UML Diagram

1) Problem Statement

We are attempting to develop software that allows the end user to generate UML diagrams that appropriately and accurately model software and programs.

2) System Personnel

2.1) Users are Dr. Hutchens. He is a professor of computer science.

2.2) System developers are Eric Dougherty, Kelsey Fulton, Ryan Peterson, Matthew McAnulty, and Timothy Kettering.

3) Operational Settings

3.1) Target platforms are Windows, Mac OS X, and Linux.

3.2) The required software environment is the Java Runtime Environment 8.

4) Functional Requirements

4.1) Functional Description

4.1.1) The UML editor allows the user to create class boxes, remove class boxes, move class boxes, and create relations. Relations can be labeled, deleted, whole or dotted. The arrowheads on the ends of relations can be swapped, set to two way, and set to either composition, aggregation, generalization, or association. The editor has a horizontal file menu, vertical context menu, workspace, and scrollbars that appear when the diagram becomes too large.

4.1.2) The UML editor, which opens automatically at full screen size, will allow the user to create a UML diagram. The window will display a horizontal file menu, a vertical context menu, and a large work space. The class button is the only button initially available to the user. Clicking the button will create a class box. Once the box is created, the user can fill out the areas of the class box: class name, attributes, operations, and there is a fourth box for any other miscellaneous information. A user can then click, and drag the class box to the desired area on a display grid. The display grid becomes visible only when the user clicks to move the box. Once the user is done moving the box, the box will snap to the nearest grid location, and the grid will disappear. If the user attempts to drag the box out of the bounds of the workspace, then the box stops moving or scrollbars appear to accommodate this. All objects on the UML work space are selectable. Once a UML object is selected the toolbar buttons will change to reflect the actions available for the selected option. By selecting the class box the user can select two more options from the toolbar. They can delete the class box, and they can create relations. To delete a class box, the user must select the box and choose the delete option from the context menu. Once the user has two class boxes, they can create a relation between the class boxes using a button from the context menu. When a box is selected, the user can create a relation by clicking the relation button then clicking another box. The user can then select the relation and choose between many different

types of relations using the various image buttons in the context menu. The user can also add some text to the middle of the relation if they choose to do so. The file menu is capable of saving diagrams, opening diagrams, printing diagrams, creating a new file, and exiting the application. In the application there are also several alert and warning messages if the user attempts to exit the program without saving, create a new file without saving, attempt to create a relation by not clicking on another box, etc.

4.2) User Interface

4.2.1) The user interface includes a window when the application is opened. The window includes a file menu bar, a context menu bar, a workspace, and scrollbars if needed. Options will appear on the context menu bar once an object is selected. A class box once selected will expand to give the end user access to four editable fields; : class name, attributes, operations, and there is a fourth box for any other miscellaneous information. A relation once selected will display a field for labeling the relation by clicking on the field.

4.2.2) The menus included are a file menu bar and a context menu bar. The file menu bar contains file which contains new, print, save, save as, open and exit. The context menu bar contains buttons for add box, add relation, delete box, aggregation, composition, association, generalization, solid line, dashed line, single-ended, double-ended, and swap arrowhead.

4.2.3) The UML editor has one window that appears when you open the application. The window contains a file menu bar, a context menu bar, and a workspace.

4.3) Use Cases

Please see the Use Case document

5) Non-Functional Requirements

5.1) While important to the individual user, the data is not of great importance overall. Therefore, we chose not to emphasize reliability; however, JavaFX has proven to be very reliable as the program has not crashed during regular use.

5.2) We do not want any visible performance issues, so we took care to keep code clean and refactored it when necessary – sometimes even rewriting certain sections entirely. The program is responsive on all tested devices. The program compiles cleanly and without errors.

5.3) Emphasis was placed on usability so that even in the early stage the user could easily navigate the program, and understand what each button did. In an effort to add to the usability the program also hides features contextually so that they are only visible when they can be used. The tooltips on the buttons also help the user as they explain what the buttons do when hovered over.

5.4) We placed a high emphasis on portability so that the program was transferrable across Windows, Mac OS, and Linux through Java.

6) Future Enhancements

If we had more time, the most important future enhancement would be the ability to add multiplicity text on relations. It would also be good to give the user the ability to select and move multiple relations by clicking and dragging them to a new location in the work area. The ability to color the boxes, relations, and workspace could also be a nice feature. Text font, size, color, style, etc. is something else that could be added as another feature. Providing the option of a relation that is vertical in direction, horizontal in direction, or both but never diagonal is also something that may be nice to add. With this feature, we could possibly have some relations share the same arrowhead and all merge to a similar box. The team could also add more items to the menu bar such as Edit, Preferences, Help, etc. Each of these categories could have many of their own items such as undo/redo in edit, color changing in Preferences, and text styling in Edit for instance.