Time Sheet – Statement of Requirements

COMP3910 Assignment 1

Tony Pacheco & Danny DiIorio

Table of Contents

[Purpose 2](#_Toc528405613)

[Scope 2](#_Toc528405614)

[Perspective 2](#_Toc528405615)

[Class Diagram 3](#_Toc528405616)

[Use Cases 4](#_Toc528405617)

[Test Plan 5](#_Toc528405618)

[Interface Requirements 6](#_Toc528405619)

[General Users 6](#_Toc528405620)

[Administrators 6](#_Toc528405621)

[Operating Environment 6](#_Toc528405622)

[UI Mock-ups 6](#_Toc528405623)

[Build and Run Instructions 7](#_Toc528405624)

[Configurations 7](#_Toc528405625)

[Importing 7](#_Toc528405626)

# Purpose

The Timesheet application will be an online system which allows users to login and access/edit workplace timesheets and allows an administrator to manage the user’s accounts.

# Scope

The Timesheet application will be implemented without a database and will instead only store data as long as the server is left running. Because of this, it will also not be a truly distributed application; only keeping data on the local machine running it.

# Perspective

The application will store the following data in its Java backend:

1. User (Employee) Data for each user in the system
   1. Name
   2. Employee Number
   3. Username
2. Timesheet Data

Each timesheet records the hours worked by a single employee on a given week. Each will contain the following data:

1. Employee Number
2. Employee Name
3. Week Number (0-52)
4. Week Specifier (the date on which the week ends)
5. The following, each as a set of data to displayed as the rows of a timesheet table
   1. Project Number
   2. Work Package identifier
   3. Total number of hours worked for the week
   4. A column for each day of the week containing the number of hours worked that day
   5. Additional optional notes

Each row above represents a week of work hours on a given work package of a specific project.

# Class Diagram



# Use Cases

The two types of users which the application will support are general users and the system’s administrator. General users will be able to manage their own account settings and create and edit their own timesheets. Administrators will be able to do anything that general users can do, as well as manage the creation, modification, and deletion of other user accounts.



# Test Plan

1. Login page
   1. Users will enter a correct combination of password and username
2. Logout
   1. User can logout of their account by pressing the logout button in the header dropdown menu
   2. After pressing the logout button, they are directed to the login page
3. User landing page
   1. Page with a list of saved timesheets (empty at first)
   2. Header containing links to View Timesheets, New Timesheet, User dropdown
   3. A welcome message will appear with the user’s name on it
4. Administrator landing page
   1. Same as page as users
   2. Header will have an additional link Users, which will lead to a list of users
5. View timesheet
   1. This page shows a list of saved timesheets, organized by date
6. Create timesheet
   1. When a new timesheet is created five rows will appear
   2. It will be defaulted as current week
   3. Clicking on the add row button will add a row to the timesheet table
   4. Time entered during the week must be in the unit of hour from 0.0 to 24.0, it may be integer or with one decimal place
   5. WP (Working Project) must have alphabetical value combined with numerical value
   6. Total number of hours should appear before the column of the day of the week column
      1. The calculation is done automatically; however, it is only done when the user clicks the save button
7. View users page
   1. This page will show all the users in a list with all their information
   2. The admin can click on a button to edit, directing them to an edit page
   3. The admin can also click on a remove button to delete the user
8. Edit users page
   1. The admin can edit all fields related to the user
   2. The user’s password can be reset to a default value by clicking the reset password button
9. New user page
   1. This page allows a manager to create a new user by filling out the following fields:
      1. Employee number
      2. Name
      3. Username
      4. Password

# Interface Requirements

## General Users

When a user first starts the application, they will see a login page, if they have a registered account, they can login with their username and password. If they do not have a registered account, an administrator will need to create an account for them.

After a registered user successfully logs in, they will see a page with the list of their saved timesheets, organized by date. Each timesheet can be opened and edited by clicking on it.

Users can create a new timesheet by clicking on “New Timesheet” in the header. This will open a page with a new timesheet with five empty rows. By clicking on a pencil icon in the last column of the table, the user will be able to edit the contents of that row in the table, which allows them to input/edit hours worked for any day of the week.

This page will also have buttons on a top toolbar which allow them to view a different week’s timesheet, create a new timesheet, change their password, or logout of the application.

## Administrators

If an administrator logs in to the application, they will see a page with the list of their saved timesheets, organized by date. Each timesheet can be opened and edited by clicking on it.

Admins have a header link to see a page with a list of all the users in the system. They will be able to click on an edit button beside each user to edit them, click a button beside each user to remove them, or click a button at the bottom of the list to create a new user.

The administrators will also have the same options in the toolbar as the general users so that they can create and edit their own timesheets. They will have access to view other users’ timesheets.

# Operating Environment

The back-end will be implemented using Java instead of a database, and a Wildfly 13 server to serve the application webpages.

The front-end will use Java Server Faces and Prime Faces for the user interface. Front-end styling is handled by a custom stylesheet and Materialize 3rd party CSS library.

The management of the state of the application, and communication between the mock-database and the user interface will be implemented using Java Beans

# UI Mock-ups

See image accompanying this document.

# Build and Run Instructions

## Configurations

* Java development Kit 8
* Eclipse Java EE IDE
* Wildfly 13 and JBOSS\_HOME define

## Importing

File->import projects from folder -> and Import successfully