**JavaDraw GUI**

Version 1.0

**Zach Carnago**

Code/Debug Help

Documentation

Use Case Diagram

**Cody Roberson**

Code

Code Lead

**Tyler Cole**

Team Leader

Code

**Andrew Durkiewicz**

Code

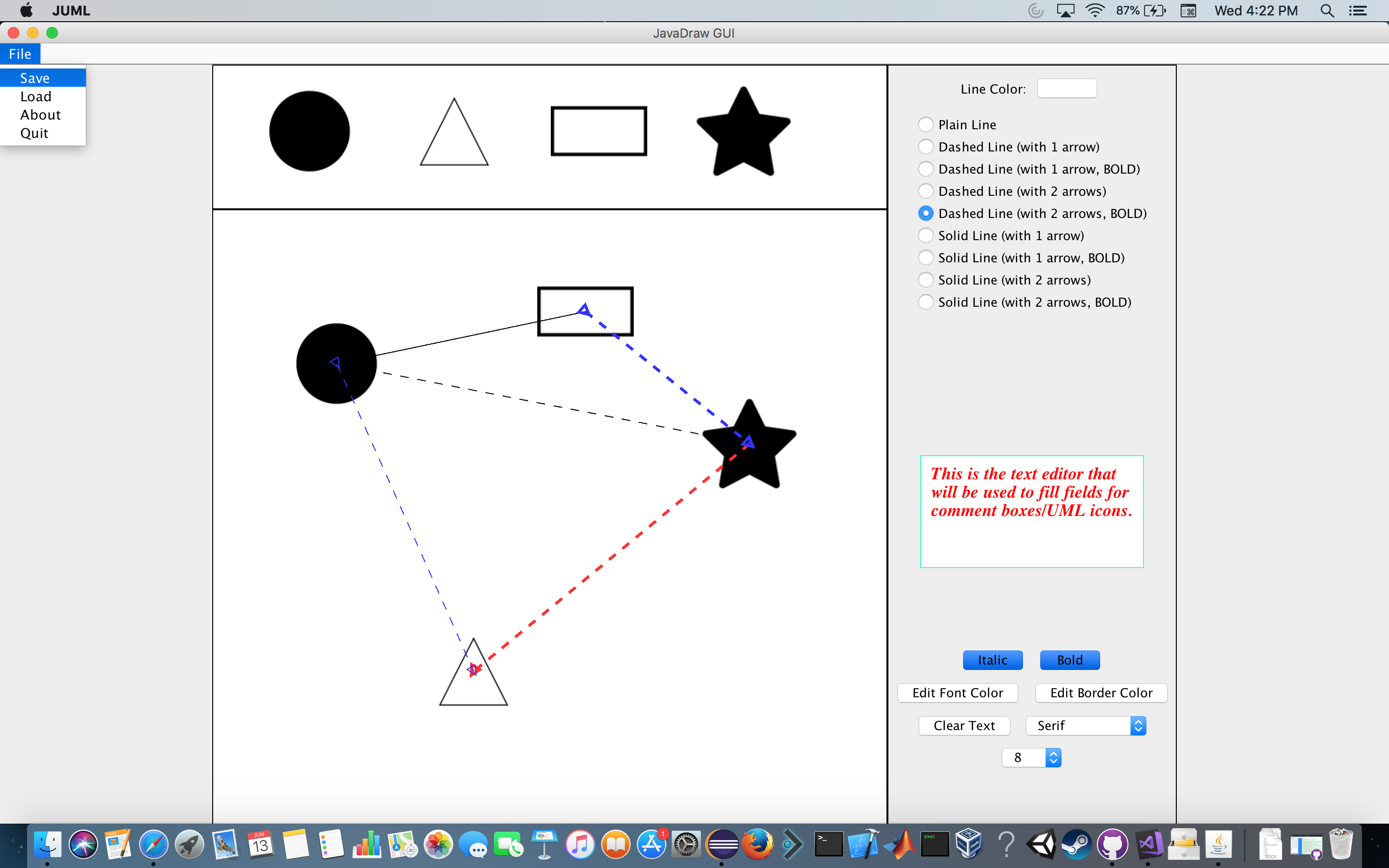
**Document Authors:**

Zach Carnago

Andrew Durkiewicz

**Customer Representative:**

Dr. Gonzalez-Sanchez



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Requirements/Document** | **Planning** | **Modeling** | **Coding** | **Delivery** |
| **Zach Carnago** | **25** | **1** | **1** | **4** | **0** |
| **Cody Roberson** | **0** | **2** | **2** | **10** | **0** |
| **Tyler Cole** | **0** | **2** | **1** | **10** | **1** |
| **Andrew Durkiewicz** | **0** | **2** | **1** | **35** | **0** |
| **Total:** | **25** | **7** | **5** | **59** | **1** |
|  |  |  |  | **Total Hours:** | **97** |

**Implementation Roles:**

x represents easier requirements

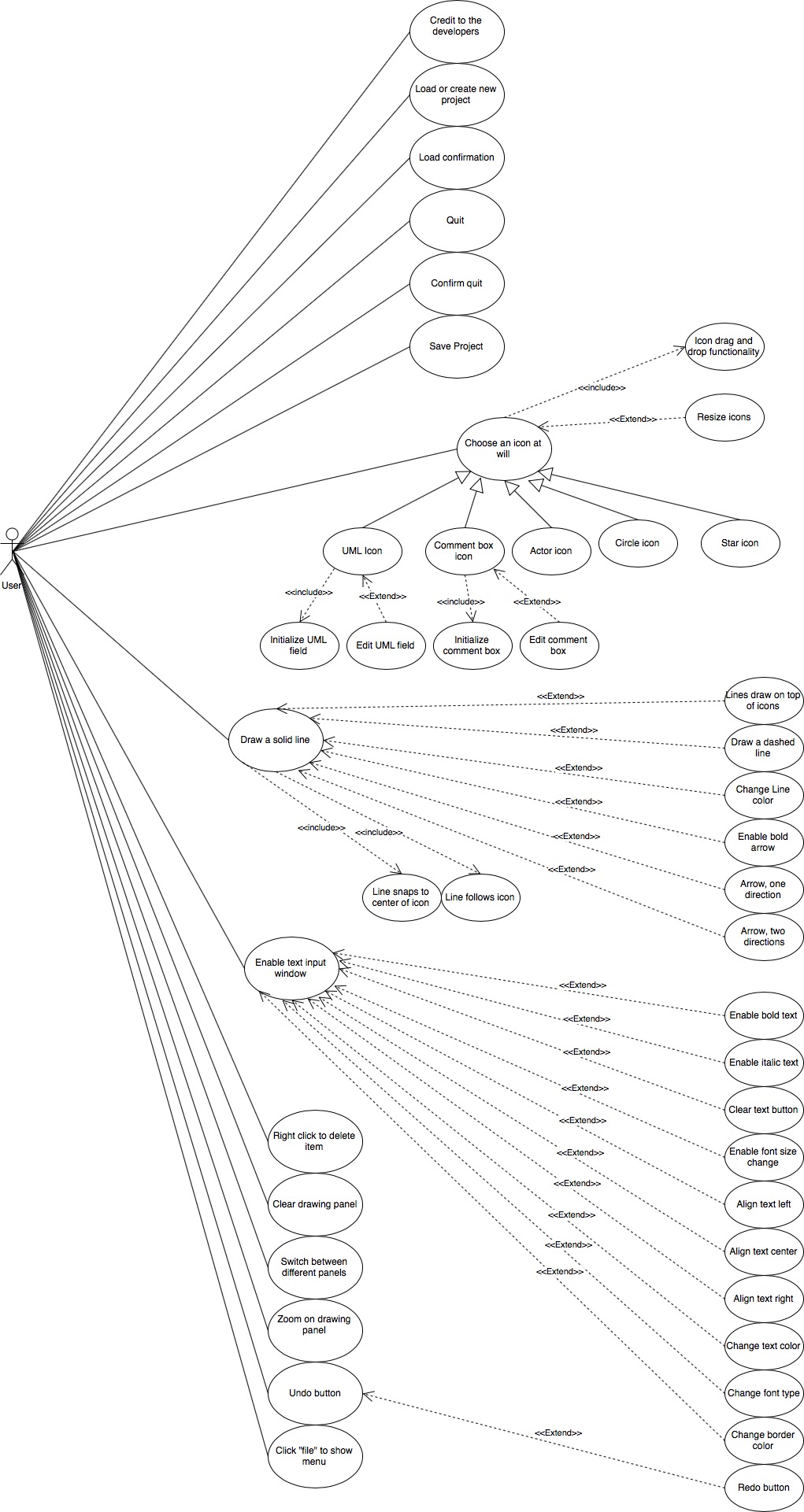
xx represents more difficult requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Implemented Requirements** | **Zach Carnago** | **Cody Roberson** | **Tyler Cole** | **Andrew Durkiewicz** |
| **Draw a solid line** | **x** |  |  |  |
| **Draw a dashed line** |  |  | **xx** |  |
| **Change line color** |  |  | **x** |  |
| **Enable bold line** |  |  | **x** |  |
| **Arrow, one direction** |  |  | **xx** |  |
| **Arrow, two directions** |  |  | **xx** |  |
| **Lines snap to center of icon** |  |  | **xx** |  |
| **Lines draw on top of icons** |  |  | **xx** |  |
| **Enable text input window** |  |  |  | **xx** |
| **Enable bold text** |  |  |  | **x** |
| **Enable italic text** |  |  |  | **x** |
| **Clear text button** |  |  |  | **xx** |
| **Enable font size change** |  |  |  | **xx** |
| **Change text color** |  |  |  | **xx** |
| **Change font type** |  |  |  | **x** |
| **Change border color** |  | **x** |  |  |
| **Load project** |  | **xx** |  |  |
| **Load confirmation** | **x** |  |  |  |
| **Save project** |  | **xx** |  |  |
| **Quit project** |  | **xx** |  |  |
| **Confirm Quit** | **x** |  |  |  |
| **Credit to the developers** |  | **x** |  |  |
| **Click "file" to show menu** | **x** |  |  |  |

**Introduction:**

The purpose of this product is to give people an exceptional organizational tool. With this tool, developers will be able to make structured and accurate UML diagrams as well as Use case diagrams.

**Use Case Diagram:**



**Use Case List:**

UC\_01: Load or create new project

UC\_02: Save project

UC\_03: Choose an icon at will

UC\_04: Circle Icon

UC\_05: Actor icon

UC\_06: Star Icon

UC\_07: Comment box icon

UC\_08: Initialize comment box

UC\_09: Edit comment box

UC\_10: UML icon

UC\_11: Initialize UML fields

UC\_12: Edit UML fields

UC\_13: Icon drag and drop functionality

UC\_14: Resize icons

UC\_15: Draw a solid line

UC\_16: Draw a dashed line

UC\_17: Enable one-direction arrow

UC\_18: Enable two-direction arrow

UC\_19: Enable bold arrow

UC\_20: Change line length

UC\_21: Change Line Color

UC\_22: Line snap to center of icon

UC\_23: Line follow icon

UC\_24: Enable text input window

UC\_25: Enable bold text

UC\_26: Enable italic text

UC\_27: Clear text button

UC\_28: Enable font size change

UC\_29: Align text left

UC\_30: Align text center

UC\_31: Align text right

UC\_32: Change text color

UC\_33: Change font type

UC\_34: Change border color

UC\_35: Right-click to delete item

UC\_36: Clear drawing panel

UC\_37: Switch between different panels

UC\_38: Zoom on drawing panel

UC\_39: Undo button

UC\_40: Redo button

UC\_41: Credit to the developers

UC\_42: Lines draw on top of icons

UC\_43: Quit project

UC\_44: Confirm quit

UC\_45: Click "file" to show menu

UC\_46: Load confirmation

**Flow Of Events**

**UC- 01**

Name: **Load**

Description:

When the user interacts with the system for the very first time, they will be displayed a blank canvas with everything available for use. They can either use the canvas given to them, or they can load an existing project by using the pulldown tab "file".

Pre-Cond:

1. System is open

Post-Cond:

1. The user chooses to load an existing project.
2. The user chooses to use the blank drawing panel.

Fail Cond: The user is unable to load an existing file.

Actors: User

**UC – 02**

Name: **Save project**

Description:

When the user is working on their project, or finish their project, they have the option to save the project under a name and in a location of their choosing.

Pre-Cond:

1. The user must have something drawn in the drawing panel.

Post-Cond:

1. The user can see a saved file in the location chosen with the name chosen

Fail Cond:

1. The user cannot find the file where it was saved
2. The file does not contain the correct saved material (corrupt file)

Actors: Users

**UC - 03**

Name: **Choose icon at will**

Description:

The user will be able to choose any icon available and drag and drop them at will.

Pre-Cond:

1. System is open

Post-Cond:

1. The icon that the user chooses to use drags and drops properly.

Fail Cond:

1. The user is unable to use one or more of the icons available.

Actors: User

**UC - 04**

Name: **Circle icon**

Description:

When the user is choosing an icon, they will be able to choose a circle.

Pre-Cond:

1. The system is open

Post-Cond:

1. The user is able to properly use the circle icon

Fail Cond:

1. The user is unable to draw a circle in the drawing panel.

Actors: User

**UC - 05**

Name: **Actor Icon**

Description:

When the user is choosing an icon, they will be able to choose an actor symbol

Pre-Cond:

1. The system is open

Post-Cond:

1. The user is able to properly use the circle icon

Fail Cond:

1. The user is unable to draw a circle in the drawing panel.

Actors: User

**UC - 06**

Name: **Star Icon**

Description:

When the user is choosing an icon, they will be able to choose a star.

Pre-Cond:

1. The system is open

Post-Cond:

1. The user is able to properly use the star icon

Fail Cond:

1. The user is unable to draw a star in the drawing panel.

Actors: User

**UC - 07**

Name: **Comment box icon**

Description:

When the user is choosing an icon, they will be able to choose a comment box.

Pre-Cond:

1. The System is open

Post-Cond:

1. The user is able to properly user the comment box icon

Fail Cond:

1. The user is unable to draw a comment box icon in the drawing panel

Actors: User

**UC - 08**

Name: **Initialize comment box**

Description:

When the comment box is dropped into the drawing panel, the user will get a prompt telling them to input the desired comment. Once finished, the user clicks enter and can see their comment.

Pre-Cond:

1. The user must be using the drag and drop functionality of the comment box icon
2. The user must drop the comment box icon onto the drawing panel
3. The user will then be prompted to input the desired comment
4. The user clicks "ok"

Post-Cond:

1. The comment box is placed into the drawing panel with the correctly formatted comment.

Fail Cond:

1. The user is unable to drag and drop a comment box
2. The user is unable to format the comment correctly
3. The user is unable to see the comment desired

Actors: User

**UC – 09**

Name: **Edit comment box**

Description:

When the user places the initialized comment box, they are able to edit it if needed by double clicking on the desired comment icon.

Pre-Cond:

1. The user will have placed a comment box somewhere in the drawing panel.
2. The user will double click on an existing comment box.
3. The user will be able to edit the text in the comment via the text editor.

Post-Cond:

1. The comment box will be edited accordingly.

Fail Cond:

1. Failure if the comment box cannot be edited.
2. The comment box fills with incorrect text.

Actors: User

**UC - 10**

Name: **UML icon**

Description:

The user will have the option to user a UML icon that can be filled with information regarding classes in programming. It will include a field for class name, variables, and methods.

Pre-Cond:

1. The system is open.

Post-Cond:

1. The user will drag the icon to the drawing panel

Fail Cond:

1. Failure if the user is unable to user this icon.

Actors: User

**UC - 11**

Name: **Initialize UML text fields**

Description:

When the user drags the icon onto the drawing panel, the drop functionality will enable them to enter text into the appropriate fields.

Pre-Cond:

1. The user drops the UML icon onto the drawing panel.
2. The user is prompted to enter the class name
3. The user is prompted to enter the class variables
4. The user is prompted to enter the class methods.

Post-Cond:

1. The Icon is visible in the drawing panel with the fields correctly filled.

Fail Cond:

1. The icon is not filled properly with desired fields.
2. The icon is not drawn on the drawing panel.

Actors: User

**UC - 12**

Name: **Edit UML text fields**

Description:

Once the user has initialized the text fields of the UML icon, they are able to edit this text again by double clicking on the desired/qualified icon.

Pre-Cond:

1. An initialized UML icon must be somewhere on the drawing panel.
2. The user will double click said icon to enable editing.
3. The user will fill the class name field
4. The user will fill the class variables field
5. The user will fill the class methods field.
6. The user will click "ok" button

Post-Cond:

1. The icon is now redrawn on the drawing panel with the updated fields

Fail Cond:

1. This only fails if the text is not edited correctly.

Actors: User

**UC – 13**

Name: **Icon drag and drop functionality**

Description:

The user will be able to drag and drop the icons from their housing panel onto the drawing panel.

Pre-Cond:

1. The system is open.

Post-Cond:

1. The user is able to drag and drop any of the icons onto the drawing panel whereer they choose.

Fail Cond:

1. Failure if the user is unable to drag and drop the icons wherever they choose.

Actors: User

**UC - 14**

Name: **Resize icons**

Description:

Once an icon of any type is placed onto the drawing panel, the user will be able to click on the icon to reveal click and drag points to be able to resize the icons.

Pre-Cond:

1. The drawing panel will have at least one icon inside of it.
2. The user will click on an icon they wish to resize.
3. The icon will reveal click and drag points.
4. The user will resize the icon.

Post-Cond:

1. The icon is either resized or left alone

Fail Cond:

1. Failure if the user attempts to resize the icons and it does not resize.

Actors: User

**UC - 15**

Name: **Draw a solid line**

Description:

The user will be able to draw a simple solid line that can be used to associate items or simply connect items.

Pre-Cond:

1. The system is open
2. The user will choose the plain line option
3. The user will choose an initial point for the line to be started.
4. The user will choose an end point for the line to finish

Post-Cond:

1. A plain line will be drawn.

Fail Cond:

1. Failure if the line does not draw properly, or at all.

Actors: User

**UC - 16**

Name: **Draw a dashed line**

Description:

The user will have the option to draw a dashed line.

Pre-Cond:

1. The system is open
2. The user will choose the dashed line option.
3. The user will choose an initial point for the line to be started.
4. The user will choose an end point for the line to finish

Post-Cond:

1. A dashed line will be drawn.

Fail Cond:

1. Failure if the line is not drawn correctly, or at all.

Actors: User

**UC – 17**

Name: **Enable one-direction arrow**

Description:

Each line type will have the option of adding an arrow to the end point of any line type.

Pre-Cond:

1. The user chooses a line type with one arrow enabled.
2. The user will draw a line

Post-Cond:

1. The arrow placed in the drawing panel is dashed.

Fail Cond:

1. The drawn line is not dashed.
2. The line is not drawn.

Actors: User

**UC – 18**

Name: **Enable two-direction arrow**

Description:

Each line type will have the option of adding two arrows, one at the start point, and one at the end point.

Pre-Cond:

1. The user chooses a line type with two arrows enabled.
2. The user will draw a line.

Post-Cond:

1. The arrow placed in the drawing panel will have two arrows

Fail Cond:

1. The drawn line does not have two arrows.
2. The line is not drawn

Actors: User

**UC – 19**

Name: **Enable bold arrow**

Description:

Each line type will have the option of enabling a bolded line.

Pre-Cond:

1. The user will choose a line type with bold enabled.
2. The user will draw a line.

Post-Cond:

1. The arrow is placed in the drawing panel with bold enabled.

Fail Cond:

1. The drawn line is not bolded.
2. The line is not drawn.

Actors: User

**UC – 20**

Name: **Change line length**

Description:

If the user places a line in the drawing panel, they can double click on the line to reveal a click a drag point on the line that will allow the line length to be extended or shortened.

Pre-Cond:

1. The drawing panel will have at least one line drawn in it.
2. The user will double click on a line to reveal a click and drag point.

Post-Cond:

1. The user will either shorten or lengthen their line.
2. Click anywhere outside the line to disable.

Fail Cond:

1. The click and drag point does not reveal when double clicked.
2. The user is unable to use the click and drag points to lengthen or shorten lines.

Actors: User

**UC – 21**

Name: Change line color

Description:

Before the user draws a line on the panel, they have the option of choosing from a wide variety of line colors.

Pre-Cond:

1. The system must be open
2. The user will select a line type
3. he user will click on the line color button.
4. The user will choose a color and click "ok"
5. The user will draw a line.

Post-Cond:

1. The line will be drawn in the correct location and will be the correct color

Fail Cond:

1. The color of the line is incorrect when drawn

Actors: User

**UC - 22**

Name: **Line snap to icon**

Description:

When the user has pressed the button to place any type of line, and the user is attaching lines to icons, they will be able to snap the line to the outside edge of the icon relative to where the line touches the icon with their mouse cursor.

Pre-Cond:

1. The user will have at least one icon present in the drawing panel.
2. The user will press a button to enable them to draw any type of line.
3. The user will snap to one of the designated points on an icon.
4. The user will click to draw the line.

Post-Cond:

1. The user has successfully drawn a line

Fail Cond:

1. The user is unable to draw a line snapped to icons.

Actors: User

**UC - 23**

Name: **Line follow icon**

Description:

The user will have a line connected to an icon somewhere on the drawing panel. The user will be allowed to move the icon and the line will follow and reformat as it moves.

Pre-Cond:

1. The drawing panel must contain a line connected to an icon.
2. The user will choose to reposition an icon by dragging and dropping again.

Post-Cond:

1. An icon will be moved and the line will move accordingly.

Fail Cond:

1. The line does not properly follow the icon

Actors: User

**UC – 24**

Name: **Enable text input window**

Description:

The user will be able to type inside of the text editor to fill comment/UML and other applicable fields.

Pre-Cond:

1. The system will be open
2. The user will be able to use the text editor for filling fields that are applicable.

Post-Cond:

1. N/A

Fail Cond:

1. The text editor window does not function properly.

Actors: User

**UC - 25**

Name: **Enable bold text**

Description:

When the user is typing in the text editor, they can toggle bold for the text.

Pre-Cond:

1. The user will have some text inside the text editor.
2. The user will toggle the bold button on.

Post-Cond:

1. The user will have bold text in the editor.

Fail Cond:

1. The text in the editor is not bold when the button is toggled.

Actors: User

**UC – 26**

Name: **Enable italic text**

Description:

When the user is typing in the text editor, they can toggle italics for the text.

Pre-Cond:

1. The user will have some text inside the text editor.
2. The user will toggle the italics button on

Post-Cond:

1. The user will have italicized text in the editor.

Fail Cond:

1. Failure if the text in the editor is not italicized after toggled on.

Actors: User

**UC – 27**

Name: **Clear text button**

Description: This button will clear all text in the text input window when pressed.

Pre-Cond: The user will have some text inside the text editor.

Post-Cond: All text inside the editor is now deleted.

Fail Cond: Failure if the text is not deleted when "clear" is clicked.

Description:

1. The user will be using the text editor.
2. The user will press the clear text button.
3. The text inside the editor is all deleted.

Actors: User

**UC – 28**

Name: **Enable font size change.**

Description:

When the user is typing text in the text editor, they can change the size of the text to 4 different sizes.

Pre-Cond:

1. The user must be using the text editor.
2. The user will choose to make the text larger or smaller.

Post-Cond:

1. The font size is changed to the desired size.

Fail Cond:

1. Failure if the size of the text is not able to be changed.

Actors: User

**UC – 29**

Name: **Align text left**

Description:

When the user is filling comment boxes or UML icons, they will have the option of aligning the text to the left side of the icon.

Pre-Cond:

1. The user will drag and drop either a UML icon or a comment box to be filled.
2. The user is prompted to use the text editor.
3. The user can choose to align text to the left once placed in the icon.

Post-Cond:

1. The text will be aligned to the left inside the desired icon.

Fail Cond:

1. Failure if the text is not aligned properly to the left side of the icon.

Actors: User

**UC – 30**

Name: **Align text center.**

Description:

When the user is filling comment boxes or UML icons, they will have the option of aligning the text to the center of the icon.

Pre-Cond:

1. The user will drag and drop either a UML icon or a comment box to be filled.
2. The user is prompted to use the text editor.
3. The user can choose to align text to the center once placed in the icon.

Post-Cond:

1. The text will be aligned to the center inside the desired icon.

Fail Cond:

1. Failure if the text is not aligned properly to the center of the icon.

Actors: User

**UC – 31**

Name: **Align text right**

Description:

When the user is filling comment boxes or UML icons, they will have the option of aligning the text to the right side of the icon.

Pre-Cond:

1. The user will drag and drop either a UML icon or a comment box to be filled.
2. The user is prompted to use the text editor.
3. The user can choose to align text to the right once placed in the icon.

Post-Cond:

1. The text will be aligned to the right inside the desired icon.

Fail Cond:

1. Failure if the text is not aligned properly to the right side of the icon.

Actors: User

**UC – 32**

Name: **Change text color**

Description:

The user will be able to choose the color of the text within the text editor panel. Can be done before or during text input.

Pre-Cond:

1. The user will be using the text editor to fill an applicable field of an icon.
2. The user will have some text in the editor.
3. The user will click on the text color button.
4. The user will select a color.
5. The user will click "ok"

Post-Cond:

1. The text in the editor will be the desired color.
2. The text in the desired icon is also the color desired.

Fail Cond:

1. The text color is not correct.

Actors: User

**UC – 33**

Name: **Change font type**

Description:

When the user is typing inside the text editor, they will have the option to change the style of font.

Pre-Cond:

1. The user is editing in the text editor.
2. The user selects the drop-down menu for font type.
3. The user chooses the desired font for their project.

Post-Cond:

1. The user is able to type in the desired font within the text editor.

Fail Cond:

1. Failure if the font style does not reflect the one chosen.

Actors: User

**UC – 34**

Name: **Change border color**

Description:

The text input window contains a solid black border that can be changed to allow for customization of your working space.

Pre-Cond:

1. The system is up and running.
2. The user selects the "border color" button in the editor panel.
3. The user selects the color desired.
4. The user clicks "ok"

Post-Cond:

1. The border of the frame will be changed to the desired color of the user.

Fail Cond:

1. Failure if the color of the border is not the same as the one selected by the user.

Actors: User

**UC - 35**

Name: **Right-click to delete item**

Description:

When the drawing panel is filled with icons and lines, the user will have the ability to delete any of the items with a right click.

Pre-Cond:

1. There must be at least one line/icon in the drawing panel.
2. The user will right click an item that they want deleted.

Post-Cond:

1. The line or icon that was right clicked is now deleted.

Fail Cond:

1. The item that is right clicked is not deleted.

Actors: User

**UC - 36**

Name: **Clear drawing panel**

Description:

The user will have the option to delete every item in the drawing panel by pressing a "clear" button.

Pre-Cond:

1. There will be at least one item, icon or line.
2. The user will click the "clear" button.

Post-Cond:

1. The drawing panel is clear.

Fail Cond:

1. The drawing panel does not clear when the button is clicked.

Actors: User

**UC - 37**

Name: **Switch between different panels**

Description:

The user will have the option to switch between 4 different drawing panels, completely independent of each other.

Pre-Cond:

1. The system is open.

Post-Cond:

1. The user is able to toggle between the 4 different drawing panels.

Fail Cond:

1. The user is unable to use the 4 panels independently of each other.

Actors: User

**UC - 38**

Name: **Zoom on drawing panel**

Description:

The user will be able to zoom in and out of their drawing panels to get a better understanding of the size and scope of the project.

Pre-Cond:

1. There must be something in the drawing panel in order to be zoomed in and out of on.

Post-Cond:

1. The user is able to successfully zoom in and out.

Fail Cond:

1. The

Actors: User

**UC - 39**

Name: **Undo button**

Description:

If the user has at least one icon/line on the drawing panel, the redo button can be pressed and the last icon/line that was placed on the drawing panel is erased.

Pre-Cond:

1. The user has placed some items inside the drawing panel, lines or icons.
2. The user clicks the undo button.

Post-Cond:

1. The last icon/line drawn is no longer in the drawing panel.

Fail Cond:

1. The icon that was last drawn is not deleted.
2. A different item other than the last drawn is deleted.

Actors: User

**UC - 40**

Name: **Redo button**

Description:

If the user chooses to undo something they will have the option to redo it.

Pre-Cond:

1. Something inside the drawing panel must be undone.
2. The user clicks the redo button.

Post-Cond:

1. The last thing deleted in the drawing panel is now back in its original position..

Fail Cond:

1. If the button is pressed and nothing is redone.

Actors: User

**UC – 41:**

Name: **Credit to the developers**

Description:

When the user clicks file in the system, they will have the option to choose "about" which will reveal all of the developer names, roles, and system functionality.

Pre-Cond:

1. The system must be open.
2. The user will click "file"
3. The user will click "about"

Post-Cond:

1. The user will be displayed all of the information stated in the description.

Fail-Cond:

1. The system does not display the "about" information.

Actors: User

**UC – 42:**

Name: **Lines Draw on top of icons**

Description:

When the user draws a line when there are icons present in the drawing panel, the lines will draw on top of the icons and when icons are dropped on top of lines, they will draw underneath the lines.

Pre-Cond:

1. The system is open
2. The user will draw a line in the drawing panel
3. The user will drag and drop an icon on top of the line

Post-Cond:

1. The line will redraw on top of the icons

Fail-Cond:

1. The lines do not draw on top of the icons when icons drop on lines

Actors: User

**UC – 43**

Name: **Quit**

Description:

The user will have the option to quit via the "file" pulldown menu. When doing so, they will be asked to save if they have not saved or quit without saving changes.

Pre-Cond:

1. The system is open
2. The user will choose the option to quit
3. The user will be prompted to save if they have not done so.
4. The user will cancel quit in order to save, or quit without saving

Post-Cond:

1. The system has been closed.

Actors: User

**UC – 44**

Name: **Confirm Quit**

Description:

When the user selects quit from the "file" menu, they will be prompted to confirm this decision just in case it was accidental.

Pre-Cond:

1. The user will select the file menu
2. The user will select quit

Post-Cond:

1. The user will be prompted to confirm their "quit" choice.

Fail-Cond:

1. N/A

**UC – 45**

Name: **Click "file" to show menu**

Description:

When the user selects "file" at any point while editing a project to be displayed a list of options to proceed.

Pre-Cond:

1. The system must be open
2. The user will click the "file" button

Post-Cond:

1. The user will be displayed a list of options once "file" is clicked

Fail-Cond:

1. Nothing is displayed when the user clicks "file".

**UC – 46**

Name: **Load confirmation**

Description:

When the user selects "load" from the "file" menu, they will be displayed a prompt to confirm their choice to load. If they continue, all progress on drawing panel will be replaced with the loaded file.

Pre-Cond:

1. The system will be open.
2. The user will select "file"
3. The user will select "load"
4. The user will be asked if they are sure they want to continue
5. The user will either cancel, or confirm the choice

Post-Cond:

1. The user will either load or continue what they were doing.

Fail-Cond:

1. The user is not displayed the warning..

**Non-Functional**

1. The program will be able to house a minimum of 30 icons within the drawing panel
2. The actor icon will have a smiley face
3. The program shall run on Java
4. The program shall be compatible with both macOS Sierra and Windows 10
5. The user should find the program to be understandable
6. The program should load within 0.5 seconds
7. The program will be a single threaded application
8. Lines connecting icons should appear with little latency
9. The program will utilize the javax.swing class
10. Use an array to store lines that are drawn
11. Use an array to store icons that are drawn
12. Undo will use Last in first out (LIFO) algorithm
13. Clear drawing panel (UC\_08) will finish in 0.5 seconds or less
14. Clear text from editor (UC\_35) will finish in 0.5 seconds
15. The default text type will be serif
16. The default text size will be 8
17. The default line color will be black
18. The default line type will be solid with no arrows on either side
19. We will use an array list to handle all data that will be saved for the user
20. The Redo functionality will use the first in first out (FIFO) algorithm.