**Team QQ Technical Design Document**

**Cody Malone, Corey Massey, Kelsey St. Clair**

Executive Summary

Imagine bringing an object of fantasy and mysticism into this world and making it a reality. If the objects you hear about in fantasy or science fiction stories could be achievable through modern technology and properly integrated in a current setting, would that not be an accomplishment toward which to strive? Introducing the Neumont Marauder’s Map. Our goal with the NU-rauder’s Map is to capture the mysticism of everyone’s favorite fantasy series (Harry Potter) while making it practical in a modern environment at the same time. Never again will anyone who attends this university have a question as to the whereabouts of their instructors, faculty members, or other students. Everyone’s location coinciding with their schedule will be displayed on the NU-rauder’s Map for all to see.

While we do not have the legal right to track student or faculty members movements, our map will still function through information which is already public university knowledge. Based on everyone’s schedule for the quarter, their virtual figure will move around the building accordingly. As these icons move throughout their day weaving through the corridors of Neumont, the user will be able to interact with the icons to locate those individuals amongst the crowds. If the user decides to search for a particular individual, then the person in question’s icon will transform into a new unique icon which advertises their presence amongst the chaos of the school. Imagine there is a student using this program. He notices an icon representing a professor move throughout his day, and the user who is viewing this wants to know where he can locate said instructor when he is not preoccupied with classes. Through the process of searching for the sought after teacher or adjusting the time to a later moment, the user will determine his location and discover when the instructor has free time and then his need to obtain this information elsewhere will be satiated.

An interactive map keeping track of student and faculty schedules along with a plethora of useful, ease-of-access information about said individuals is a truly monumental feat that few would attempt. The potential uses for this program are numerous and would only serve to aid in productivity and management for the university. While the task set before our team may be monumental, our strength and resolve will see it to completion and implementation. Through our unwavering resolve and dedication, we will construct this piece of software which will change the way establishment are able to keep track of and manage the time of their constituents. The NU-rauder’s map will contain something of value to all parties who seek to use it and will represent the future that technology has the capability to build.

Special Terms/Definitions

Marauder’s Map – a magical map used in the Harry Potter franchise that displays the location of every person at any time in Harry’s school

NU-rauder’s Map – based off the Marauder’s Map, it is our teams attempt to make a technological savvy version of the magical map that will display the location of everyone in our school at any time

Teacher – a person who has a permanent job at the university, has office hours, teaches classes, and spends the entirety of their day at the school

Adjunct – a person who teaches classes on the side and leaves the building in-between classes to attend their fulltime occupation

Faculty – a person who works at the school fulltime but does not teach classes

Requirements Map

Once the NU-rauder’s Map is complete, the user will be able to influence and interact with key elements of the map. The following interactions have already been established by our group as those the user will be able to directly control:

* The user will be able to create a person to place on the map, decide if that person is a teacher, student, adjunct, or faculty member, and enter information about that person for later use.

When the user first creates a person, the user will navigate the menu to find the appropriate JMenuItem for creating a new person. Once clicked, a new JOptionPane will open in which there will be text input fields. The user will then be prompted to enter information about the new person, specifically their name, status at school (teacher, student…etc.), how long they have been at the university, and their class schedules. The classes to choose will be displayed in JCheckBoxes of which the user will select the appropriate ones for the new person. The new person object is then sent to the controller where it adds a new person to the person library and saves the updated list.

* The user will be able to delete a person they previously created.

In order to delete a person from the list, the user will open the menu and locate the appropriate JMenuItem for deleting an already existing person. Upon choosing this option, a JOptionPane will open which displays a current list of active person objects and prompts the user to enter a name for removal. The chosen person object is then passed to the controller which removes the object from the person library and resaves the list for updated use.

* The user will be able to view different floors on the map at any time.

If the user wished to view different floors on the map, then he would look toward a feature located in the main user interface. Next to the active map, there will be two JButtons which will allow the user to either view a higher or lower floor until the limit is reached on both ends. Once the appropriate button is clicked, the current floor is removed from the user map and the new floor is redrawn by the UI. The exception to this process would be if the user’s map was already viewing the top or bottom floor. If the user attempted to view a higher floor than the top floor, then the program would be unable to complete said action and the floor would remain the same.

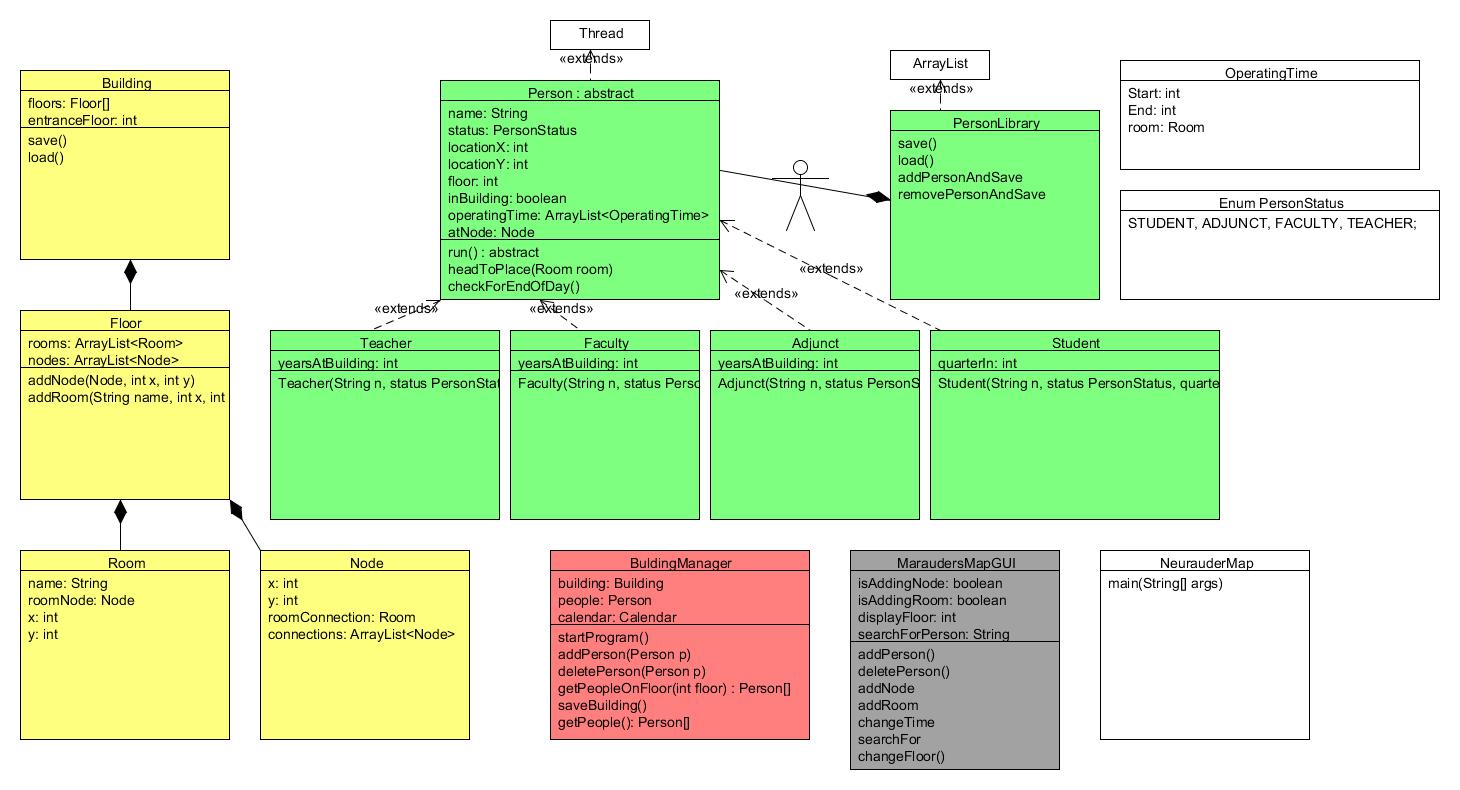
* The user will be able to change the current time at any point and see the icons rearrange to their appropriate locations.

If the user wished to adjust the current time of the map, there would be a clock located on the left side of the map. The clock is contained within a JTextField that is open to editing from the user. Once the time is adjusted, it is passed into a method that will update the person object positions on the map. The icons will then warp to their set positions for that time. The area to which they are warped if at all will be determined by the amount of time skipped. If time is only incremented by a minute or two, then the icon’s positions will most likely not adjust at all. Their positions will be based entirely around their set schedules.

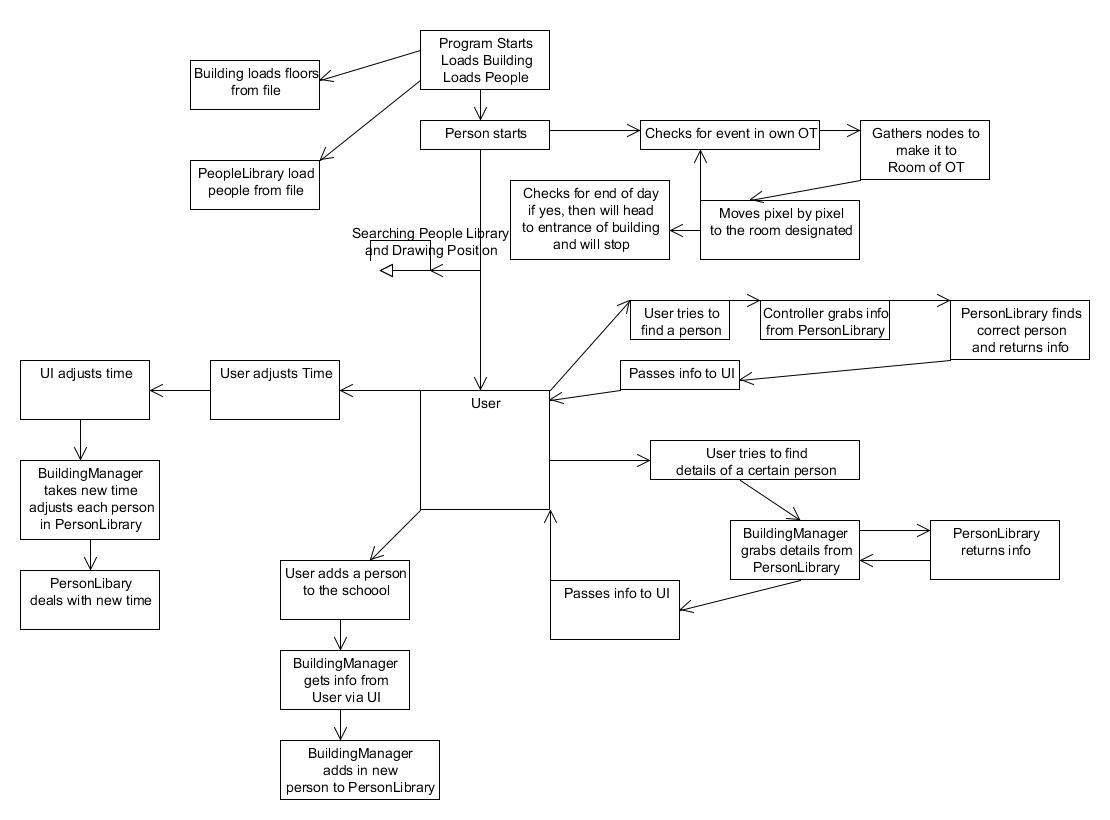
* The user will be able to search for a person on the map at any time.

If the user desired to search for a specific person on the map, he would utilize the onscreen JButton for the act. On the UI, there will be a labeled search button which upon being clicked will open a JOptionPane for user access. The user will be prompted to enter a person’s name. If the system can determine that said person exists, then the person will be located. The current floor of the map will be adjusted to that person’s current floor and the color of his icon will change to signal to the user that the unique icon is the person for which he was searching. Alongside this, an arrow might display above the icon to better signal to the user the location of their sought after person.

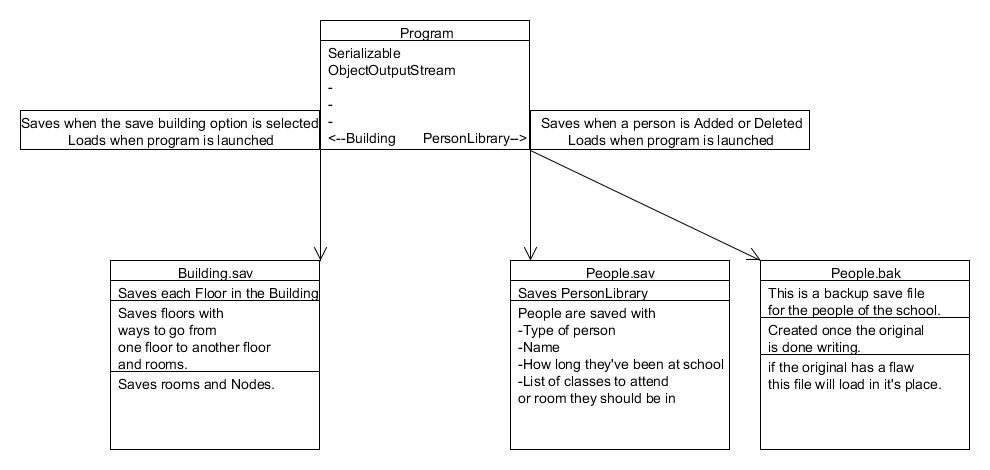
UML Diagram

****

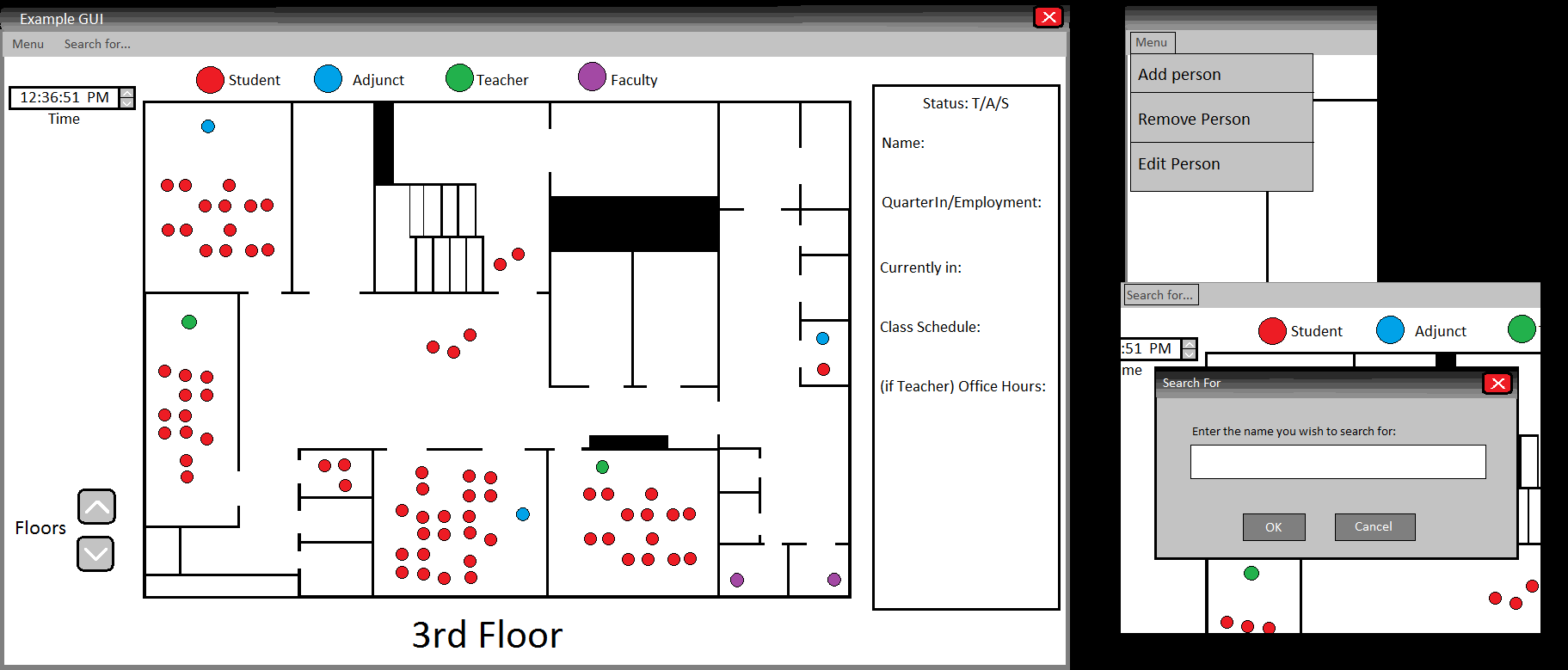
Process Flow Diagram



Database/Storage Diagram



GUI Screen Shot



Test Cases

Program runs without breaking.

All images load in the window.

Up button

* Changes floor window image and label
* Changes node map according to floor
* Floors go up until floor 4 without failing and without going past that floor

Down button

* Also changes floor window image and label
* Changes node map according to floor
* Floors go down until floor 0 without breaking or going past that floor

Menu button

* Opens up to display Add and Delete menu items

Add person menu item

* Opens new window without failing
* Window asks for name and the status of the person and classes or office hour times
* Check boxes asking for statuses select only one and react accordingly
* Invalid input does not break the program
  + Entering numbers, symbols, or long strings do not create errors

Delete person menu item

* Opens a new window without failing
* Window shows a list of the names of people already in system
* Type name of person you wish to delete
* Does not crash if input is invalid
  + Entering numbers, symbols, or long strings do not create errors
  + Returns error if person does not exist

Search for person menu

* Opens a new window
* Asks for persons name to locate
  + Accepts numbers or symbols without breaking
  + Returns appropriate error if person is not located or does not exist

Clock text field

* Accepts strings, symbols, or long integers
  + Returns an error to the user if invalid input is entered