CSC 340 Computer Ethics and Software Engineering

Spring, 2018

Alex Naegle

05/11/2018

Lab Management System

Final Report

Contents

[I. Introduction 2](#_Toc513815224)

[1. Problem Statement 2](#_Toc513815225)

[2. Proposal 2](#_Toc513815226)

[II. System Description 2](#_Toc513815227)

[III. System Requirements 2](#_Toc513815228)

[1. Functional Requirements 2](#_Toc513815229)

[2. Non-functional Requirements 4](#_Toc513815230)

[IV. Use Case Diagram 4](#_Toc513815231)

[V. Class Diagram 5](#_Toc513815232)

[VI. Sequence Diagrams 6](#_Toc513815233)

[VII. State Diagram 9](#_Toc513815234)

[VIII. Activity Diagrams 10](#_Toc513815235)

[IX. Database Design 13](#_Toc513815236)

[1. ER Diagram 13](#_Toc513815237)

[2. Table Schema 13](#_Toc513815238)

[X. Conclusion 13](#_Toc513815239)

[XI. Data Dictionary 14](#_Toc513815240)

# I. Introduction

## 1. Problem Statement

It is very inconvenient for students to have to check each lab on the 4th floor of Wallace for availability, as each lab has a different class schedule and fills up at different points of the day.

## 2. Proposal

We propose a system that shall allow users to view the availability of computers inside a lab. It will include the number of computers available/busy at that moment, whether or not a class is in it right now, and the availability of the printer and projector in that room.

# II. System Description

A system to show the availability of each computer in a lab. Including the total number of computers, what type of computers they are (Mac, Windows, etc.), the total number of computers available, and the total number of computers occupied.

The system will also have a simulator to simulate a computer login or logout.

# III. System Requirements

## Functional Requirements

R1. The system shall display the status of each computer in a lab

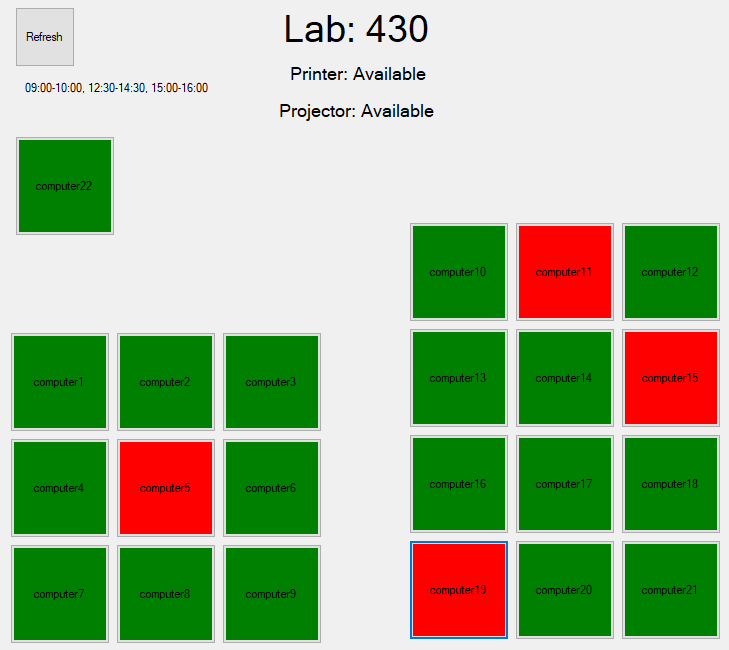


Figure 1 Screenshot of Lab Manager

* 1. The system shall retrieve the schedule of the lab from a database.
  2. The system shall display the status of the lab, including the schedule, availability of each computer, availability of printer, available of projector, and the types of computers in the lab.
     1. If there is class in the lab, the system shall display a warning message until the class is over.
  3. The system shall retrieve the status of the computers from a database.
  4. The system shall update the status of the computer in the graphic user interface (GUI).
     1. If the computer is available, a green color is assigned to the computer.
     2. If the computer is busy, a red color is assigned to the computer.

R2. The system shall be able to be refreshed

* 1. The system shall re-retrieve lab data from a database.
     1. The system shall get the schedule of the lab, the availability of each computer in the lab, and the availability of the projector and printer.
  2. The system shall update the status of the lab, computers, printer, and projector
     1. The system shall update the GUI of the ‘class in progress’ warning message, computers, printer, and projector.

R3. The system shall allow a tester to simulate the action of changing a computers state (logged in or logged out).

* 1. The tester shall be able to click on the computer GUI to change its state.
     1. If the computer is available, the status in the database gets changed to busy and the color of the GUI changes to red.
     2. If the computer is busy, the status in the database gets changed to available and the color of the GUI changes to green.

## Non-functional Requirements

NR1. The system shall allow up to 300 users at once

NR2. The system shall have a response time of at most 500ms from refresh of a lab.

NR3. The system shall update schedules weekly to remain accurate.

# IV. Use Case Diagram

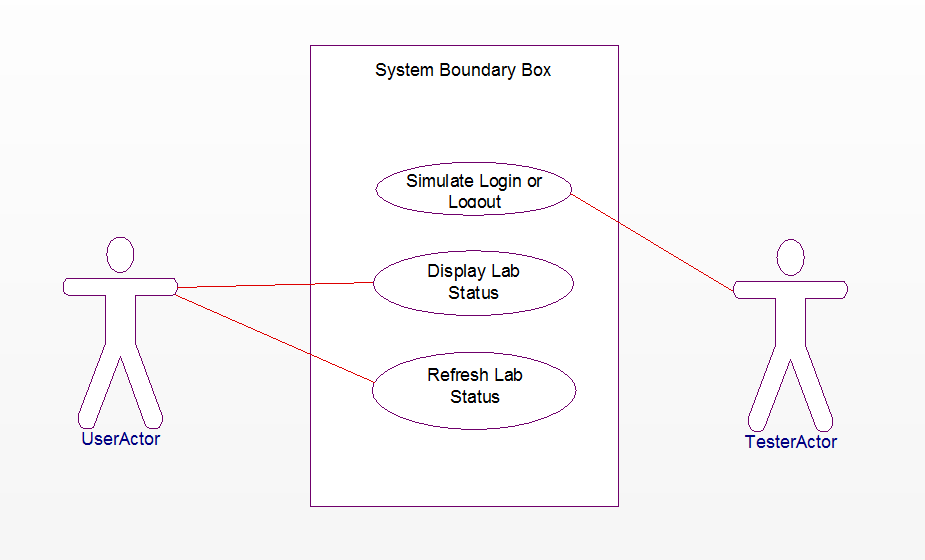


Figure 2 Use Case Diagram

UC1: Simulate Login or Logout: A TesterActor can click on a computer GUI object to change its state in the program GUI and in the database. (Refer to R3)

UC2: Display Lab Status: When first opening the program the lab status will be displayed. (Refer to R1)

UC3: Refresh Lab Status: A UserActor can click on the refresh button to update the entire lab status. (Refer to R2)

# V. Class Diagram

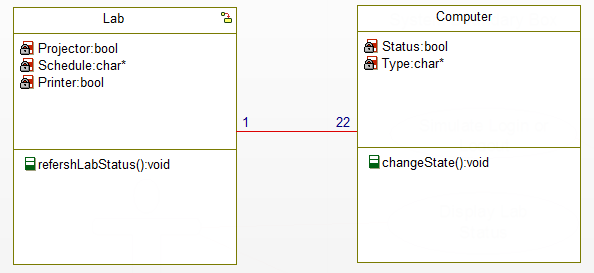


Figure 3 Class Diagram

Lab: Contains the status for the projector and printer, and the schedule for the lab.

refreshLabStatus() updates all the attributes of the lab from a database.

Computer: Contains the status for each computer inside the lab.

changeState() changes the computer from available to busy, or from busy to available

Many computer classes are associated with the lab class. This is so when the lab is updated it can also update the computers.

# VI. Sequence Diagrams

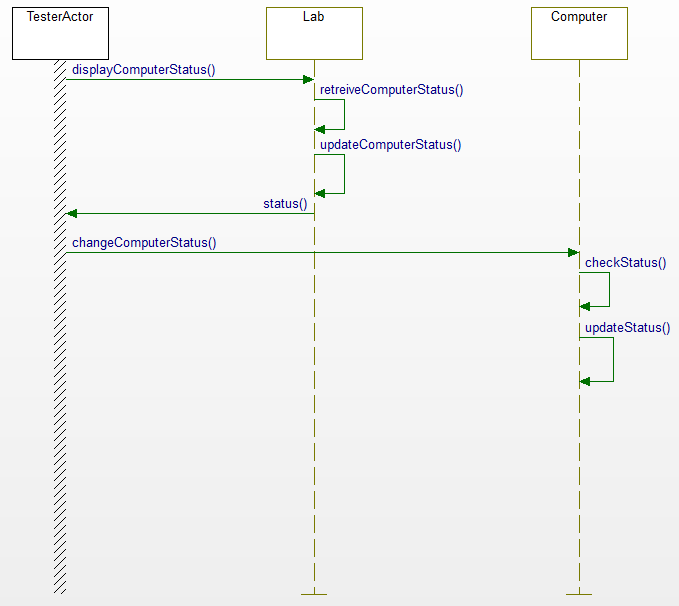


Figure 4 Sequence Diagram for Simulate Login or Logout

The computers status will get changed based on what it is currently is. Then that will be updated in the lab GUI, then displayed to the user.

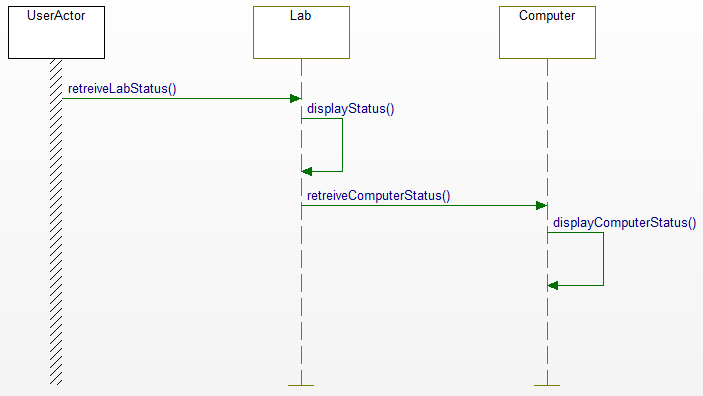


Figure 5 Sequence Diagram for Display Lab Status and Refresh Lab Status

To display the lab status when the program first loads we need to get and display the statuses of the computers. Then we will get and display the status of the lab and its attributes.

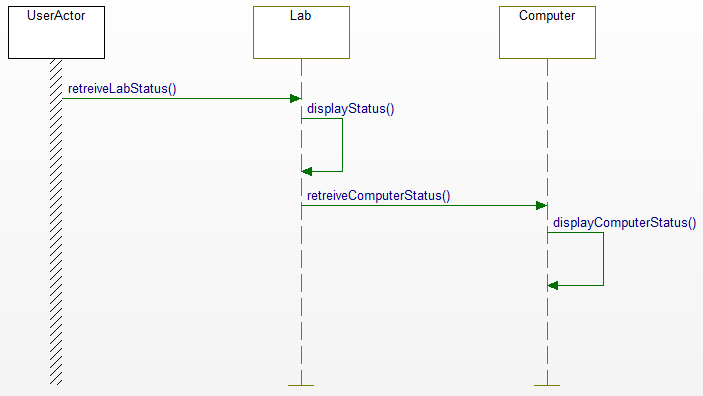


Figure 6 Sequence Diagram for Display Lab Status and Refresh Lab Status

When the refresh button is clicked we will re-retrieve the lab status, display them. Then re-retrieve and display each computers status.

# VII. State Diagram

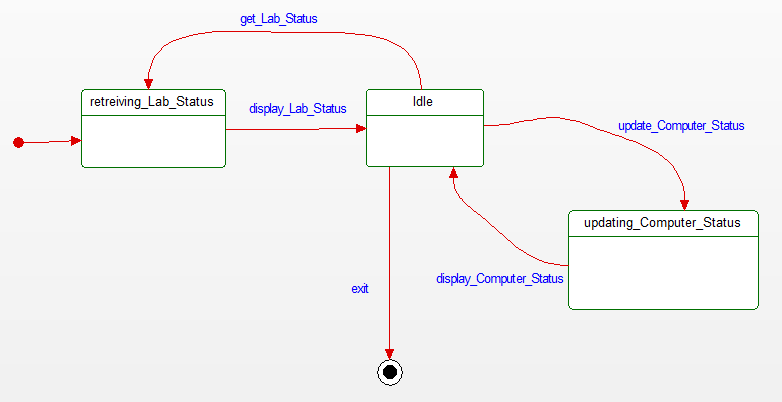


Figure 6 State Diagram

There are 3 states, 2 are waiting to receive data from the database and the other is idle. From idle you can update a computers state, refresh the lab, or exit the program.

# VIII. Activity Diagrams

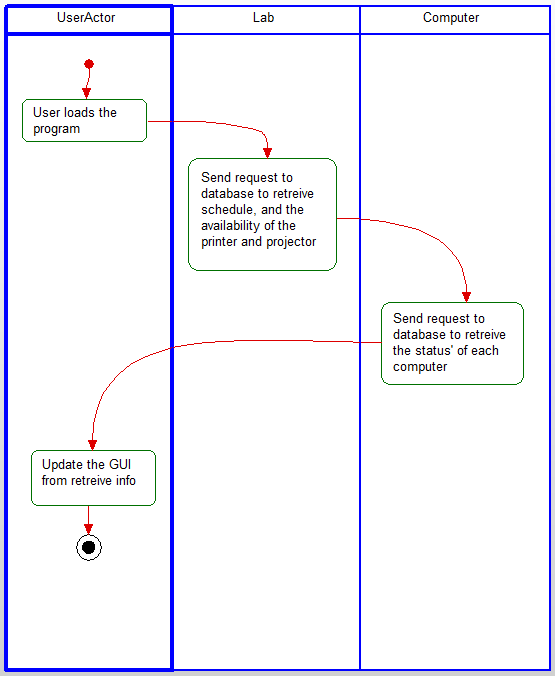


Figure 7 Activity Diagram for Display Lab Status

When the user loads the program, we need to get all the data to initially display to them.

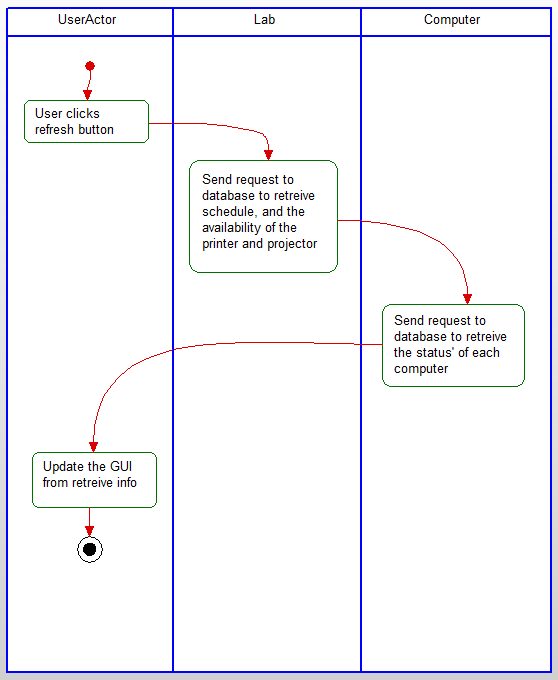


Figure 8 Activity Diagram for Refresh Lab Status

When the user hits the refresh button, we need to re-retrieve all the data from the database and then update the GUI.

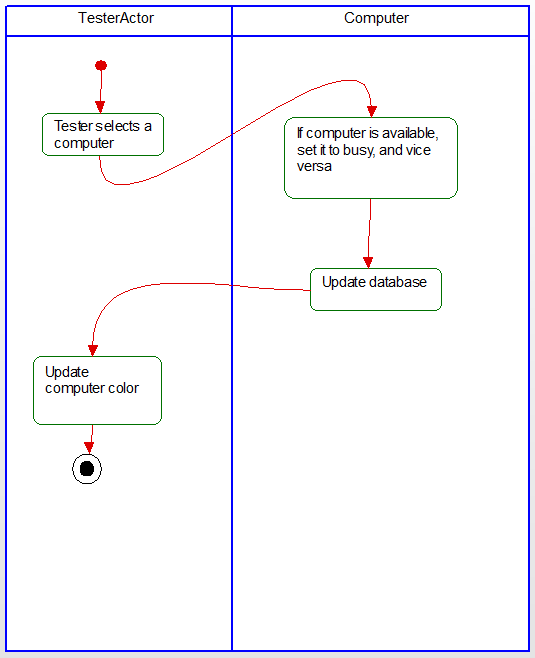


Figure 9 Activity Diagram for Simulate Login/Logout

Whenever a tester clicks on a computer, we check what the computer is, then update the database to be the other, then update the GUI.

# IX. Database Design

## ER Diagram



## Table Schema



Lab and computer are connected by their room number. Printer and projector are either available or not. The status of the computer is either busy or available, and type is PC, Mac, or other.

# X. Conclusion

Analysis and design for a system that would allow the user to check the availability of computers in the labs of the 4th floor of Wallace in real time. Also designed a simulator to simulate users logging in and logging out to allow for troubleshooting.

# XI. Data Dictionary

1. Computer: The class that stores the state of each computer in a lab.
2. Lab: The class the stores the state of the projector, printer, and is connected to the computer class.
3. Printer: A Boolean variable determining if the printer is available or not.
4. Projector: A Boolean variable determining if the projector is available or not.
5. Schedule: A string of all the times that classes are in the lab.
6. Status: A Boolean variable determining the whether the computer is available or busy.
7. TesterActor: An actor in the case diagram. It represents the tester who will use a simulator to perform login and logout actions in the system
8. Type: A string variable that shows what type of computer it is. i.e. PC, Mac, etc.
9. UserActor: An actor in the use case diagram. It represents the user of the system.