IT7320 – Software Development and Testing

Assignment 1

Time Keeping Application

*Compiled August 2018*

*Aleisha Perry - 2115039*

*Anthony Sirvid - 2150957*

*Dan Mota - 2150708*

For

*Chalinor Baliuag*

*Due Date: Friday 10th August, 2018 – 11:55pm*

Table of Contents

[Executive Summary 1](#_Toc521496731)

[Opportunity Context 2](#_Toc521496732)

[Requirements 2](#_Toc521496733)

[Planning & Design 3](#_Toc521496734)

[Initial Features & Functions 3](#_Toc521496735)

[Advanced Features & Functions 3](#_Toc521496736)

[Resources 3](#_Toc521496737)

[System Components 3](#_Toc521496738)

[Development components: 3](#_Toc521496739)

[Database components: 3](#_Toc521496740)

[Human Resources 4](#_Toc521496741)

[Classes 4](#_Toc521496742)

[User 4](#_Toc521496743)

[DBConnection 4](#_Toc521496744)

[StopWatch 5](#_Toc521496745)

[Tracker 5](#_Toc521496746)

[Project 5](#_Toc521496747)

[Task 5](#_Toc521496748)

# Executive Summary

Students in today’s society share the common issue of having poor time management skills that negatively affect their classes and study time. Our goal is to develop a timekeeping desktop application connected to an online database where individuals can create time schemes and measure their work done against their time estimates. This will help students to gauge their performance and improve their time management skills.

This project involves app and database development with java and MySQL. We will create an application that tracks time against predetermined times for projects.We will also develop a database to record user logins, projects, and tasks.This will be an open source project upon the completion of development, so future teams can modify and better the application to their specifications.

This project will be completed on or by Friday 10th August, 2018. There will be no cost in developing this project, and the client is ourselves.

We are a team of three students completing our undergraduate studies at WelTec. Our project team involves the project manager Anthony Sirvid, and our lead developer Dan Mota, and developer and documenter Aleisha Perry.

# Opportunity Context

Students are notorious for having bad time management skills, especially when it comes to their study. Time management tools are increasingly becoming the way of managing different tasks and projects people may have, and of keeping people’s schedules in one place for ease and simplicity. Not keeping track of tasks and mismanagement of a student’s time often leads to negative impacts on their work.

It is often tedious to manually create and keep track of time schemes, especially with all their related tasks and the times dedicated to each one. As such, there is a need to develop a tool to automate this process. There are already many tools available on the market, one for example, MS Project. This software package has a price tag of around two thousand US dollars, often remaining outside the means of access for students.

This project is designed to be a free alternative to MS Project for students.

# Requirements

We will be developing a ubiquitous time management system that provides a platform students can track and manage their project tasks with. This will be achieved by creating an application where students can create time schemes and track their progress. The system will be able to record and save created projects, tasks and their related notes.

# Planning & Design

## Initial Features & Functions

* Time stamp - beginning and end
* Notes – describe tasks during the time stamp periods
* Desktop application – access to online database
* Data/information stored online database
* Free user registration for individuals
* Open source
* History documentation – displays all time stamps with their recorded tasks
* Comparison of calendar and history, store as documentation
* Project tracker – convert tasks into a time percentage to show user how well or poorly they are doing against their initial planned time (green light/red light)
* Project selection – multiple projects, select between them

## Advanced Features & Functions

* Teams feature
* Email registration – “forgot password” function
* Import function
* Calendar events – remind users of upcoming deadlines, time remaining

# Resources

## System Components

### Development components:

* Windows 10 Enterprise
* Java IDE – Eclipse Photon
* Eclipse Photon v4.8.0
* Java Runtime Environment: Java SE-1.8
* Internet connection
* EGit v5.0.1.201806211838-r
* MySQL Workbench 6.3 CE v8.0.12 for Windows (x86, 64-bit)
* MySQL JDBC Driver v5.1.46

### Database components:

* Web Database – Wamp 3.1.3
* MySQL Workbench 6.3 CE
* User
* Projects
* Task

User components:

* Application
* Internet connection

## Human Resources

* Anthony - Project Team Leader, Designer, Developer, Documenter
* Aleisha - Documentation Leader, Designer, Developer
* Dan – Developer Leader, Designer, Tester, Documenter

## Classes

### User

Methods

### DBConnection

Methods

*public int checkCredentialsConflict( User user )*

Auxiliary method for creating a new user. The methods queries the database for matching entries for username and email address which must be unique. The method returns a 0 if no conflicts are found, a 1 if a matching username exists, and 2 if a matching email address exists in the database.

*public boolean createUser( User user )*

This method attempts to connect to the database. Upon success, a prepared insert statement is created. The username, password, and email attributes of class User are set into the statement. Lastly, the statement is executed and both prepared statement and connection are closed. True is returned if the insert is performed successfully, otherwise the method returns false. The boolean is used by SignUpGUI’s evaluateResult method to either display an information alert to the user if true, or an error message if false.

*public boolean login( String handle, String password )*

This method attempts to connect to the database. Upon success, a prepared select statement is created. The handle parameter is set into the statement and the query is executed. If a match is found, the variables dbUsername, dbPassword, and dbEmail are initialised using the ResultSet. If either the variables dbUsername and dbPassword, or the dbEmail and dbPassword matches the method parameters, true is returned after closing the statement and connection. Otherwise, a false signal is returned. These returns are used to either transition to the next GUI or alert the user that incorrect credentials have been entered.

### StopWatchGUI

Methods

### TrackerGUI

Methods

*private void openREADME()*

This method will open the README text file from the directory in Notepad. It executes when the user goes to Help on the menu bar.

*private void logout()*

This method will log the user out of the system. It will disconnect the connection to the database, hide the current gui, and load the LoginGUI gui. It executes when the user goes to Menu on the menu bar, and selects Logout.

*private void loadStopWatch()*

This method will load the StopWatchGUI gui, closing the currently open window. It executes when the user goes to Menu on the menu bar, and selects StopWatch.

*public void populateProjectTable(Project project, DefaultTableModel projTblModel)*

This method will populate the table with all tasks and each task’ notes, depending on which project the user selected.

*public void loadCreateTask()*

This method will load the TaskTrackerGUI gui, closing the currently open window. It executes when the user goes to Menu on the menu bar, and selects Task Tracker.

### CreateTaskGUI

Methods

*private void openREADME()*

**Same as TrackerGUI method of same name**

*private void logout()*

**Same as TrackerGUI method of same name**

*private void loadStopWatch()*

**Same as TrackerGUI method of same name**

*public void loadTaskTracker()*

This method will load the TaskTrackerGUI gui, closing the currently open window. It executes when the user goes to Menu on the menu bar, and selects Task Tracker.

*public void createTask(String userToken, JTextField tName, JTextArea tDesc)*

This method will pull the data the user has entered from the text field and text field area on the gui and store them directly in the database.

*public void resetFields(JTextField tName, JTextArea tDesc)*

This method will simply reset the text field and text feild area on the gui, as in, delete all data inputted and set it to a blank string.

### Project

Methods

### Task

Methods

### Note

Methods