OOP 2016 Spring Semester Project

CST Alumni Information System

Software Requirement Specification (SRS)

2016/2/20

Chunyan Ji

Computer Science and Technology Program

United International College

1. Table of Contents

2. Introduction

2.1. Purpose

The CST Alumni Information System (**CSTA**) system is aimed to collect CST Alumni information, and analyze the graduation and career information about all CST alumni.

In many occasions we need to provide information about our alumni, for example, how many of them went to the graduate schools, how many of them started work for industries, etc. **CSTA** will be used for teachers to enter alumni information. With all the info collected, the system can generate valuable analysis data about CST graduates.

(Notes: the following instruction is not supposed to be in the SRS. It is for some tips for your design.)

Three-tier architecture should be used in this project.

1. *UI: Java Swing is to provide GUI*
2. *Java: Business logic should be written in Java classes. UML diagram should be designed first, and have the strarUML generate the basic java classes for you.*
3. *Database: MySQL will be used as the database. Design your tables to store information need for the system. JDBC technology will be used to connect to the tier 1 Java classes to the database class.*

2.2. Document Conventions

This document uses Arial font style to present its content.

This SRS is written by Gigi. Students should consider her as your customer. Communicate with Gigi on user requirements.

This SRS is used for your to design your system. Students can add more requirements to make the system better. You just need to discuss your thoughts with Gigi before finalizing your system design.

3. System Features

3.1. User of the system

3.1.1 **The user of the CSTA system is the alumni coordinator. He/she will be the only user.**

The Alumni Coordinator is the only user of the CSTA system. That means the system does NOT need to have “Login" feature.

3.2. Enter Alumni information

3.2.1 **Description**

The system will allow the user to enter the alumni basic information: name, ID, sex, major, Graduation Year, What to do after graduation (Graduate School or Working, or stay home), etc.

3.2.2 **Functional Requirements**

REQ-1: The user enters alumni information; the system will store the info into the database.

REQ-2: The system should allow the user to search for an alumna’s information using the student's ID.

3.3. Enter Graduate School information

3.2.1 **Description**

If the student goes to graduate school, the system will allow the user to continue to enter the student's graduation school information: Name of the university, country, city, rank, etc. Make the rank a selection box with two selections: top 100 universities or not.

3.2.2 **Functional Requirements**

REQ-1: The user enters alumni's graduate school information; the system will store the info into the database.

REQ-2: The system should allow the user to search for graduate school information such as university name, country, city, etc. The user can use student ID to search the graduate school, which that student joined. But the system does NOT need to show which students joined this graduate school.

3.4. Enter Company information

3.2.1 **Description**

If the alumni started to work for a company, the system will allow the user to continue to enter the company information: Name of the company, country, city, etc.

3.2.2 **Functional Requirements**

REQ-1: The user enters alumni's company information; the system will store the info into the database.
REQ-2: The system should allow the user to search for the company information using the student's ID if the alumni joined the company after graduation. But the system does NOT need to show which students joined this company.

3.5. Analysis

3.2.1 Description

The system will allow the user to do data analysis according in two ways: 1. based on graduation year. 2. All alumni so far.

3.2.2 Functional Requirements

REQ: The system will be able to show to the user (each year or in total):

1. The percentage of the students who went to graduate school after graduation.
2. The percentage of the students who started working after graduation.
3. The percentage of the students who stayed at home after graduation.
4. The percentage of the graduates who studied or worked overseas.
5. The percentage of the graduates who studied in the top 100 universities.
6. The above data should be calculated in two forms: Overall percentages and percentages for each year.

3.6. Extra Points

3.2.1 Description

Design your system to be able to expand to UIC instead of CST. The system will not only be able to handle CST alumni, it will also be able to record all UIC alumni, and show the analysis on all UIC alumni.

3.2.2 Functional Requirements

REQ-1: The system should be able to accept alumni information of all majors.
REQ-2: The system should be able to show the analysis data of all UIC alumni.

1. The percentage of the students who went to graduate school after graduation.
2. The percentage of the students who started working after graduation.
3. The percentage of the students who stayed at home after graduation.
4. The percentage of the graduates who studied or worked overseas.
5. The percentage of the graduates who studied in the top 100 universities.
6. The above data should be calculated in two forms: Overall percentages and percentages for each year.

4. External Interface Requirements

4.1. User Interfaces

GUI Swing will be used for user interface.

Design and draw your UI below.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

If we expand this project to be used by all UIC students, you system should be able to handle it. Design your system to have the ability to be used 50000 students. Test system performance with more users.

5.2. Security Requirements

N/A

5.3. Software Quality Attributes

1. The system must go through system testing and never crashes when the user inputs incorrect data.
2. Make the system scalable, so it may be used by all UIC students.
3. All the Java classes, methods should be designed in a way that is easily to be accessed by GUI interface. Consider we will reuse the code in the future.
4. Write necessary comments in your code.