UMLGibson, Iteration 3 (Final Version)

# Problem statement:

A UML class diagram maker with a GUI. Must have the ability to add class boxes and draw relationship arrows. Diagrams should be able to be exported.

# System Personnel:

## Description of Users:

The userbase is Dr. Hucthens as well the students who want to try it out.

## Description of System Developers:

Development team is The Slackers. We specialize in Java and JavaFX. We do not discriminate against operating systems. This is the third version of UMLGibson, the first program we are releasing as a team; experience has been achieved and is still to be procured. We are comprised of Carlos Beltre, Phil Landis, Jason Korablin, Leo London, and Joseph Asbury. We have not gained or lost any members throughout the development of this program; all have been working on it consistently through its life cycle.

# Operational Setting:

## Target Platforms:

This program will work on Macs, Linux machines, and Windows machines. It has been tested on all three operating systems with the same functionality observed for all of them. In fact, it should operate in accordance with these specifications0 on pretty much any operating system that supports Java.

## Required Software Environment:

Java.

## Useful Optional Software Environment

N/A

# Functional Requirements:

## Functional Description:

### Overview

UMLGibson is a fully-featured UML diagram editor. It can create diagrams with class boxes (each have three fields) and the standard UML relationship arrows.

### Feature List

This is the final release of this software.

* Full UI, explained later.
* Draggable toolbox to add boxes and lines.
  + Can show/hide.
  + Toolbox has full UI.
    - Class box button to insert a box.
    - Select button to drag already made classboxes.
    - Images of the objects in the toolbox with tooltip descriptions instead of text in the toolbox.
* Boxes.
  + Boxes are able to be added to the diagram and moved around.
* Menu bar, explained later.

## User Interface

Uses CSS and images for relationship arrow buttons.

### Overview

* Point-and-click.
* Standard window.

### Menus

* Standard menu bar.
  + File
    - New
    - Save
    - Exit
  + View
    - Toggle Toolbox
  + Help
    - Show Help
      * Link to text file
    - About
      * Dialog box

### Windows

* Main window.
* Toolbox window for selecting what to put on the main window (boxes and relationship arrows).

### Inspectors

N/A

## Use Cases

|  |  |
| --- | --- |
| Learn how to use the program | |
| User clicks “Help” in menu bar. | Help drop down menu appears. |
| User clicks “Help.” | Help file opens. |

|  |  |
| --- | --- |
| Reset an already made diagram | |
| User clicks class box button in toolbox. | Nothing happens. |
| User clicks on main window. | Class box is placed. |
| User clicks “File” in the menu bar. | File drop down menu appears. |
| User clicks “New.” | Canvas is cleared. |

|  |  |
| --- | --- |
| Learn who made program | |
| User clicks “Help” in menu bar. | Help drop down menu appears. |
| User clicks “About.” | About dialog box is displayed. |

|  |  |
| --- | --- |
| Draw a very simple diagram: two boxes | |
| User clicks class box button in toolbox. | Nothing happens. |
| User clicks on main window. | Class box is placed. |
| User clicks on main window elsewhere. | Class box is placed. |

|  |  |
| --- | --- |
| Draw a very simple diagram: three boxes and two aggregation arrows between them | |
| User clicks class box button in toolbox. | Nothing happens. |
| User clicks on main window. | Class box is placed. |
| User clicks on main window elsewhere. | Class box is placed. |
| User clicks on main window elsewhere. | Class box is placed. |
| User clicks on aggregation arrow in the toolbox (open diamond). | Nothing happens. |
| User clicks first box | Nothing happens. |
| User clicks second box. | Aggregation arrow is drawn between first and second box. |
| User clicks second box. | Nothing happens. |
| User clicks third box. | Aggregation arrow is drawn between second and third box. |

|  |  |
| --- | --- |
| Move an already made box (multiple trials for testing purposes) | |
| User clicks on “Select” button in toolbox. | Nothing happens. |
| User clicks and drags a class box. | Class box is moved. |
| User clicks and drags the same class box elsewhere. | Class box is moved elsewhere. |

|  |  |
| --- | --- |
| Export a the simplest diagram of one box | |
| User clicks on class box button in toolbox. | Nothing happens. |
| User clicks on main window. | Class box is placed. |
| User clicks “File” in menu bar. | File drop down menu appears. |
| User clicks “Save” in the File drop down menu. | Dialog box for selecting where to save the diagram appears. |
| User clicks “Save.” | UML diagram is saved. |

|  |  |
| --- | --- |
| Toggle toolbox off | |
| User clicks “View” in menu bar. | View drop down menu appears. |
| User clicks “Toggle Toolbox” when it is unchecked. | Toolbox closes. |

|  |  |
| --- | --- |
| Toggle toolbox on | |
| User clicks “View” in menu bar. | View drop down menu appears. |
| User clicks “Toggle Toolbox” when it is checked. | Toolbox opens. |

|  |  |
| --- | --- |
| Activate Gibson Mode | |
| User clicks |  |

## Other Interfaces

N/A

# Non-Functional Requirements

## Reliability

* Needs to avoid all forms of crashing whatsoever. Crashes are unacceptable, as the code is structured quite well and is not too complex.
* A scrollbar needs to be available to scroll along a class diagram because they can become quite extensive.
* Boxes should not disappear after drawing multiple.
* Boxes should not duplicate after moving them (this was a previous issue).
* Menu bar needs to be fully functional: every single option.
* There should be some form of error handling in the code.
  + One concrete example: A missing help file should not crash the program, but instead alert the user that there is a missing help file

## Performance

* There must be no performance issues at all, whether in terms of memory leakage during the running of the program or startup time.
  + This is reasonable, considering the size of the project.
  + Slowdowns are unacceptable and the application will be tested to make sure such is not a problem.

## Usability

* Help topics must be kept this as basic as possible because good software should not need a huge amount of documentation provided to the user.
  + The goal is that the program will be self-explanatory
* Features of the program must be easily accessible and not require many clicks. This is reasonable because there is no real need for many nested menus.
* Menu bar needs to be accessible from the top menu bar on OS X as opposed to only in the window.

## Portability

* Application must work on any system that supports Java.
  + Application must have the same functionality and utility no matter what operating system it is run on (should not be a major issue).
* Application must be a JAR file with a few other resources if any (probably only the help text file).
  + Application must have a reasonable file size.
* Saved diagrams must be in PDF format and be of a reasonable size (should not be a major issue).

# Future Enhancements

* Who knows… This is the last iteration, but we’ll listen to our users.