Logan Bissonnette
(V00729109)
Fraser DeLisle
(V00753730)
Derek Roberts
(V00698880)

SENG 310 Final Project Report

Summer 2012

Table of Contents

Executive summary 3
Project and Group Information 3
Domain and Stakeholders 3
Tasks 4
Prototype 5
    Defects in the Final Prototype 6
Usability Requirements 7
    Data collection and analysis: 7
Evaluation 8
Discussion and Recommendations 8
    All I want to do is add an answer... 8
    How do I submit this? 9
    Did that quiz just start? 9
    Where do I need to go? 9
    I need to click on that text? 9
Demo Evaluation Sheet 10

Executive summary

The final product of the project was a prototype of a full online learning environment system, with the capability to be expanded into a proctoring system as well. We came up with the design goals for our project based on the results of our interviews with teachers and students from UVic and other universities.

The main issue we had with the project was with Java Swing and NetBeans; neither provided a user friendly experience and we spent more time trying to overcome these limitations than we did working on the user interface.

We did usability testing on only one of our main user groups - students. Although we did not have any teachers test our project, we felt that we had satisfactory results from testing with students. Another group of students evaluated our system based on heuristics. These students provided us with feedback on how our system conformed to Nielsen's Usability Heuristics.

We feel that the GUI works well and we recommend that future developers continue to work on adding additional modules. The prototype we have can be turned into a production system, but more modules should be added such as forums, calendars, and a UI for students to access resources.

Project and Group Information

Our group consisted of three members; Logan Bissonnette, Fraser DeLisle, and Derek Roberts. Each group member took on a unique role when creating the project.

Fraser DeLisle was the group leader and coordinator. He worked on making sure progress was made, as well as working on integrating all of the different java files so different modules could be loaded as well as working on some of the modules himself.

Logan Bissonnette worked mainly on modules and java swing. He fixed up the bugs and issues in the project as well and working on error conditions and testing. Logan also spearheaded the documentation, and worked on editing and formatting all of the written reports.

Derek Roberts did 90% of the Lumzy prototyping. He worked tirelessly on the initial design and came up with a good looking paper/Lumzy prototype. Derek also worked on some of the modules and design in java swing on the final portion.

Domain and Stakeholders

Online education is a growing domain, and because of this more software is becoming available for use. One of the main concerns about distance education is the concern of students cheating. We feel that this part of the domain is the biggest challenge because no matter how hard we try to prevent students from cheating; students will always find a way to cheat. For our project, we leave the job of preventing

cheating to 3rd party software and focus more on the educational environment aspect of the project.

The primary stakeholders in our system are the students and teachers who are going to be working directly with the system.

Teachers need to be able to create and mark both tests and assignments as well as upload resources. Other features such as the calendar and forums must also be usable by the teachers.

We will have many students who will also need to interact with the system. Students must be able to access resources and take tests and assignments.

Our secondary stakeholders are the proctors. Proctoring wasn't a much requested feature; the people we interviewed were not interested in this feature. We feel like the proctoring component is a secondary goal in the system. Proctoring on the system can take place through other authentication methods, such as 3rd party facial recognition software, mouse and key press tracking software, or biometric software. Because of this proctors are not a primary system user.

Tasks

There were many tasks which we would have developed if this project were to go to production. The following tasks have been implemented partially or fully by our system.

Student Tasks	Teacher Tasks
Login	Login
View new announcements and information	View new announcements and information
See class information	Create new test/quiz
Take quizzes/tests	Upload new resources.
Change application settings	Send notification to students/TAs about new resources
Logout	Change application settings
	Logout

There were many tasks which we wanted to have implemented but did not make it into our final prototype. These are the tasks we would have implemented if we continued with the system.

Omitted Student Tasks	Omitted Teacher Tasks
Username/Password checking	Username/Password checking
GUI for changing settings	GUI for changing settings
Access uploaded resources	Edit class information
See calendar events and due dates	Set up calendar events and due dates
View and submit assignments	Upload new assignments (currently done by adding a new resource)
Forums/chat dialogues	Forums/chat dialogues
	Proctor view
	View test/quiz results

These features were not added because we did not have the time to add them, or because we felt that they were beyond the scope of the assignment.

The use cases we implemented were the ones we felt were the primary uses of the system. These use cases were: teachers adding resources, teachers adding quizzes, and students taking quizzes. The discussion of these use cases and which were chosen were covered in earlier reports.

Prototype

Our prototype focused on user feedback and system flow. We wanted to create an organized, usable GUI that was intuitive such that a new student could log in and figure out how to take a quiz and logout without any assistance.

We feel that if the users were able to successfully complete all of our use cases without any assistance then we will have fulfilled our requirements.

Defects in Initial Prototype

The following are issues identified with our first iteration of a high-fidelity prototype and do not reflect the problems with our final prototype.

General usability defects

- ComboBox with the user's name was not an appropriate spot to put the options for settings and logoff, due not being similar objects.
- Accordion widget was not intuitive. Users were unable to figure out how to work with it.
- Buttons often had different sizes, and did not always have enough whitespace.
- Components resized themselves frequently.
- No shortcuts to make navigation faster.
- GUI wasn't similar enough to other applications systems and not as easily operated.
- The system did not contain any in-system help - always requires offline documentation.

Usability defects for uploading a new resource

- No confirmation of successful upload.
- No option to upload multiple files.
- No error or helpful message if no file uploaded at all.
- Notify options and checkboxes not self-explanatory.

Usability defects for creating a new test

- Too many redundant buttons.
- Buttons are neither lined up nor grouped by functionality.
- Navigation with ComboBoxes unclear, should change to Previous and Next buttons.
- Overall process and layout for creating questions is unclear and complicated.
- User may enter a negative number of questions or sections.
- Having a single entry and ComboBox for multiple choice questions is non-intuitive.

Usability defects for taking a test

- Window title missing.
- No progress bar.
- Too much white space according to one user.
- Buttons labelled inconsistently.
- No option to save and quit.
- Flag Procto button unclear.
- No confirmation after clicking Finish or Cancel buttons.
- Unable to save and return to answers.
- No feedback if any questions are missed or skipped.
- Test is in popup window rather than main window. Leaves course materials available for cheating

Defects in the Final Prototype

Many of the above defects have been fixed in our final prototype, but not all of them were. We have added some functionality such as progress status during quizzes, we have redone the test creation system, and we have fixed the sizing of components throughout the system.

Unfortunately some fixes were not implemented, often due to issues with Java Swing. For example, we were unable to change the system so that you are unable to use the main system during the test - this is because the application stopped functioning when the test window was made modal. This is an issue with how NetBeans sets up the code for the application when using their GUI builder.

Some features such as the ability to save and resume later do not exist because the system is a prototype. In the final version a lot of this functionality would happen automatically without the need for pressing a save button.

Usability Requirements

For our online proctoring system, we identified numerous usability requirements by interviewing potential users from all of the user groups which were recognized to be major stakeholders in the software system. Following is a list of the most important usability requirements we found.

Qualitative usability requirements:

- The software system should be easy to learn.
- Frequently used features should be readily available.
- Names and locations of features should be recognizable.
- Error states should be prevented.
- System should give comprehensible error messages.
- Users should be able to correctly identify system state.

Quantitative usability requirements:

- Time required to create a 20 question test should be less than 15 minutes.
- The system should be able to run error-free for more than 2 hours.
- User should be able to accurately predict the results of pressing a button 90% of the time.
- User should be able to find the correct button to press to find a specific feature in less than 5 seconds.
- Users should make no more than 1 incorrect button press when trying to find a specific feature.

Metrics:

- Time between choosing an action to perform and locating the corresponding button/link etc.
- Number of clicks/keystrokes required to activate specifically frequently used features.
- Minutes taken to create a test with 20 questions.
- Number of results of specific button presses a user can correctly identify.
- Number of incorrect buttons pressed when trying to activate a specific feature.

Data collection and analysis:

All of the data on the requirements listed above could be collected by a combination of direct and indirect user observation methods. Indirect user observation with video recordings would be the best method to collect data on timing of actions, time required per task, etc. Direct observation would allow for collection of finer details, such as how intuitive the locations of features are. Data collection could be performed in a controlled lab setting.

The quantitative data could be analysed by descriptive statistics such as standard deviation and by inferential statistics if the number of subjects evaluated is high enough. Visual data representations including graphs and tables can be used to visually analyze both the quantitative and qualitative data.

Evaluation

Most of the defects were found during the usability and heuristics inspections that were performed on our project. From the walkthrough evaluation and usability inspection we found our system for the iteration that it was performed under.

The main issues with the system were around navigating to specific modules (ie. finding out where to go), using the test creation module, and identifying system state. To resolve these problems we changed buttons and labels to be more straightforward, we redesigned the test creation section, and we added dialog boxes, titles, and other identifiers to help users tell what state the system is in.

Since then we have fixed our system. We brought back another user to try out the system again, and found that the usability of the system did improve, but we do not have enough data to make any conclusions about the usability of the system overall.

A summary of the defects of the system can be found in the defects subsection under the prototype section of this report.

Discussion and Recommendations

Most of the issues with our project have been resolved. We have worked hard to make sure that our final prototype is as usable as possible. Following are issues we uncovered during the making of our project.

All I want to do is add an answer…

Our page for creating tests was difficult to use. Users were unable to identify how the different test sections were supposed to work, how to create new answers for multiple choice questions, nor how to mark which answer is correct.

Part of the problem was the inappropriate use of specific GUI components such as spinners, as well as having multiple methods of adding new questions.

We greatly simplified this part of the GUI and now feel that it is perfectly usable.

How do I submit this?

Many modules of our project were missing a submit button. Users would create a quiz, or complete a test and were unable to figure out how to actually save and complete their task. This has been resolved in most modules.

Did that quiz just start?

In one of our iterations of our prototype, quizzes and tests started as soon as you pressed the button to open them. This left users unsure of what just happened - they weren't certain if the quiz started or what the new window was for. We fixed this by adding a dialogue to let the user know that they were about to begin a new quiz.

Where do I need to go?

In one iteration of our prototype, we did not use simple names for common features. By using names such as 'Interactive Testing' some users were confused as to what the section actually contained. Once users figured our labels meant they could recall it, but using complex names for simple sections did not make the system more usable.

To fix this we simplified the names on many features. For example, 'Interactive Testing' became 'Tests and Quizzes' to help the user identify where they need to go.

I need to click on that text?

In our accordion widget we had text links that would open up new functions. These text links could only be clicked once, and had to be deselected and reselected to be able to click them again. To make matters worse, the text links didn't even look like text links, they looked just like plain text.

It took a lot of work, but we changed the text links into buttons. Unfortunately adding buttons in the panels did not help because they blended in with the rest of the accordion bar.

Eventually we formatted the buttons to look more like HTML hyperlinks. We feel that this provided the user with a familiar experience, and the functionality of the links should be clear to users in our target audience.

Demo Evaluation Sheet

SENG 310 -HCI (Final Project Demo)		Evaluation Criteria (assign a value between [1 – 10] or do not know)			
Monday, July 30, 2012		***Does the prototype satisfy the requirements of an OLP?***	***Does the prototype show the important tasks and scenarios of an OLP?***	***Is it a high fidelity prototype?***	***Is the GUI Appealing to you?***
MS01: 8:30 - 8:40 am	*Richard McKenzie, Jian Guan, Nicolas Guillemot, Ben Mahlman and Meghan Reid*	7	7	7	10
MS02: 8:45 - 8:55 am	*Daniel Potts, Brendan Clement and Travis Gorbahn*	***Not Applicable***			
MS03: 9:00 - 9:10 am	*Adam Anderson, Geoff Gollmer, Jamie Bracke and Ryan Tandy*	7	7	6	6
MS04: 9:15 - 9:25 am	*Fraser DeLisle , Logan Bissonnette and Derek Roberts*	10	10	10	10
MS05: 9:30 - 9:40 am	*Echo Yu, Evian Zhou, Hiroki Yun, Jeff Li and Richard Hsu*	8	8	6	5
MS06: 9:45 - 9:55 am	*Dong Zhou, Tsu-hao Chang, Matt and Ding Zhang*	7	7	7	4

Thursday, August 2, 2012		*Does the prototype satisfy the requirements of an OLP ?*	*Does the prototype show the important tasks and scenarios of an OLP?*	*Is it a high fidelity prototype?*	*Is the GUI Appealing to you?*
TS01: 8:30 - 8:40 am	*Micheal Atavine, Rowan MacLeod, Ryan Beldessi and Ben Cunningham*	8	6	7	8
TS02: 8:45 - 8:55 am	*Jeremy Moseley, Adrian DeLisle , Colson Driemel and Kai Fuglem*	9	7	7	5
TS03: 9:00 - 9:10 am	*Marcus Greenshields, Brad Lucas, Daniel Remesch and Cale McNulty*	7	8	8	10
TS04: 9:15 - 9:25 am	*Justin Guze, Ian Brown, Scott Low, and Paul Hunter*	8	8	9	5
TS05: 9:30 - 9:40 am	*Adrian Kilian, Isabelle Dufour, and Mike Newton*	9	8	8	9
TS06: 9:45 - 9:55 am	*Sid McLaughlin, Michael Peng, Chris Batt, Amanda Dash and Romil Khanna*	7	7	7	6
TS07: 10:00 - 10:10 am	*Ali Alsaihaty, Hanying Lu, Jorge Conde Gomez Llanos and Xiuxia Chen*	8	7	9	9