Timesheet RESTful Interface

COMP3910 Assignment 3

Requirements & Design Document

Tony Pacheco & Danny Di Iorio

December 2, 2018

Table of Contents

[Purpose 2](#_Toc531435545)

[Scope 3](#_Toc531435546)

[Perspective 3](#_Toc531435547)

[Class Diagram 3](#_Toc531435548)

[Database Entity Relational Diagram 5](#_Toc531435549)

[Operating Environment 6](#_Toc531435550)

[REST API Specification 6](#_Toc531435551)

[Required Request Calls 6](#_Toc531435552)

# Purpose

The Timesheet RESTful service will provide an interface for logged in users to view, edit, add, and remove timesheet and employee data, through various HTTP requests.

# Scope

A JBoss Wildfly 13 server will provide a RESTful interface to all its services, while data storage will be handled by a MySQL database and Java Persistence Architecture transactions.

# Perspective

The application will store the following data in its database:

1. User (Employee) Data for each user in the system:
   1. First Name
   2. Last Name
   3. Employee Number
   4. Username
   5. Password
   6. Admin status
2. Timesheet Data:

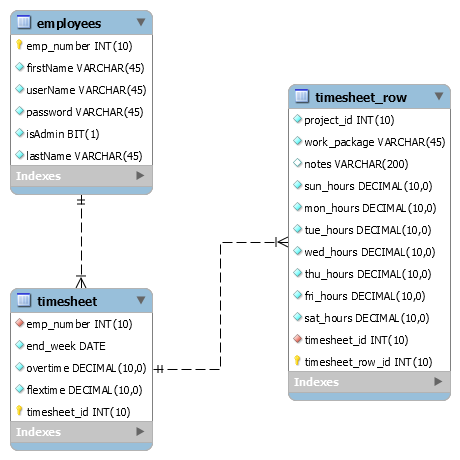
Each timesheet records the hours worked by a single employee on a given week. Each will contain the following data:

1. Employee Number
2. Employee Name
3. Week Number (0-52)
4. Week Specifier (the date on which the week ends – Friday)
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

# Database Entity Relational Diagram



# Operating Environment

The operating environment will be split up into three layers: *middleware* *tier*, *business objects tier*, and *persistence tier*.

The *middleware tier* consists of a RESTful interface, which provides request services for various GET, PUT, POST, and DELETE requests that interact with data stored in the database. Users will be required to successfully send a login request and receive a valid token prior to any interaction with the REST services.

The *business objects tier* will use JPA session and entity beans to handle the RESTful interface functionality and represent the data within the database, respectively. We have designed entity beans to represent each database table, with …..

The *persistence tier* consists of a MySQL relational database designed by us. The database will be connected to the other layers with a data source running on JBoss Wildfly 13 server.

# REST API Specification

## Required Request Calls

|  |  |  |
| --- | --- | --- |
| URI | HTTP Method | Payload Format |
| /auth | POST – sent username and password and receive token if it is valid combo | JSON |
| /timesheets | GET – list all timesheets | JSON |
|  | POST – submit a new timesheet | JSON |
| /timesheets/{timesheetId} | GET – get an existing timesheet | JSON |
|  | PUT – update an existing timesheet | JSON |
|  | DELETE – cancel an existing timesheet | JSON |
| /employees | GET – list all employees/users | JSON |
|  | POST – create a new employee/user | JSON |
| /employees/{empNumber} | GET – get an employee profile | JSON |
|  | DELETE – remove an employee | JSON |
| /employees/{empNumber}/timesheets | GET – get the timesheets for a specific employee | JSON |
|  |  |  |