	
CPSC 1620 Intro Programming	*(Computer Science)*	*(Show all students)*
SOCI 1000 Intro Sociology	*(Sociology)*	*(Show all students)*
INDC 3840 Polynesian Interpretive dance	*(Interpretive Dance)*	*(Show all students)*

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