Carleton University TA Application and Evaluation System *(cuTAES)*

Project Description
by Christine Laurendeau

Background

Every term, the School of Computer Science hires dozens of teaching assistants (TAs) to help course instructors provide the highest quality learning experience for students. The successful candidates for TA positions are selected from a wide range of qualified applicants. As a result, a friendly and easy-to-use system is required to manage student applications for TA positions. The current approach for reviewing student applications for TA positions is predominantly a manual process. Furthermore, when course instructors need to evaluate their TAs' performance at the end of the term, they have no system currently in existence to electronically enter their TA evaluation information. Convenient access to this data is essential for continuing to hire only the best applicants as TAs. The Carleton University TA Application and Evaluation System (*cuTAES*) aims to address this lack of automation.

System Requirements

The complete *cuTAES* system must support the following two types of users:

- Student applicants
- Administrators

The system must provide the following features for the student applicant users:

- *Create an application*

  The system displays a pre-configured list of courses being offered in the current term, and the applicant selects a course for which to apply to be a TA. The user is prompted for their *general information*, including student number, first and last names, email address, major, year standing, CGPA, and major GPA. This information is only entered once per session. The applicant is then prompted for information *specific to the course* for which they are applying. This specific information includes a list of related courses taken (with term, year and final grade), a list of related courses for which the applicant has already been a TA (with term, year and name of supervisor), a list of relevant work experience (with responsibilities, duration, start and end dates), and a list of faculty references selected from a pre-configured list of faculty members. Once an applicant has entered all the information and saved the application, they are prompted to indicate whether they want to create a new application to be TA for another course, or whether they are done with the application process. If they choose to apply for another course, their general information is automatically filled in for the new application, and they only need to enter the information specific to the new course. New applications are assigned an automatically generated application number, and they are set to a status of *pending*. They are also saved to permanent storage in a file.

- *Edit an application*

  Student users can view their applications in order to edit them. They cannot view or edit applications created by other users. All the information entered by the applicant can be modified. The applicant can also cancel an application if they are no longer interested in being TA for a course. If they do so, then the status of the application automatically changes to *closed*. *Closed* applications can be viewed, but they cannot be edited. Edited applications are saved to permanent storage in a file.

The system must provide the following features for the administrator users:

- *View an application*

  An administrator can view an application for a TA position in the following ways:
  - by entering an application number
  - by entering an applicant's student number
  - by entering an applicant's name (first or last or both)

  If more than one application fits the selection criteria, they are displayed one at a time, in the same format as the student user enters the data. The administrator can choose to view the next application, or the previous one, or return to the main menu.

- *Assign successful applicant*

  From the *View an application* feature, the administrator can assign the TA position to the applicant, meaning that the applicant has been selected as a TA. This action automatically changes the status of the application from *pending* to *assigned*. It also changes the status of all other applications by this applicant to *closed*, since a student can be TA for only one course per term, so only one of their applications can be successful. An application that is already *closed* cannot become *assigned*.

- *View a summary of* pending *applications*

  An administrator can view a summary of applications with a status of *pending* in the following ways:
  - for one selected course, sorted by applicant's major GPA
  - for all courses, sorted first by course number, then by applicant's major GPA

  A report-type summary is created and displayed. The user also has the option to save the summary to a file. The data included in the summary consists of the course number and the applicant's *general information.*

- *View a summary of* assigned *applications*

  An administrator can view a summary of successful applications, i.e. all applications with a status of *assigned*. A report-type summary is created and displayed. The user also has the option to save the summary to a file. The data included in the summary consists of the course number and the applicant's *general information.*

- *View and edit a TA evaluation*

  For every successful application, i.e. one that results in the applicant being hired, a corresponding blank TA evaluation is created automatically. The administrator can edit an evaluation by entering the evaluation number, or by entering a TA's student number, or by entering a TA's name (first or last or both). TA evaluations are saved to permanent storage in a file.

Constraints

The *cuTAES* system must abide by the following constraints:

- Source code
  - The code must be written in C++
  - The code must compile and execute in the 32-bit VM provided for the course
  - No libraries or library elements other than the ones explicitly listed below can be used
  - The Curses library or the GTK+ library **can** be used
  - Elements of the C++ standard library that **can** be used (see this C++ library reference for details):
    - From the *C library*: functionality that is not otherwise available in C++
    - From the *Containers* group: nothing!
    - From the *Input/Output* group: everything
    - From the *Other* group: only `<string>` and `<stdexcept>` can be used
  - The use of any other element of this library and the use of any other library is **strictly prohibited**
- User Interface (UI)
  - The UI must use the Curses or GTK+ library; its general look-and-feel should be professional and easy to use