

# UBC Graduate Information System (GIS)

---

Project Design

University of British Columbia Okanagan

COSC 304 – Fall 2009

Version: 1.3  
Date: 11/22/2009

# 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.

# Table of Contents

Project Team .....	1
Document Control Sheet .....	1
1. Executive Summary .....	4
2. Domain Assumptions .....	5
2.1 Users .....	5
2.2 Students .....	5
2.3 Courses.....	5
2.4 Graduation .....	5
2.5 Jobs .....	5
2.6 Contact Info.....	5
2.7 Comments .....	5
2.8 Degrees .....	5
3. Entity Description.....	6
3.1 Entity Relationship Model.....	6
3.2 Entity Attributes.....	6
3.2.1 SystemUser .....	6
3.2.2 Student.....	7
3.2.3 CourseTaken.....	7
3.2.4 Graduation .....	7
3.2.5 Job .....	7
3.2.6 ContactInfo.....	8
3.2.7 Degree.....	8
3.2.8 Comment .....	8
3.3 Relational Schema.....	8
4. Human Interface .....	10
4.1 Web Interface .....	10

4.1.1 Student Interface .....	10
4.1.2 Faculty Interface .....	11
4.2 Planned Features .....	11
4.2.1 Student.....	11
4.2.2 Faculty .....	11
4.2.3 Administrator .....	11
5. Current Progress .....	12
5.1 Introduction .....	12
5.1.1 Purpose of Document .....	12
5.1.2 Document Scope .....	12
5.2 Progress Report.....	12
5.2.1 User Requirements for Students.....	12
5.2.2 User Requirements for Faculty .....	13
5.2.3 User Requirements for Administrators.....	13
5.2.4 Progress Summary .....	13
5.3 Future Plan .....	13
5.3.1 User Requirements for Students.....	13
5.3.2 User Requirements for Faculty .....	13
5.3.3 User Requirements for Administrators.....	14
5.3.4 Future Plans Summary .....	14

# 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.

## 2.7 Comments

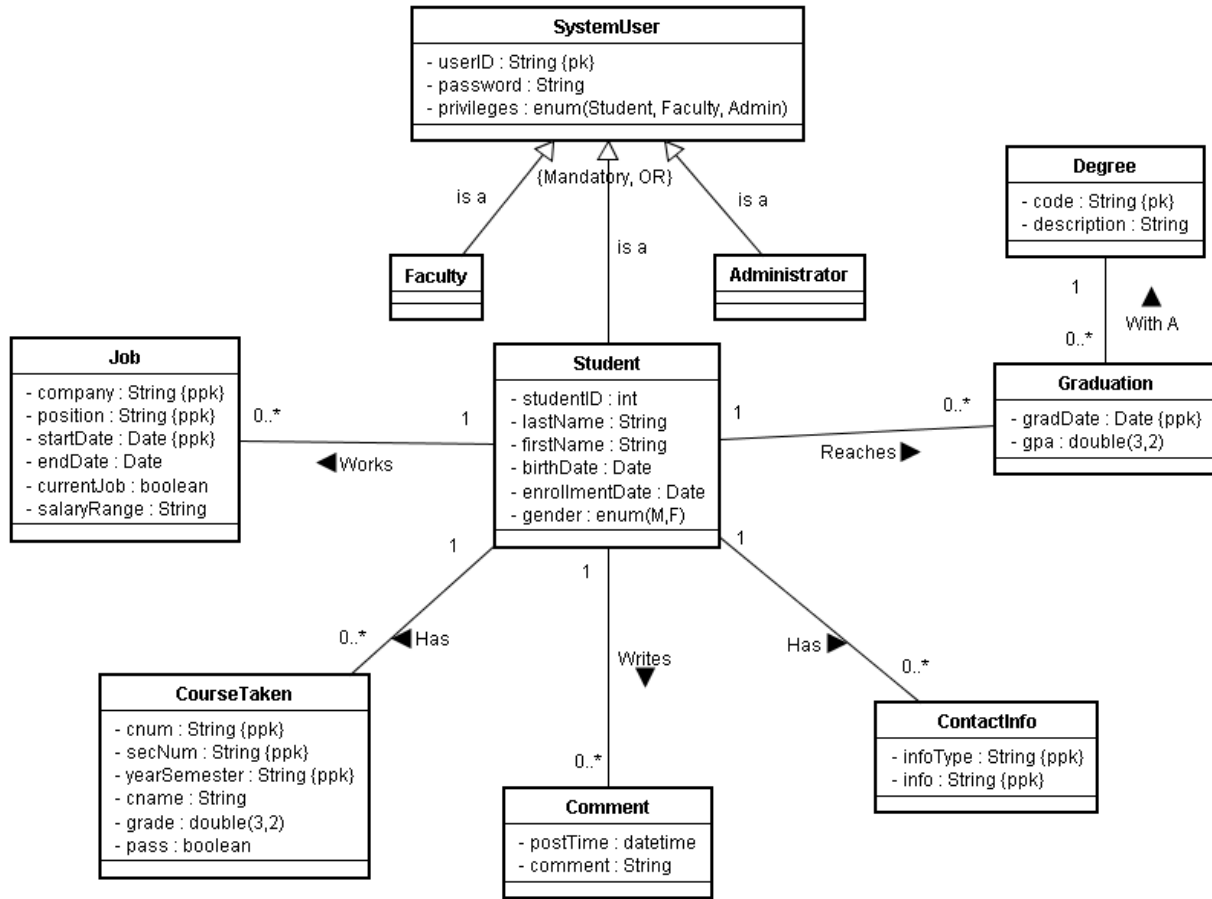
- Students and faculty can post comments over time on a student's page.

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

### 3.2.2 Student

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

### 3.2.3 CourseTaken

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

### 3.2.4 Graduation

<i>Attribute</i>	<i>Description</i>
<b>degree {ppk}</b>	Degree the student graduated with
<b>gradDate</b>	Date the student graduated
<b>GPA</b>	Student's grade point average upon graduation

### 3.2.5 Job

<i>Attribute</i>	<i>Description</i>
<b>company {ppk}</b>	Company the student worked for
<b>position {ppk}</b>	Position the student held in the company
<b>startDate {ppk}</b>	Date the student began this job



<b>endDate</b>	Date the student ended job (NULL if current job)
<b>currentJob</b>	Boolean value determining whether student currently holds this position
<b>salaryRange</b>	Optional field for student to mention salary range for job

### 3.2.6 ContactInfo

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

### 3.2.7 Degree

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

### 3.2.8 Comment

<i>Attribute</i>	<i>Description</i>
<b>postTime {ppk}</b>	Time the comment was posted
<b>comment</b>	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(
  studentID VARCHAR(50) NOT NULL,
  lastName VARCHAR(50),
  firstName VARCHAR(50),
  birthDate DATETIME,
  enrollmentDate DATETIME,
  Gender Bit,
  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 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
);

CREATE TABLE CourseTaken(
    studentID VARCHAR(50) NOT NULL,
    cnum VARCHAR(7) NOT NULL,
    cname VARCHAR(50),
    secNum VARCHAR(3),
    yearSemester VARCHAR(5),
    grade DECIMAL(3,1),
    pass BIT,
    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 DATETIME NOT NULL,
    gpa DECIMAL(3,1),
    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

Name Student ID Birthday Gender	<b>Current Job</b> ( <i>Modify</i> )	<b>Contact Info</b> ( <i>Modify</i> )
	Company Position Start Date	<b>Phone</b>  12345566
		<b>Email</b>
<b>Employment History</b>		notmine@gmial
Company Position Startdate - enddate	<b>Websites</b>	
	Facebook account	
Company Position StartDate - enddate	webserver	
	<b>Mailing Address</b>	
		123 Mailing Address
		456 other address
<b>Academics</b>		
Degree GPA Graduation Date Honours	<b>Courses</b>	
	CPSC 1620 Intro Programming	2.2(Pass) (Fall 2009)
	SOCI 1000 Intro Sociology	1.1(Fail) (Spring 2008)
	INDC 3840 Polynesian Interpretive dance	4.0(Pass) (Summer 2008)
<b>Comments</b>		
Comment 1 2009-08-13		
Comment 2 2009-12-11		
Comment 3 2010-03-10		

### 4.1.2 Faculty Interface

<b>Show:</b> ( <i>Students</i> ) courses ( <i>Companies</i> ) ( <i>Degrees</i> )		
<b>Search:</b> _____		
<b>Name</b>	<b>Department</b>	
CPSC 1620 Intro Programming	( <i>Computer Science</i> )	( <i>Show all students</i> )
SOCI 1000 Intro Sociology	( <i>Sociology</i> )	( <i>Show all students</i> )
INDC 3840 Polynesian Interpretive dance	( <i>Interpretive Dance</i> )	( <i>Show all students</i> )

## 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. Current Progress

## 5.1 Introduction

This section serves as a forum to provide an up-to-date status report on the progress of the GIS project. It consists of a current status on the implementation of customer requirements as well as a plan of future developments.

### *5.1.1 Purpose of Document*

This purpose of this section is to inform the client of and assist in the evaluation of the current progress on the GIS project. It is intended to provide both a clear assessment of the current progress and distinct future plans in order to facilitate customer feedback and suggestions.

### *5.1.2 Document Scope*

As this document is created half way through the development cycle, it is not guaranteed that it serves as an exact reflection of what has been and will be achieved on the GIS project. The only purpose of this document is to facilitate communication between the client and development team at the interim check point.

## 5.2 Progress Report

### *5.2.1 User Requirements for Students*

ID	Description	Comment
	Shows Personal Information	100% done
	Shows Graduation Information	90% done: GPA not a function of Course grades
	Displays Courses Taken	100% done
	Displays Contact Information	100% done
	Update Contact Information	50% done: cannot yet add new type of CI
	Display & Add Comments	100% done
	Security: Only Student may view Grades, no other student can view info	100% done

### ***5.2.2 User Requirements for Faculty***

ID	Description	Comment
	Displays Complete List of Students, Courses & degrees	100% done
	Allows one to search complete list	100% done
	Allows one to view a list of students who have taken a specific course	100% done
	Allows one to view a list of students who hold a specific degree	100% done
	Access aforementioned tables from student or prof pages	100% done
	Allows one to view information on a specific student	100% done
	Reset page after search	100% done
	Security: Professors unable to modify data on student pages, or view grades	100% done

### ***5.2.3 User Requirements for Administrators***

Administrator requirements have not yet been implemented.

### ***5.2.4 Progress Summary***

At this stage, approximately 75% of the Student user's features have been completed. Currently under development are the options to add extra contact information and the automatic updating of the GPA when courses are added or modified.

## **5.3 Future Plan**

### ***5.3.1 User Requirements for Students***

ID	Description	Comment
	Update job status	0% done
	Modify/remove comments	0% done

### ***5.3.2 User Requirements for Faculty***

No known future requirements.

### ***5.3.3 User Requirements for Administrators***

<b>ID</b>	<b>Description</b>	<b>Comment</b>
	Displays Complete List of Students, Courses & degrees	0% done
	Allows one to search complete list	0% done
	Allows one to view a list of students who have taken a specific course	0% done
	Allows one to view a list of students who hold a specific degree	0% done
	Allows one to view information on a specific student	0% done
	Allows one to add students graduation dates, and courses to the system	0% done

### ***5.3.4 Future Plans Summary***

The next stage of development (after completing those requirements currently being worked on) is to complete the “update job status” and “remove comments” features, followed by designing a basic Administrator interface and an “add student/course/graduation” system.