CS 346 – Individual Project – Part 1: Functional Specification

Due: Monday 9/25/2017 at 8:00PM

100 points [counting for one fifth of your individual project grade, i.e., 3% of your course grade]

*This is an **individual** project*

UWO WebCLICKER is a web-based classroom response system that enables an instructor to ask questions of his or her students during class and adapt his or her teaching dynamically based on the collected answers. You are already familiar with a prototype of this app since you are using it in CS 346 this semester. The main goal of the term project in this class is to build a final product that is fully functional, usable, efficient, and secure. The first couple of stages (i.e., requirements analysis and design) in this project will be completed individually. The bulk of the project will be completed in teams. This first part of the project deals with requirements analysis and is to be completed individually.

This document will first list the functional features of the current prototype. Then it will ask you to produce a list of features that users of this app would find useful. As you know, this web app has two sides, the student side and the instructor side. Therefore, the two main categories of stakeholders will be students and instructors. Your task is to collect (and rank) a list of the functional features that each stakeholder would want. You are free to interview fellow students (outside of CS 346!) and instructors (same comment!) who have used a clicker system and/or those who have expectations about what an ideal clicker system would be able to do to enhance learning.

First, here is a list of the existing functional features of the prototype used in class.

Student side:

- Password-protected log in and access to all (and only) student functions
- Password change
- Record login stats (number of password changes, last login time, last logout time) in a database
- Request the active question; display the active question (or an error message); submit student answer to the server; score the answer and store answer/score in a database
- Review past questions as well as student's scores on these questions
- Explicit log out (to clear session data)

Instructor side:

- Password-protected log in and access to all (and only) instructor functions
- Password change
- Pre-load questions into the database; each question must come with an automated grader program for the student's answer
- Select and activate one question at a time; collect students' answers and display them as they come in using a bar chart, as well as a timer for the activation time
- Deactivate the current question; store statistics about the question (date, start time, end time, average score, number of correct answers) in a database
- Deactivate all questions (to deal with questions that were left active by mistake or that are activated too early)
- Display the scores for all students and all questions asked on a given day; export this data to a spreadsheet.

- Explicit log out (to clear session data)

Second, you are asked to provide two lists of *additional* functional features for the UWO WebCLICKER app, one list of features for the student side, and one list for the instructor side. Each list must contain at least eight features and must be ranked from most important to least important. Each feature must be as precisely described yet remain as concise (one or two sentences) as those listed above. Make sure that your features overlap as little as possible, that is, each feature should be significantly different from all other features in its list.

Your grade will be proportional to the number of “good” (i.e., useful, clearly stated, original, etc.) features you include. There is no upper limit on the number of features you submit. Make sure to format your submission as two separate, ordered lists. Do not limit yourselves to features that you think you can implement. Include ALL features that you think are best for this app. You will only be asked to implement a subset of the suggested features and I will make sure to require only features that are doable with the techniques that are covered in this class. However, you can and should think big(ger) for this part of the project. Think of it as a wish list for the very best clicker system that you (and the instructor) could get if time and money were not an issue.

In addition to the two required lists just described, feel free to include a third (optional) list of features that would be useful for a third stakeholder, namely the maintainer of this system, which may not be an instructor nor a student, but a developer/system administrator. I am thinking for example of a server log feature. Can you think of others?

Note that to earn a good grade on this part (which amounts to a significant 3% of your overall course grade), you need to devote a good amount of time, thinking, and research to it. Shoddy work will be easy to spot and penalize. Start early and use the week you are given wisely. You should definitely think of this as an open-ended assignment where the requirements listed above (e.g., 8 items per list) are only minimal ones or lower bounds.

Submission procedure

Before the deadline, submit a single text file (or pdf, Word, OpenOffice document; but no other format please) to the D2L dropbox for this project, namely *Pindiv1*.