

Ernesto Martinez  
CS 202  
February 17, 2017

## Program 3 Design

My UML currently have eight classes. It may change after I start coding but it will really depends on how the client program will be affect it. I will be using one abstract top class that will force the derived classes to be implemented. I am using this method because the derived classes which will be the communication services will have the same methods will very small differences in behavior between them.

The most important class in the program will be my Manager class which will have an objects of all the derived class which are refering to the communication services. I am not going to be using dynamic binding in this assignment, therefore upcasting operations will not take place.

Now, the abstract base class will be the System class, referring to the hub of the broadcasting system. It will derive three classes which will be the communication services. Such as the Facebook class, Radio Class, and the Phone Class. This abstract base class will have virtual pure functions with behaviors of a communication services such as the availability of the service ( for example site may be down sometimes, the radio may not be working 24/7 or the phone may be unlocked by the company at a given time. ) , the speed of reaching the person ( facebook user may be logout during that time, or phone may be in mute, but the radio as long it is on all the time, will reach the person quickly.) , and also the accuracy of the information ( sometimes facebook pages display fake information ). The derived classes will be forced to implement these functions and they may have different characteristics each one.

The data structure used in this program will be composed by four classes. The Binary Search Tree class which will be top of the structure, will manage the nodes composition and will sort the names by its last name. For example, a last name per each node, and when the BST display, it will display the last names in order. The Binary Search Tree node class will manage the info inside the node, it will contain an object of type Linear Linked

List inside. Each node in the tree will have a list inside with the first names in order, these first names will be vinculated to a same last name. For example, a node can have all the first names, but what really defines the node is the last name. Let's suppose that we have a node with a char \* array and that array of chars have the last name "Martinez" inside, well, the node will contain the object of the list, and that list will have all the first name of the people with the "Martinez" last name. The sorting operation will be key part of the binary search tree class to order to last name, so the information is well distributed alphabetically.

The Linear Linked List class will manage the first names of a binary search tree node. It will contain the nodes of the list. It also will manage the operations of the list such as adding, deleting, and most importantly, sorting which is key part of the program design. While the linear linked list node class will contain an array of characters to store the first name and will manage the operations conducted to the edition of the name. For example, my name, will be in the node of the binary search tree corresponding to "Martinez" and that node will have a linear linked list, where one of the nodes of the list will be containing an array of characters with the word "Ernesto".

Each communication service class, will have an object of type Binary search tree class. Therefore each communication service will manage their own people depending on what service those people are using.