Final Project Proposal

Kefil Tonouewa

December 3, 2018

Project Summary

The proposed project title is TIMER.

This Java application will help its user in the process of checking in and checking out of their workplace. In other words, this application will be used to keep track of duration based on the time you login and the time you logout.

The intended user of the application is a company the wages its workers based on their hourly performance. This application could also be used by anyone who wants to track the time he/she spent accomplishing a task.

The problem that this project is trying to solve is keeping track of time data digitally rather than manually.

The technology that I will be using to fulfil this project are GUI, array and array list. GUI will be used as a tool to interact with the users and the remaining will be used as a storage for the data that have been passed through the program.

1. Use Case Analysis

Type of User: Employee of the company

First, the employee will be Welcome to his/her job with a message followed by the program prompting him to enter his work ID.

Whenever the worker comes back to clock out, the same process will repeat and at the end a message box will pop up to

Type of User: Employer of the company

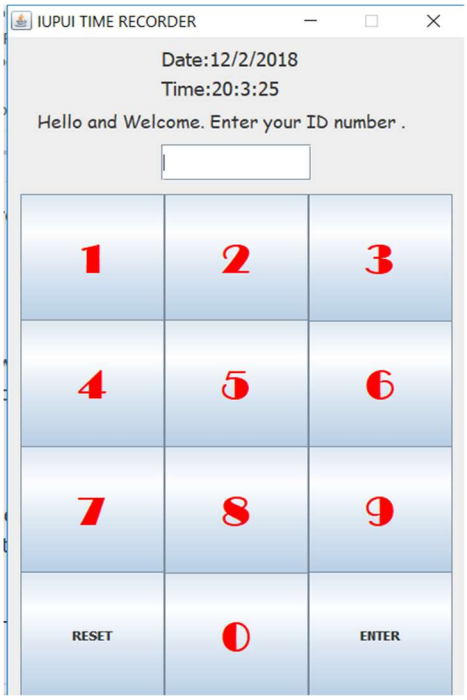
He should be able to access the file that will compute the hours that his/her workers have each done. The file name that hold those data has been name hour.txt

2. Data Design

In this project the key data will be time in, the time out and the Hour accumulated by each employee. Those data are key because in other for the duration of the employee at work to be determine, we need the time he went out minus the time he got in. Also, the Hour accumulated is a key data because without that, the worker will not be waged properly. In order to recode the time, we will be using the GregorianCalendar () and then formatting the time like this: hh:mm:ss.

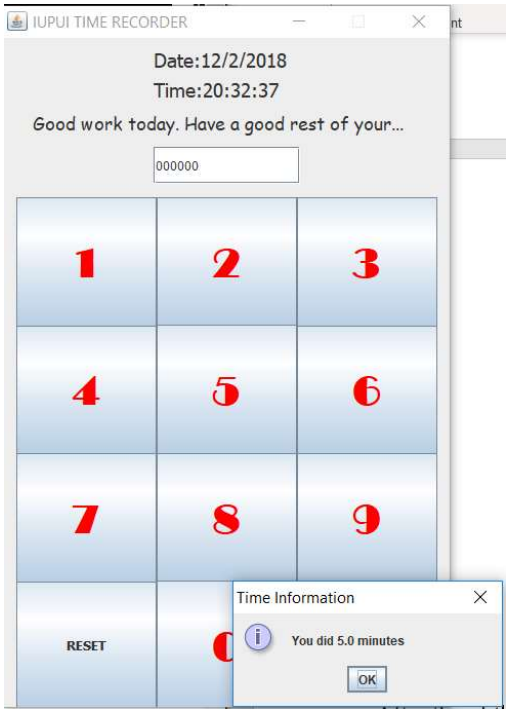
3. UI Design

When the program is first run, the GUI will look like this:



So at that point the employee will enter the ID number that has been assigned to him for this project I set up the ID as the following: (Andy: 000000, Kefil: 111111, Paul: 222222, Prasad: 333333, Jake: 444444, Gabe: 555555). Right after one of this is entered and the enter button get clicked, the GUI will look like the following:

After this if the same ID is entered again at anytime, the employee will be logout and the GUI will look like this:



Button (0-9): Used to input the ID

RESET: Used to reset the GUI back to the main screen

Enter: User to validate the user input

4. Algorithm

The first step of accomplishing our goal is to create a class of employee. After that, we will need to create a constructor of the employee class as well as some getter and setter in other to access and modify the element of the Employee class.

This is the list of the variables that the employee class will contain:

- String Name
- String ID
- Double HOURS

Within the Employee class, we will have some methods (Getters and Setters) to make the elements of the class accessible outside of the class since those will be private. Therefore, we will have:

- A method to get the Name of the employee (getName ()). This will return the name of the employee whenever it needs to be access.
- A method to get the ID of the employee (getID ()). This will return the ID of the employee whenever it needs to be access.
- A method to get the hour accumulated by the employee (getHOURS ()). This will return the hours accumulated by the employee whenever it needs to be access.
- A method to set the hour data of the employee (). This will set the hour accumulated by the worker by first getting the actual time accumulated by the employee and adding to it the new hour done by the user.

The second step is to create is a class called TIMER. This is the main class of our project. In this class we are going to set up our GUI and implement our login and log out mechanism.

- For our login and logout mechanism, this is how we will proceed.
    1. First, we need the list of the key variables that are being used:
        - An array of our preconstructed Employee class (Employee [] e = new Employee [6]).
        - An arrayList that will contain the ID of the employee that are already at work (Those who already login) (ArrayList < String> Employee_working).
    2. Since we already know the key data of the mechanism, we will set up the program such that we go through the arrayList containing the ID of the worker who already login. If a match of ID is found, we will display the logout message and delete that specific entered ID from the list of the employee that are already at work. In the opposite scenario, we will go through the array containing the list of people that there. Whenever a match is found, a we will display the welcome message and add the ID of that specific worker to the Arraylist of those who are login so that when the same ID is entered again there are able to be logout from the system.