

# Time Sheet – Statement of Requirements

COMP3910 Assignment 1

Tony Pacheco & Danny Dilorio

## Table of Contents

1. Purpose .....	2
2. Scope.....	2
3. Perspective.....	2
4. Class Diagram .....	3
5. Use Cases .....	4
6. Interface Requirements .....	5
General Users.....	5
Administrators .....	5
7. Operating Environment .....	5
8. UI Mock-ups .....	5

## 1. 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.

## 2. 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.

## 3. Perspective

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

### 1. User (Employee) Data for each user in the system

- a. Name
- b. Employee Number
- c. Username

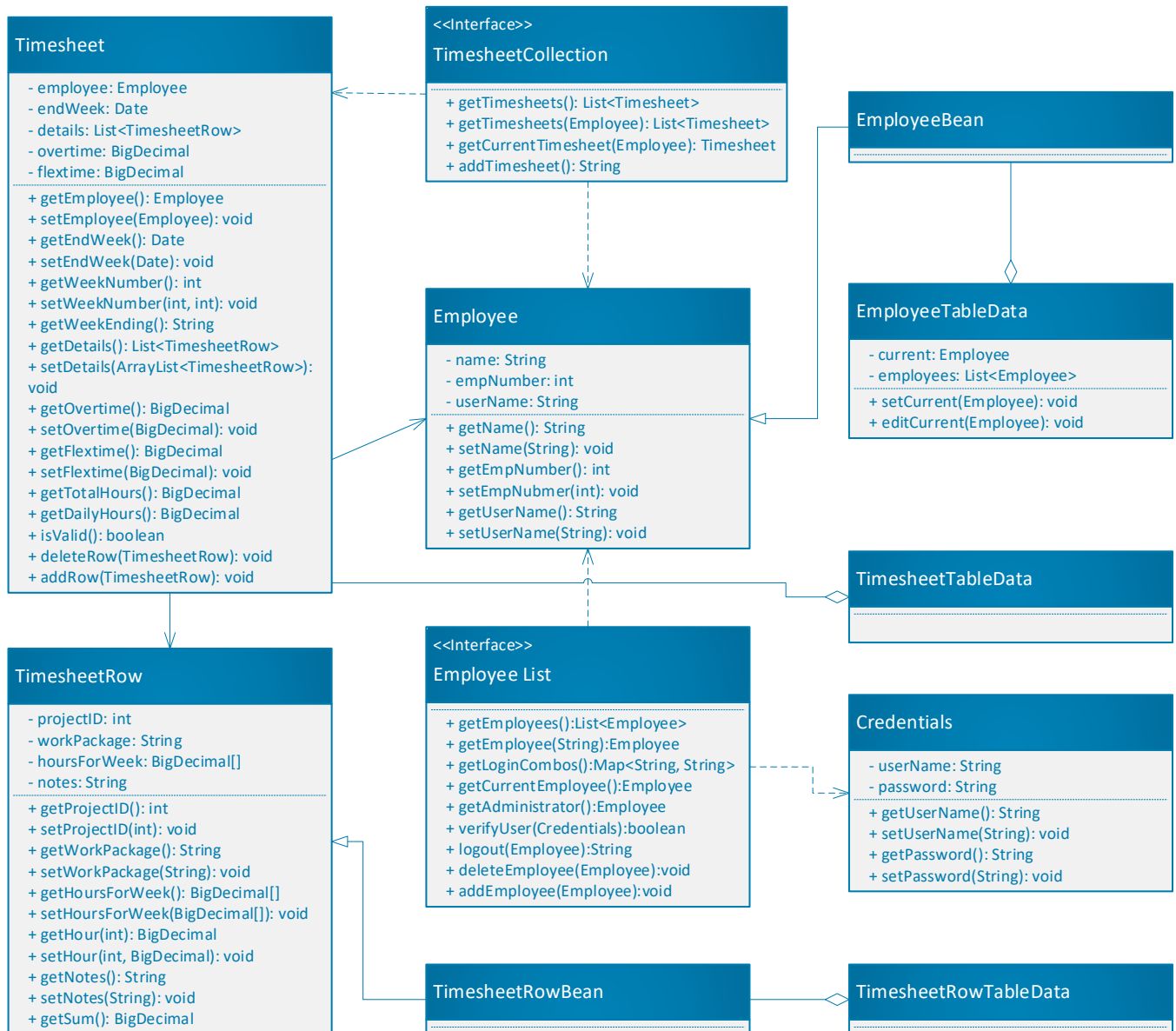
### 2. Timesheet Data

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

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

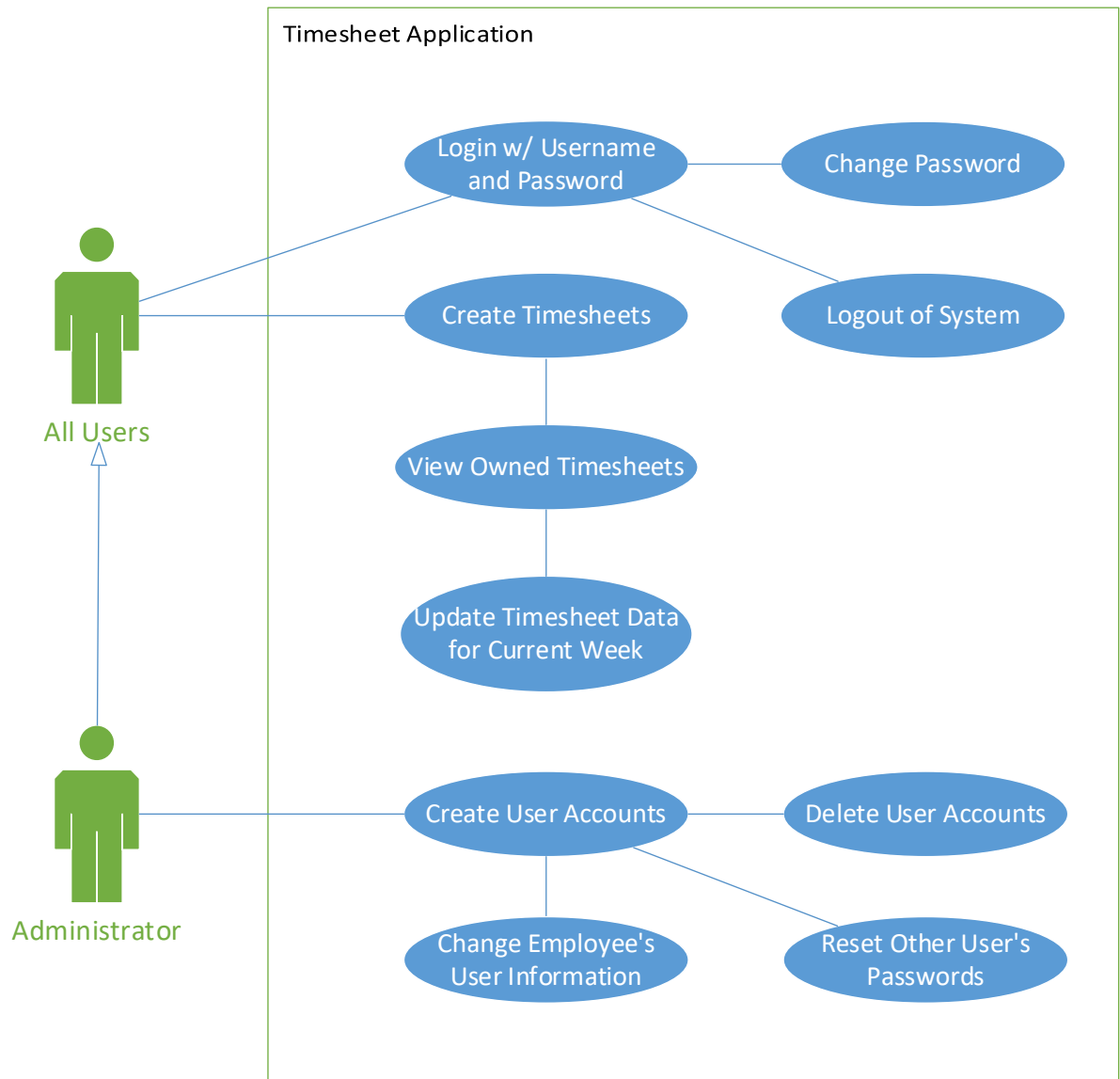
Each row above represents a week of work hours on a given work package of a specific project.

## 4. Class Diagram



## 5. 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.



## 6. 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 timesheet for the current week. If they do not have a time sheet created for the current week already, they will see an empty timesheet with five empty rows.

By clicking on a checkbox 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 instead see a page with a list of all the users in the system, and they will be able to click on any user to edit 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 not have access to other users' timesheets.

## 7. 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.

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

## 8. UI Mock-ups

See image accompanying this document.