UMLGibson, Iteration 2

# Problem statement:

A UML class diagram maker with a GUI. Has the ability to add class boxes and draw relationship arrows.

# System Personnel:

## Description of Users:

The userbase of this iteration is probably still not be the most expansive, as it is still generally only made for one person, Dr. Hutchens. However, we, the creators of the project, The Slackers, will also probably be using it when it is completed to create a UML diagram… of itself. It is still currently in development, so The Slackers will be using it the most as of now.

## 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 second version first program we are releasing as a team; experience has been achieved and is still to be procured. No new members have been added to the team and none have been lost.

# Operational Setting:

## Target Platforms:

This program will work on Macs, Linux, and Windows machines. It has been tested on all three operating systems with the same functionality (and bugs, for that matter) observed for all of them. In fact, it should be fine on pretty much any operating system that supports Java.

## Required Software Environment:

Java.

## Useful Optional Software Environment

N/A

# Functional Requirements:

## Functional Description:

### Overview

The second iteration of UMLGibson is a more featured application. It has a function now, for the most part.

### Feature List

This is still a testing release but it does have a greater utility than the first iteration (as expected). Iteration 2 is a preview of what is to come in UMLGibson.

* Still skeleton UI
  + Toolbox look has been changed
* Draggable toolbox to add boxes and lines
  + Can show/hide
  + Actually works
  + Toolbox has better UI
    - Images of the relationship arrows 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 (described later)

## User Interface

Has a similar UI to the first iteration, but is slightly better.

### Overview

* Point-and-click
* Standard window

### Menus

* Standard menu bar
  + File
    - New (nonfunctional)
    - Save (nonfunctional)
    - Exit (works)
  + Tools
    - Toolbox (works)
  + Help (works)
    - Help topics
      * Link to text file

### Windows

* Main window
* Toolbar 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. |

|  |  |
| --- | --- |
| Toggle Toolbox Off | |
| User clicks “Tools” in menu bar. | Tools drop down menu appears. |
| User clicks “Toggle Toolbox.” | Toolbox closes because it is open by default when the program starts. |

|  |  |
| --- | --- |
| Draw a very simple diagram: two boxes and a line | |
| User clicks class box button. | Nothing happens. |
| User clicks on main window. | Class box is placed. |
| User clicks on main window elsewhere | Class box is placed. |
| User clicks on association arrow in toolbox. | Nothing happens. |
| User clicks on one box and then another. | Line is drawn between the two. |

## 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.
* Boxes need to be pushed back in the window’s area if they are moved far to the edge and the window is resized.
  + Boxes should not disappear after drawing multiple.
* Toolbar needs to be fully functional, except for New and Save.
* Toolbox must be usable and re-openable. This is crucial; it was found that it is not re-openable in iteration 1.
* Lines must be able to be drawn, even if they are not able to be deleted in this version.
  + Lines should not disappear after drawing multiple.

## 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.
* UI should not be majorly overhauled from the first iteration; it should simply look better. Features of the program should not be moved around to obscure locations or even locations far away from that of the first version.

## 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 have a reasonable file size.

# Future Enhancements

* Definite
  + Fully working menu bar.
    - New
    - Save
      * To export to an image file or the like.
  + A delete tool to essentially undo actions.
  + Better UI for the application (CSS may come into effect; if it is used in iteration 2, it is subject to change).
  + Better window size-management.
  + Gibson mode. That admittedly gave away the Easter egg.
* Probable
  + Menu also appears in OS X’s menu bar
  + Resizing of boxes
* Don’t count on it
  + Sounds