Assignment 1 : Design

26th January 2018

Winter 2018

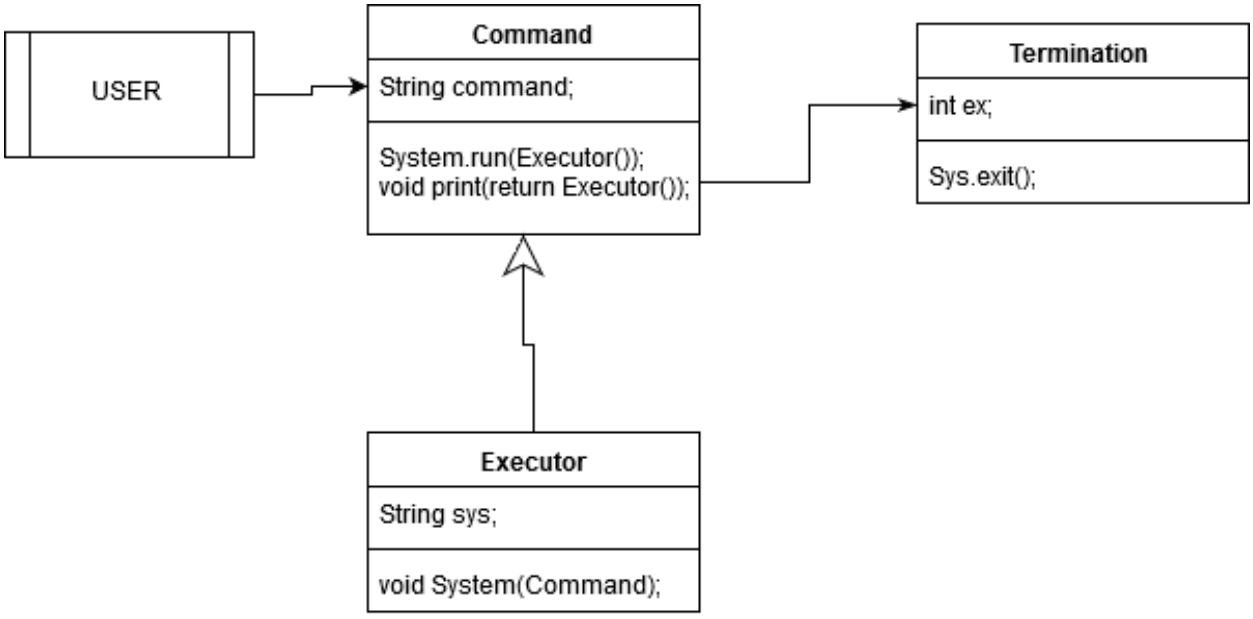
Sean Shuai, Tanish Arora

SID: 861102011, 8620122324

Introduction Design:

Our shell program is designed in such a way that when it starts up, it will read the information from the source code and execute the configuration files for the shell. It will then read the entered command by the user in the shell program and then in order, it will execute the command based on the user's interests. After the user is done with the code/commands, the user will be able to terminate and exit the shell program.

UML Diagram:



Classes:

Command

This class is going to be designed in such a way that it:

- Reads in the commands from the user
- Run them
- Reading in the command will use the system built function such as stdin(), which takes in the line entered by the user and then parses it to separate the argument from the command line itself.

Executor:

This class is going to be designed in such a way that:

- It will take in the parsed command and the argument from the ***Command*** class
- Execute those commands using System call
- Returns the output to command class to be outputted out to the user on the terminal.

Termination:

This class is going to be designed in such a way that:

- After the user inputs the end command, it will end the infinite loop and completely terminate program

Coding Strategy:

We plan to split the work in manageable chunks. Since we believe that the Command class will take the most work, we will have only one person do this. The other person will take care of the Termination and Executor since those will be less time-consuming. Sean will take care of the Command class while Tanish will do the Termination and Executor. We will meet up weekly to compare code to see if they are compatible. This should prevent too much trouble when integrating the code together to create the program.

Roadblocks:

There are several roadblocks that we are considering when creating this program. For example, we have to account for if the user will input many lines into the command. This might cause an error, so we have to be sure to code a way around this issue. The Command class might be a hard class to code due to the sheer amount of data it will handle. Also have to account for the fact if the user never exits the program. Maybe something that will prompt the user to exit after he is done with every command. This will serve as a reminder.