COMPUTER SCIENCES AND SOFTWARE ENGINEERING

AUBURN UNVERSITY

COMP 2710

**Software Construction**

Fall 2014

**Lab 3**

**Distribution Messaging System**

**Due: November 3rd, 2014**

**Analysis**

Use Cases:

1. **Read in a graph configuration file:** The program will prompt the user for a graph configuration file. If it cannot be found, it will be re-prompted.
2. **Quit the program:** The user can enter “Quit” to quit the program.

**Design**

* **Classes**
* **System Menu –** This class will be the main. It handles user decisions for system use and calls to the other classes.
  + **Functions**
    - runProgram – Presents the user with the menu. Is the gateway function to all other classes’ functions and variables. If case is to quit, system shuts down.
    - Sleep – allows the system to pause momentarily
  + **Variables**
    - None
* **Node -** a pointer variable that points to another Node class object
  + **Constructors**
    - Node(newname) – node with a new name in
    - Node() – no param
  + **Functions**
    - setNodeName – gives the node a name
    - getNodeName – gets the node’s name
    - attachNewNode – connect node to a new node
    - getAttachedNode – getting attached node at its address
    - getNumberUsed – how many nodes used of capacity 4
    - setNumberUsed – set number of used up to 4
    - setPrevious – changed the previousNode, one that is already attached
    - getPrevious – acquire the previousNode
    - setMarked – set what has been visited
  + **Variables**
    - Name – the name of a node
    - attachedNodes[4] – the name of nodes that are attached, cap = 4
    - numberUsed – is the current number of nodes used
    - previous – the node that added this current node
    - marked – the status of whether this node has been marked
* **MazeSolver** – This class is in charge of maintaining the maze and solving it
  + **Functions**
    - getFile – Gets the current file from the user.
    - buildArray – Builds nodeArray from given input file.
    - getMazePath – Gets the startNode and destinationNode variables from the configuration
    - traverseMaze – Displays the shortest path between the parameter start and end nodes. This is the maze solving function.
  + **Variables**
    - startNode – the starting position.
    - destinationNode – the desired final position.
    - numOfNodes -Number of nodes in the configuration file.
    - nodeArray – array that will store the current graph’s nodes.
    - graphConfigFile – Current graph configuration file.
    - path – that will maintain the shortest path.
    - marked – will log all nodes visited/marked.

**Class Diagram**



**Data Flow Diagram**

**Testing**

1. **System Welcome:**
   1. Check if the initial system welcome is printed correctly.
2. **Read in file:**
   1. Enter a valid file and ensure that information is read in correctly.
      1. Nodes must be connected via pointers.
         1. Check for proper pointer usage.
   2. Enter an invalid file, and re-prompt user after error message.
   3. Ensure that start and end nodes are received correctly.
      1. These two nodes must both exist.
      2. These two nodes must be connected.
   4. Traverse map
      1. Ensure that all visited nodes a properly marked and recorded.
      2. Ensure that the shortest path amongst these nodes is recorded.
      3. Ensure that both bits of information is correctly printed for the user.
   5. Read in multiple files
      1. Ensure that the program can run correctly multiple times and printed correctly.
3. **Quit the System:** 
   1. Check if the program posts a thank you message and then exits.