



## SCREEN IDLE TIME CALCULATION

Rakshitha Hegde<sup>1</sup>, Ananth Murthy<sup>2</sup>

Department of Master of Computer Application, Visvesvaraya Technological University, Belagavi

### Abstract

The "Screen idle time calculation" project will calculate the idle time that the user spent on their screen during working hours.

In an organization, accurate analyses of idle time is crucial for the efficient utilization in large project. The less idle time, the higher productivity the construction can achieve. However, it is not practical to measure idle time by human visual observation.

### Introduction:

In an organization, idle time variance the adverse impact on the profitability of an organization as a result of having paid for the labor time which did not result in any production. The separate calculation of idle time variance ensures a more meaningful analysis of the underlying productivity of the workforce demonstrated in the labor efficiency variance. In school/colleges student idle time can be tracked. As with the labor efficiency variance, the calculation of idle time variance based on the standard rate since the variance between actual and standard labor rate is separately accounted for in the labor rate variance.

The "Screen Idle Time Calculation" tracks the idle time spent by the employee or a student. It increases the company productivity. Manager of the company can analyze the working of the labor in the organization. This can be tracked based on the cursor location, If it sits idle for more than specified grace period, the idle time is saved into the database.

### Literature Survey:

Earlier one can listen for mouse and keyboard event, but only from a frame that has to be focused (for keyboard event) and / or that fills the whole screen (for mouse events). And the Mac OS X implementation of java does not even

seem to trigger any mouse motion event to a non-focused window.

In the native world, though, there are plenty of ways to get the job done, will not give perfect solution:

On "modern" Windows version (Windows 2000 and more recent), there is the `GetLastInputInfo` method in `user32.dll` that just takes care of the whole thing (gives time of the last mouse or keyboard activity)

On older Windows version, there are some obscure way to register event hooks in the system.

In X11, it is easy to query the mouse position (using `XQueryPointer` on an artificial empty window). The "Screen Idle Time Calculation" application automates the tracking of the working hour out of the java platform without the `dll` file or native access.

### Implementation:

#### Login:

The Login Form module presents a form with username and password fields. If the user enters a valid username/password combination they will enter into the system. If admin wants to login to the system enters a valid username / password combination, then he enters the admin Registration page. Once the user is logged in, the idle time tracking process will start.

#### User Registration:

A user module is a collection and categorization of personal data associated with a specific user. It can include personal information such as users' names, id, phone number and E-mail.

#### Admin Registration:

This module is a collection and categorization of personal data associated with a specific admin. It can include personal information such as admin'

names and id, their phone number, E-mail, in that form itself specify the password to the admin.

**View Details:**

In this module the admin can view the details of the particular user. The details such as user id, IP address of system used by the user, current date, Login time, logout time and number of minute's system is idle. And also admin can view the particular user idle time detail.

**De-Register:**In this module authorized person can delete the old users when no longer needed.

**Logout:**

User shut down the computer.

**Conclusion:**

This project helps us in tracking the User Idle Time. It will not consider processes running in the background. Currently it is mainly based on the input cursor. In future, it can be enhanced to find other input instance strokes.

**References:**

- <https://www.w3schools.com>
- <https://stackoverflow.com>
- <https://codeproject.com>