Code Review

Project 4, Group 14: T. Kelly, A. Kuang, M. Lee, J. Sizoo, M. Swartz

EECS 448, Dr. David Johnson

Code Review Process

My team performed a code review for the StudentUI form on 4/22. I (Thresa) acted as the reviewer and asked the team members questions about the functionality of the code. The review took about an hour to complete. The discovered defects were recorded in our team's defect tracker. The table in the Defect Tracking section of this paper lists the recorded defects.

A walkthrough code review consists of two steps: preparation and analysis. To prepare for this review, we outlined the requirements of the student window. The form needed to let the user view and edit their account info and to view, drop, or enroll in classes.

Next, I asked the team how the StudentUI was implemented. The user interface was a windows form in Visual Studio written in C#. The file included Dapper libraries, which allowed the form to access the database. The UI was created using the Visual Studio designer. In the designer, widgets were placed and styles on the student window. Functions were assigned to the widget's events. The backend logic of the form can be described as database driven development. Student information is pulled from the following database tables: student, courses, and enrollment.

After reviewing the requirements and describing the functionality, our team looked for defects. I asked specific questions about each widget's function, appearance, and handling of edge cases. Each of the defects were listed in our defect tracker, which is shown in the red column in table below.

After completing the code review, the team worked to fix the discovered defects. The fixes were added to the defect table in the green columns. The code review was a very effective way to verify that our software worked as expected.

Defect Tracking

Date Reported	Who reported	Description of defect	Date Fixed	Who fixed it	Description of how it was fixed
4/22	Thresa	In StudentUI, clicking the left listbox opens the appropriate tab and the clicked listbox text is highlighted. Changing the tab manually (using the tabs at the top) does not change which listbox text is highlighted.			
4/22	Thresa	In Student UI, the bottom of the tab box does not align with the calendar	4/22	Justin	Extended the tab box using the designer to align with the bottom of the calendar
4/22	Thresa	In Student UI, changing the window size/tab window size does not expand the displayed tables. It would be nice if the tables would anchor to the size of the tab box.			
4/22	Thresa	In Student UI, set the account information text boxes to read only. Don't allow user to type/edit them.	4/22	Justin	Each text box contains a property 'readOnly' that can be set to either true or false. By changing the property of these text boxes to true, the user can no longer edit them.
4/22	Thresa	In Student UI, editing the email and submitting it results in a strange font size	4/22	Justin	Had to resize the overall form so that each textbox could be expanded and made larger. This resulted in some reordering of other elements. However, once every textbox was made larger, the email-textbox was large enough to show an email-length sized string at the same font as each other textbox.

4/22	Thresa	In Student UI, in the enrollment table, clicking on a box should select the whole row, not just the single box.	4/22	Justin	For the enrollment datagridview, I edited the DataGridViewSelectionMode to be FullRowSelect, rather than a single cell.
4/22	Thresa	In Student UI, there is an empty row in the enrollment table. If possible, get rid of this extra row.	4/22	Justin	I believe that extra row was there because the property 'AllowUserToAddRows' on the enrollment datagrid view was set to true. This extra row provided the user with the option of directly adding an entry to the database. By setting 'AllowUserToAddRows' to false, the extra row is now gone.
4/22	Thresa	It would be nice to view the grades in the Student UI			