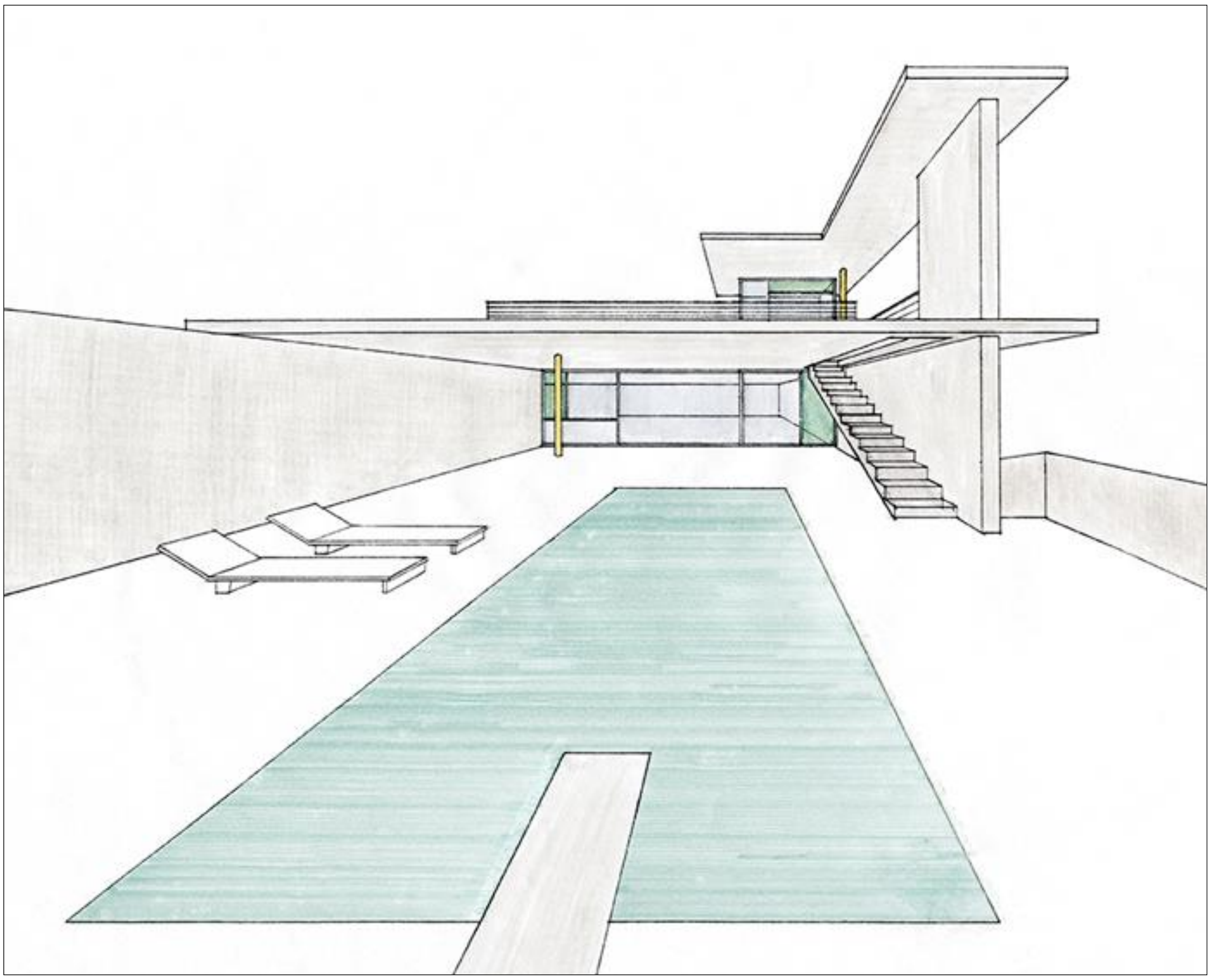
Project Manual—Time Sheet System



Team: Xray

Member: Orkut Karacalik

Chuyi Sun

Esha Agrawal

Bibi Nazneen

29. September 2018

Contents

Contents

Introduction

The German federal law for minimum wages („Mindestlohngesetz “) requires that employees with a monthly income below a certain amount (currently 850€) fill time sheets about their working hours, vacation, illness, etc. This enables examination and verification of the minimum wage guaranteed by the law.

In the university context, this means that most of the student assistants must report the times on a weekly base. Both the student assistant as the employee as well as the supervisor as representative of the employer should sign the time sheets. The university as the employer has the obligation to file the time sheets for 2 years. After that period, the time sheets shall be destructed.

The whole process is tedious and error-prone since the current solution (spreadsheets or paper forms) are inconvenient, students as well as staff easily miss to fill, print, sign, and archive the forms in a timely manner.

The Time Sheet System (TSS aims at implementing a web application that can be used to record the working time and remind the employees to sign the timesheet on time.

This user manual will provide instructions on how to access the TSS and installing it on JavaEE 7, Payara server and NetBeans. In this application, four roles are designed: assistant, supervisor, secretary, employee.

A list of Requirements

List of Completed and Missed Requirements

Functional Requirements

List of Completed requirements for **Contract** Sub Module.

Requirement No	**Short Description**	**Status**
CN1	Ability to manage Contracts	Completed
CN3	View contract Statistics	Completed
CN2	Ability to print contracts	Completed
CN4	Calculate total hours due, vacation hours, remaining hours due	Completed
CN4a	Calculation of vacation hours	Completed
CN4b	Calculation of total hours due for the contract as the sum of hours due of individual timesheet	Completed
CN4c	Calculation of hours due	Completed
CN4d	Determine the public holidays in Rheinland-Platinate	Completed

CN4e	Determine the public holidays in Germany from 01-January-2018 until 31-December-2027	Completed
CN5	Set a contract to PREPARED status as soon as it is created	Completed
CN6	Ensure that start date, end date, frequency, hours per week, vacation hours, working days per week, and vacation days per year can only be changed when the contract is in PREPARED status.	Completed
CN7	Provide assistants and supervisors with the ability to start a contract.	Completed
CN9	Ensure that only contracts can be aborted whose time sheets are in status SIGNED_BY_SUPERVISOR or IN_PROGRESS.	Completed
CN11	Warn the user if there are time sheets in	Completed

	state IN_PROGRESS that have entries, before terminating the contract	
CN12	Set a contract to status ARCHIVED as soon as all time sheets of that contract are in status ARCHIVED	Completed

List of completed requirements for **Time Sheets** Sub Module.

Requirement No	**Short Description**	**Status**
TS1	Create Timesheets for Contract once started	Completed
TS2	Timesheet entries can be changed only when the Timesheet is in IN_PROGRESS state	Completed
TS3	delete time sheets in status IN_PROGRESS once the contract is terminated	Completed
TS5	Do not delete time sheets that are in the SIGNED_BY_SUPERVISOR state	Completed

TS6	Ability to view time sheets	Completed
TS7	Ability to print time sheets	Completed
TS8	Ability to manage time sheet entries when time sheet is in IN_PROGRESS state	Completed
TS9:	Managed time sheet entries Changes when time sheet is IN_PROGRESS	completed

List of completed requirements for **Signatures** Sub Module.

Requirement No	**Short Description**	**Status**
SG1	Provide employees with the ability to sign a time sheet	Completed
SG2	Provide supervisors with the ability to sign a time sheet that is in status SIGNED_BY_EMPLOYEE	Completed

List of completed requirements for **Reminders** Sub Module.

Requirement No	**Short Description**	**Status**
RE1	Send reminder mail to employee on last day of	Complete

	timesheet if time sheet is in IN_PROGRESS state	

List of completed requirements for **Archiving** Sub Module.

Requirement No	**Short Description**	**Status**
AR1	Ability to archive time sheets that are in status SIGNED_BY_SUPERVISOR	Completed
AR3	Delete time sheets 2 years after the signature of the supervisor	Completed
AR4	Support variable archive durations	Completed

Nonfunctional requirements

List of completed nonfunctional requirements for **Access Control**.

Requirement No	**Short Description**	**Status**
AC1	authenticate users prior to giving access to any data	Completed

List of completed nonfunctional requirements for **User Interface**.

Requirement No	**Short Description**	**Status**
UI1	Support mobile devices	Completed

List of completed nonfunctional requirements for **Internationalization** Sub Module.

Requirement No	**Short Description**	**Status**
IN1	Change of User Interface Language	Completed
IN2	Ability to choose the language	Completed
IN3	Support at least two languages	Completed
IN4	Support English language	Completed
IN5	Support German language	Completed
IN6	Send reminders to users in preferred language	Completed

List of completed nonfunctional requirements for **Software Architecture**.

Requirement No	**Short Description**	**Status**
SA1	Implement layered architecture	Completed
SA2	Must contain the web and ejb module	Completed

SA3	Use third-party libraries after negotiation with the customer	Completed
SA4	Use of the database servers MySQL	Completed
SA5	Document architectural decisions	Complete
SA6	Prefix all global names with its team name	Complete

List of completed nonfunctional requirements for **Project Manual**.

Requirement No	**Short Description**	**Status**
PM1	document in a project manual	Complete
PM2	Maintain list of completed and incomplete requirements	Complete
PM3	Document description of problems occurred	Complete
PM4	Document decision made to change requirements	Made no changes to the requirements
PM5	Record time spent on project	complete

How to install the application?

System Setup

TSS is a web application, which means it runs on a server machine and responds to all the requests made by client via web browser.

A windows/Unix/Linux based system with following software and hardware requirements is required to run TSS project.

Client-side Software requirements

Latest JavaScript enabled browser to access the application.

Server-side Software requirements

Application Server (Payara)

Application server is required to host the application and provide the necessary environment for its execution. We will use Glassfish/Payara application server, available at https://www.payara.fish/downloads.

Database Server

TSS stores all data related to the system in a database. A DERBY database server is required to run TSS.

JAVA

Java Developer Kit (JDK) or Java Runtime Environment (JRE) platform is required to be installed on the server’s operating system. The latest version of JDK is available at http://www.oracle.com/technetwork/java/javase/downloads/index.html .

TSS application war

The latest version of the application in war format is submitted.

Server-side Hardware requirements

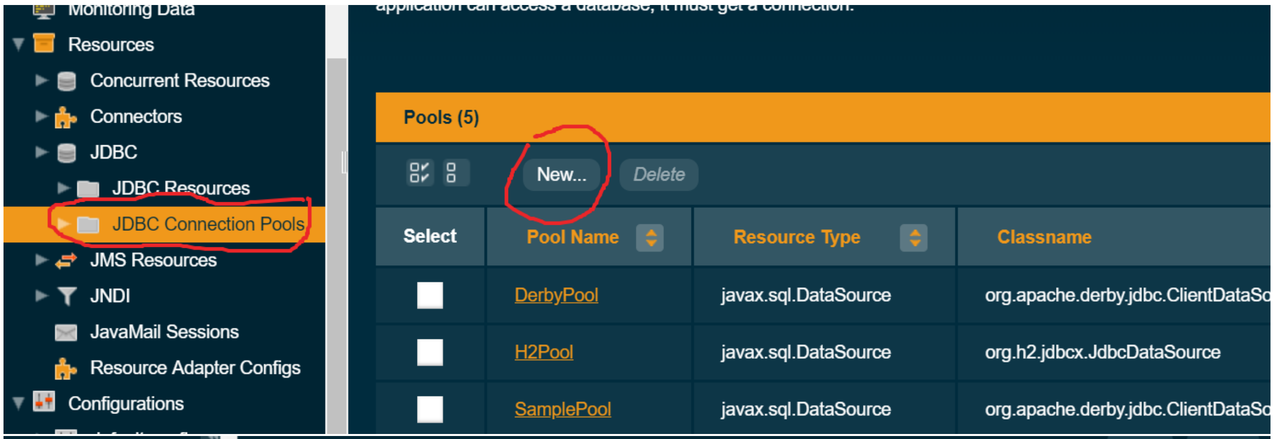
System with min 2 GB RAM.

Before installation: Create JDBC connection pool and resource

1. Start Payara Server from Terminal (How to start the server: https://docs.payara.fish/getting-started/getting-started.html)

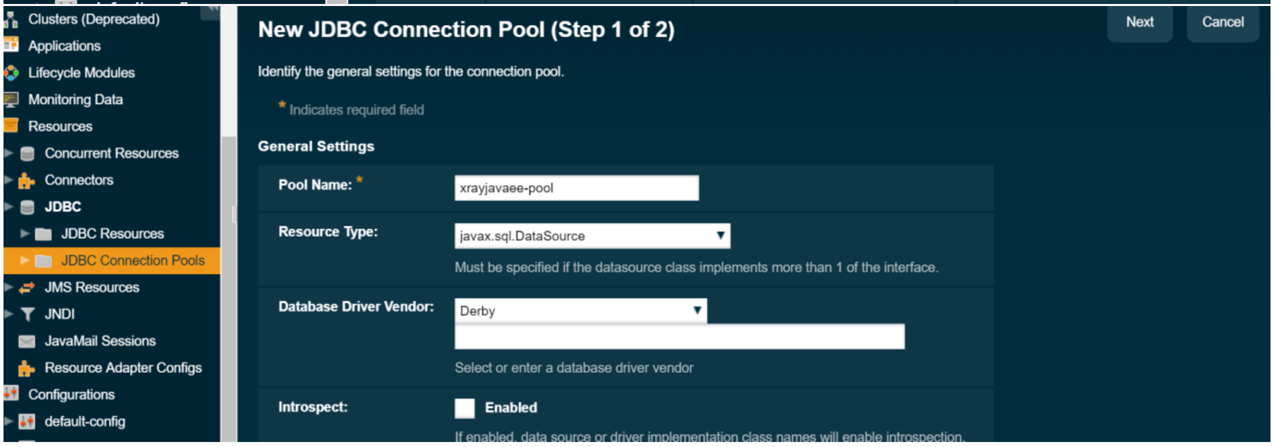
2. Navigate to http://localhost:4848 to access the Administration Console

3. Create a new JDBC Connection pool as following:



Pools (5)

Select	Pool Name	Resource Type	Classname
☐	DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSo
☐	H2Pool	javax.sql.DataSource	org.h2.jdbcx.JdbcDataSource
☐	SamplePool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSo



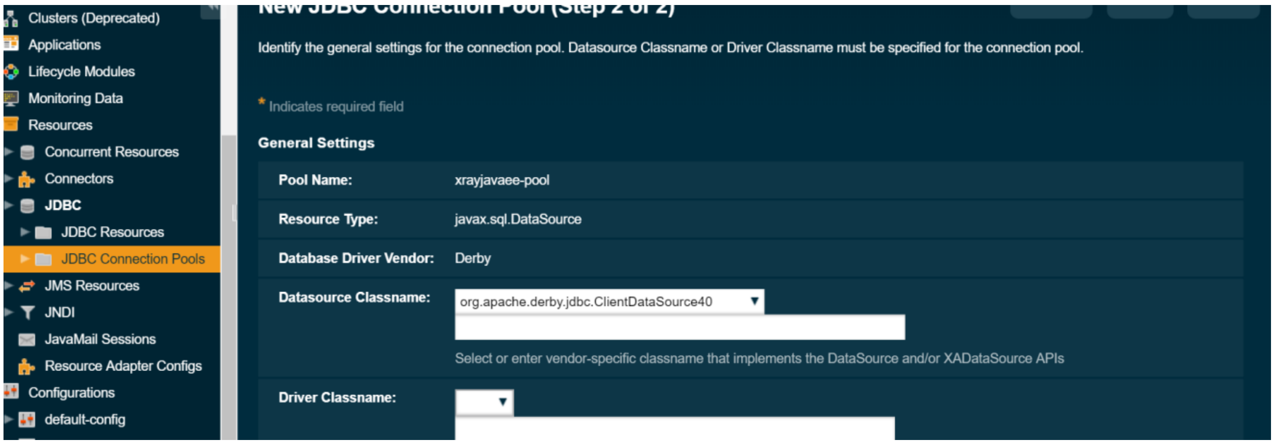
New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

* Indicates required field

General Settings

Pool Name: *	xrayjavaee-pool
Resource Type:	javax.sql.DataSource Must be specified if the datasource class implements more than 1 of the interface.
Database Driver Vendor:	Derby Select or enter a database driver vendor
Introspect:	☐ Enabled If enabled, data source or driver implementation class names will enable introspection.



New JDBC Connection Pool (Step 2 of 2)

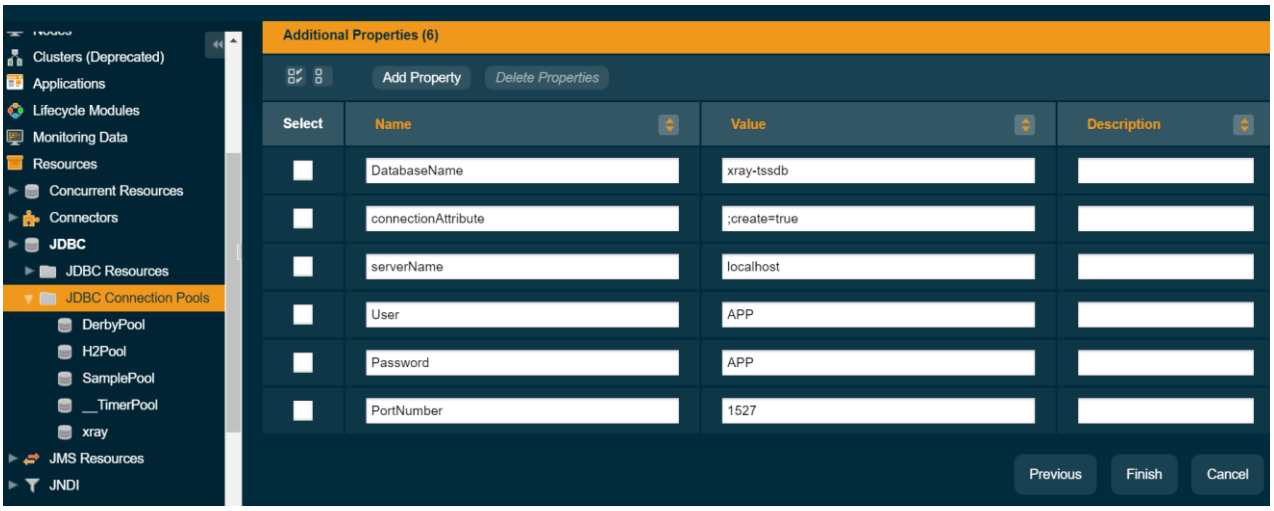
Identify the general settings for the connection pool. Datasource Classname or Driver Classname must be specified for the connection pool.

* Indicates required field

General Settings

Pool Name:	xrayjavaee-pool
Resource Type:	javax.sql.DataSource
Database Driver Vendor:	Derby
Datasource Classname:	org.apache.derby.jdbc.ClientDataSource40 Select or enter vendor-specific classname that implements the DataSource and/or XADataSource APIs
Driver Classname:	

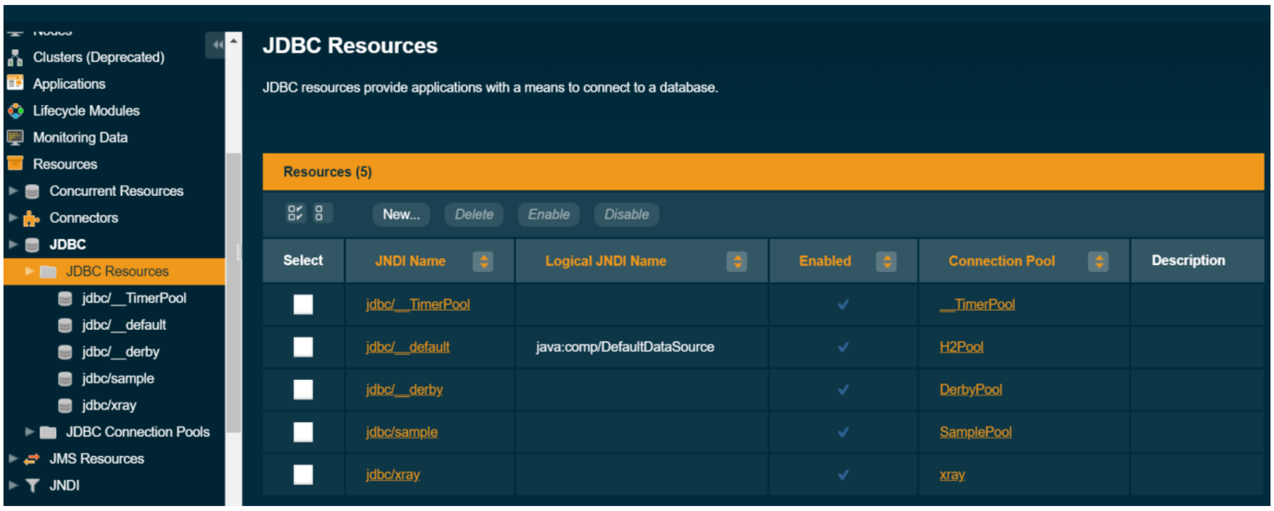
4. Edit additional properties



Additional Properties (6)

Select	Name	Value	Description
☐	DatabaseName	xray-tssdb	
☐	connectionAttribute	;create=true	
☐	serverName	localhost	
☐	User	APP	
☐	Password	APP	
☐	PortNumber	1527	

5. Create the new JDBC resource with name jdbc/xray using previous pool

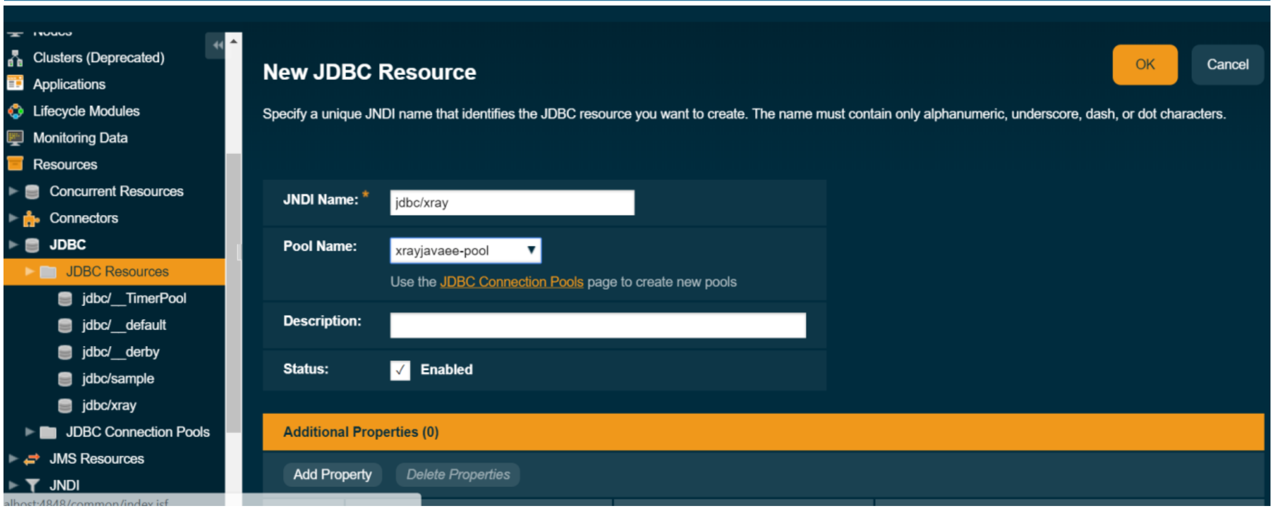


JDBC Resources

JDBC resources provide applications with a means to connect to a database.

Resources (5)

Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
☐	jdbc/__TimerPool		✓	__TimerPool	
☐	jdbc/__default	java:comp/DefaultDataSource	✓	H2Pool	
☐	jdbc/__derby		✓	DerbyPool	
☐	jdbc/sample		✓	SamplePool	
☐	jdbc/xray		✓	xray	



New JDBC Resource

Specify a unique JNDI name that identifies the JDBC resource you want to create. The name must contain only alphanumeric, underscore, dash, or dot characters.

JNDI Name: * jdbc/xray

Pool Name: xrayjavaee-pool

Use the JDBC Connection Pools page to create new pools

Description:

Status: ✓ Enabled

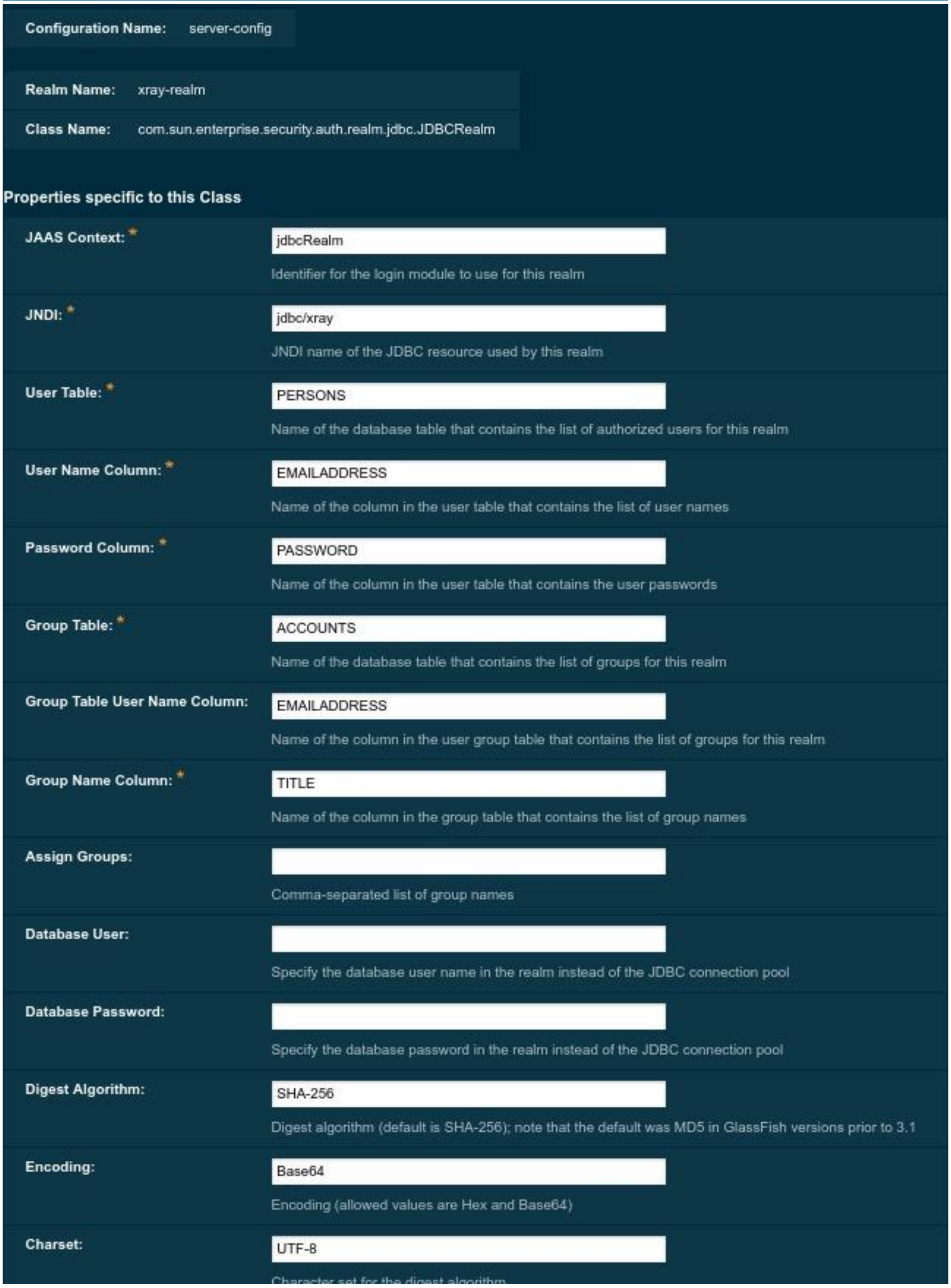
Additional Properties (0)

6. jdbc realm settings for payara

https://medium.com/@swhp/payara-security-realm-with-jdbc-9cddf0eec427

CREATE OR REPLACE VIEW ACCOUNTS SELECT R.PERSON_ID, R.ID AS ROLE_ID, P.EMAILADDRESS, P.PASSWORD, R.TITLE FROM APP.PERSONS P JOIN APP.ROLES R ON R.PERSON_ID = P.ID

This is is the sql query that should be executed before starting



Configuration Name:	server-config
Realm Name:	xray-realm
Class Name:	com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm

Properties specific to this Class

JAAS Context: *	jdbcRealm Identifier for the login module to use for this realm
JNDI: *	jdbc/xray JNDI name of the JDBC resource used by this realm
User Table: *	PERSONS Name of the database table that contains the list of authorized users for this realm
User Name Column: *	EMAILADDRESS Name of the column in the user table that contains the list of user names
Password Column: *	PASSWORD Name of the column in the user table that contains the user passwords
Group Table: *	ACCOUNTS Name of the database table that contains the list of groups for this realm
Group Table User Name Column:	EMAILADDRESS Name of the column in the user group table that contains the list of groups for this realm
Group Name Column: *	TITLE Name of the column in the group table that contains the list of group names
Assign Groups:	Comma-separated list of group names
Database User:	Specify the database user name in the realm instead of the JDBC connection pool
Database Password:	Specify the database password in the realm instead of the JDBC connection pool
Digest Algorithm:	SHA-256 Digest algorithm (default is SHA-256); note that the default was MD5 in GlassFish versions prior to 3.1
Encoding:	Base64 Encoding (allowed values are Hex and Base64)
Charset:	UTF-8 Character set for the digest algorithm

Installation

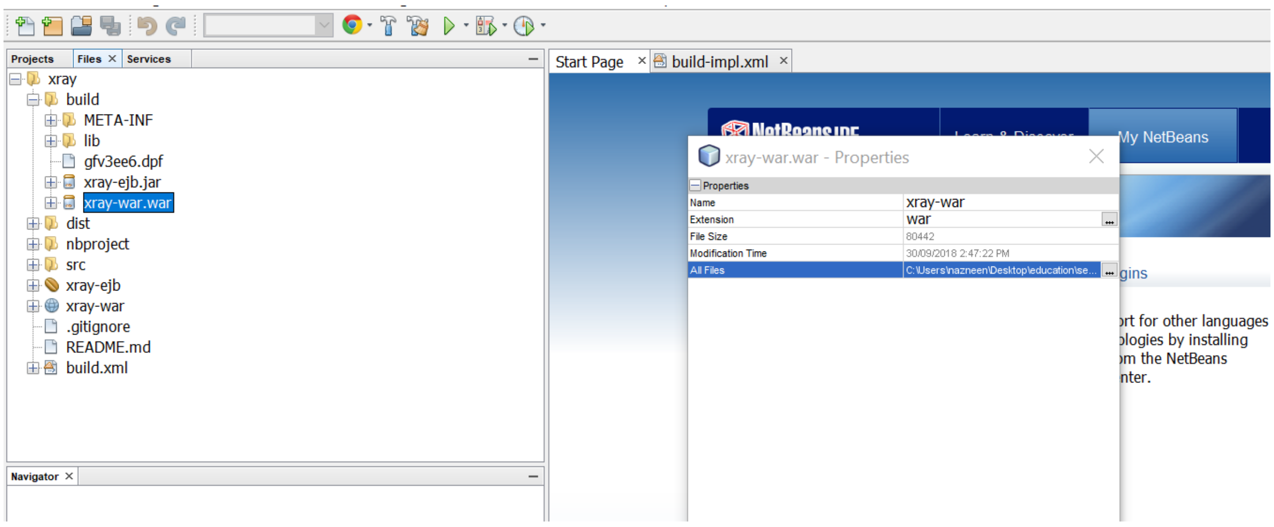
1) Download the xray project

2) Open the project in netbeans

3) Go to project **properties** -> **libraries** and import all required libraries and jar files.

4) clean build xray project in netbeans

5) if executed successfully then Go to **Files** -> **xray** -> **build** -> **xray-war.war** and go to its **properties** (by right click and select properties), from properties copy "**All Files**" path



```
xray
  build
    META-INF
    lib
    gfv3ee6.dpf
    xray-ejb.jar
    xray-war.war
  dist
  nbproject
  src
  xray-ejb
  xray-war
  .gitignore
  README.md
  build.xml
```

xray-war.war - Properties

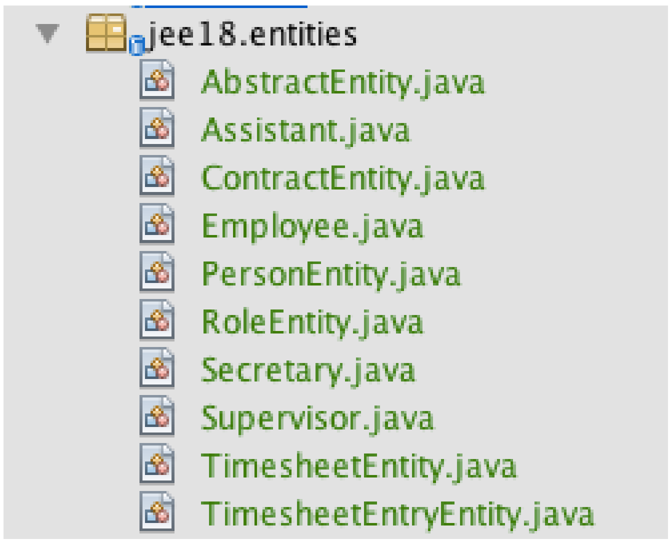
Properties	
Name	xray-war
Extension	war
File Size	80442
Modification Time	30/09/2018 2:47:22 PM
All Files	C:\Users\nazneen\Desktop\education\se...

6) go to cmd and deploy the war file " **install-directory/asadmin> deploy AllFilesPathToWar**"
(install-directory is where your Payara Server is installed)

7) Go to the browser and open localhost:8080/xray-war to see the index page

The starting point: Entity

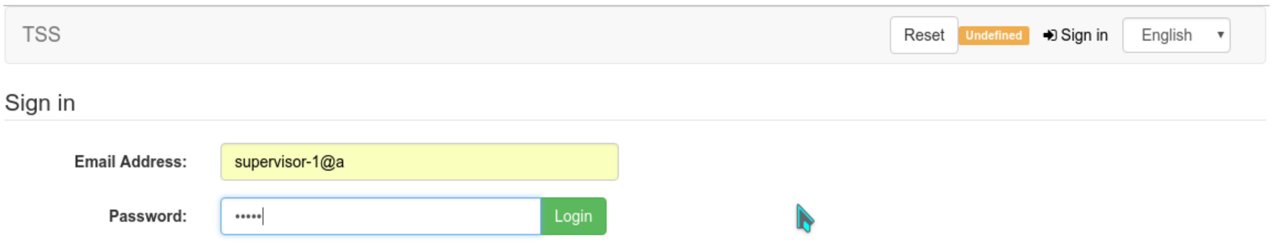
We start the project: from the entities classes and do some basic “CRUD” operations to test the connection with DB.



```
jee18.entities
    AbstractEntity.java
    Assistant.java
    ContractEntity.java
    Employee.java
    PersonEntity.java
    RoleEntity.java
    Secretary.java
    Supervisor.java
    TimesheetEntity.java
    TimesheetEntryEntity.java
```

Using the application

Login



TSS

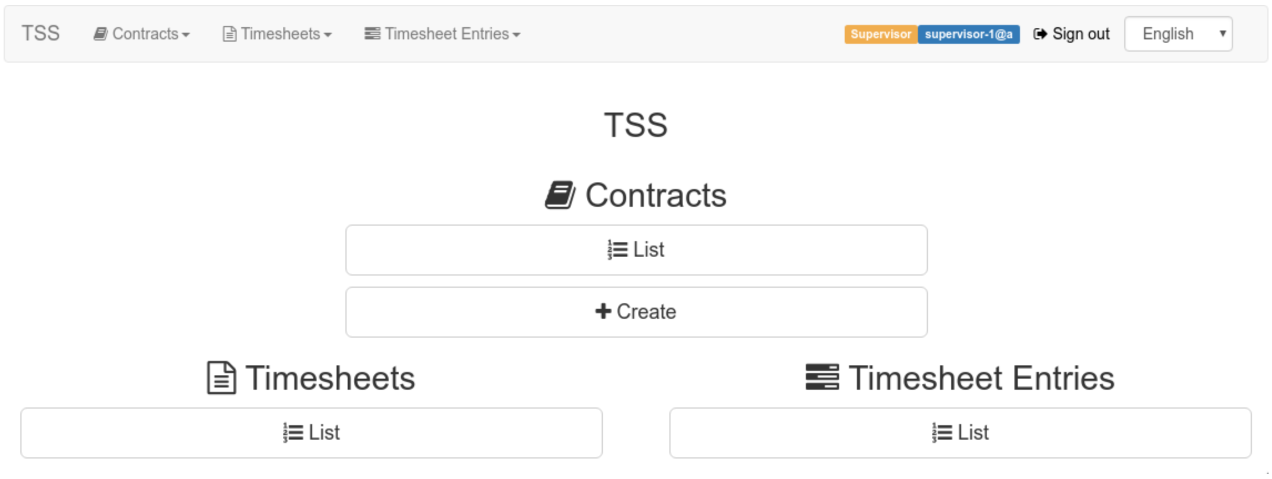
Reset Undefined Sign in English

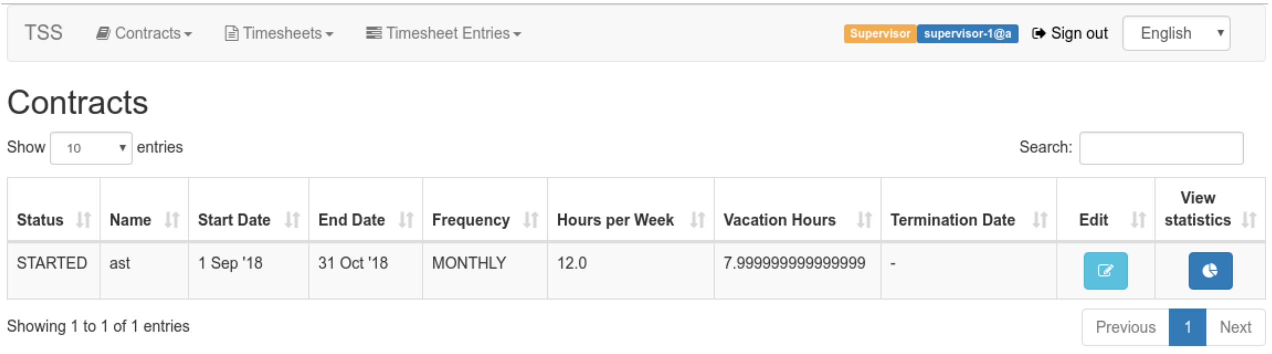
Sign in

Email Address:	supervisor-1@a
Password:	•••••

Login

Contract





Contracts

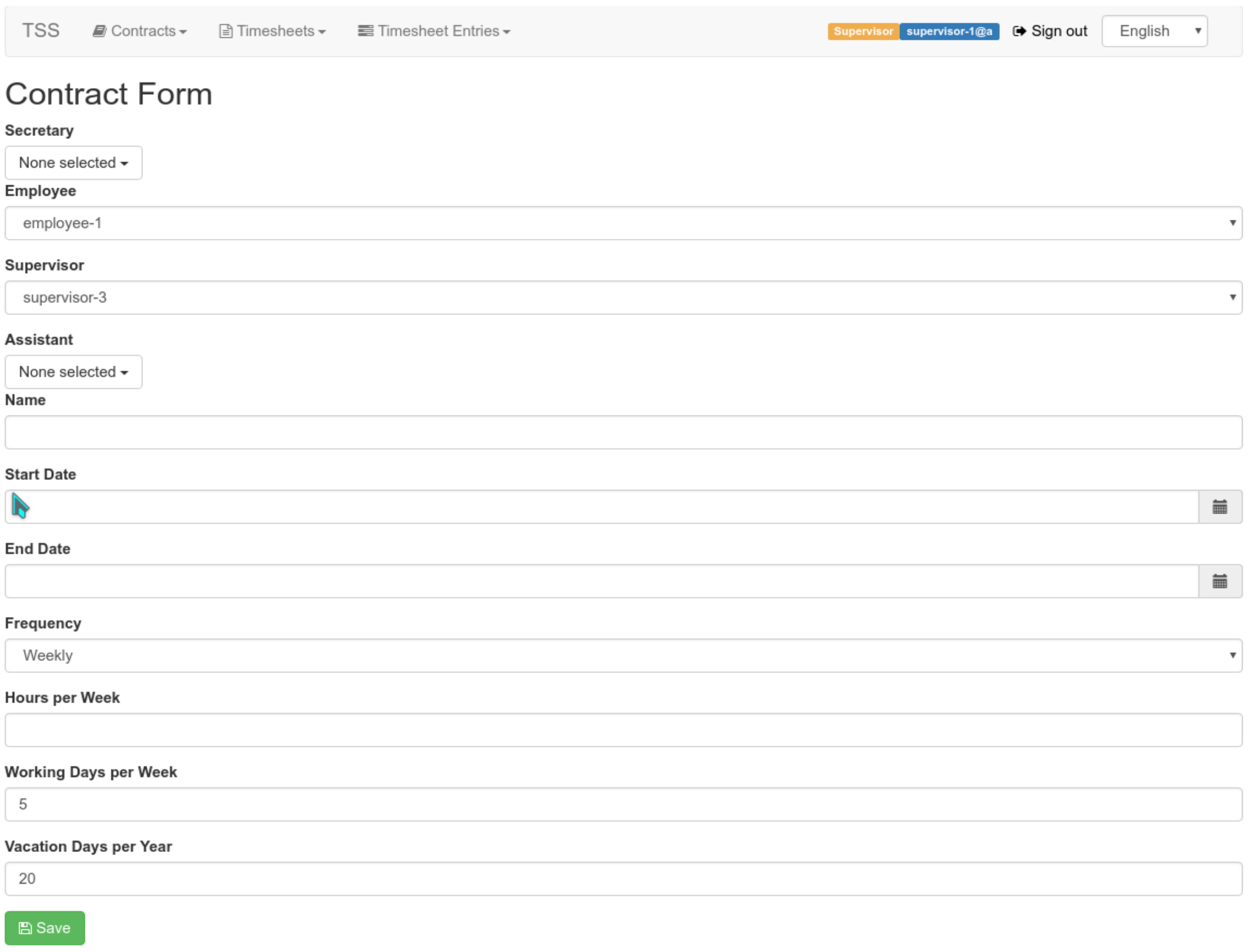
Show 10 entries

Search:

Status	Name	Start Date	End Date	Frequency	Hours per Week	Vacation Hours	Termination Date	Edit	View statistics
STARTED	ast	1 Sep '18	31 Oct '18	MONTHLY	12.0	7.999999999999999	-		

Showing 1 to 1 of 1 entries

Previous 1 Next

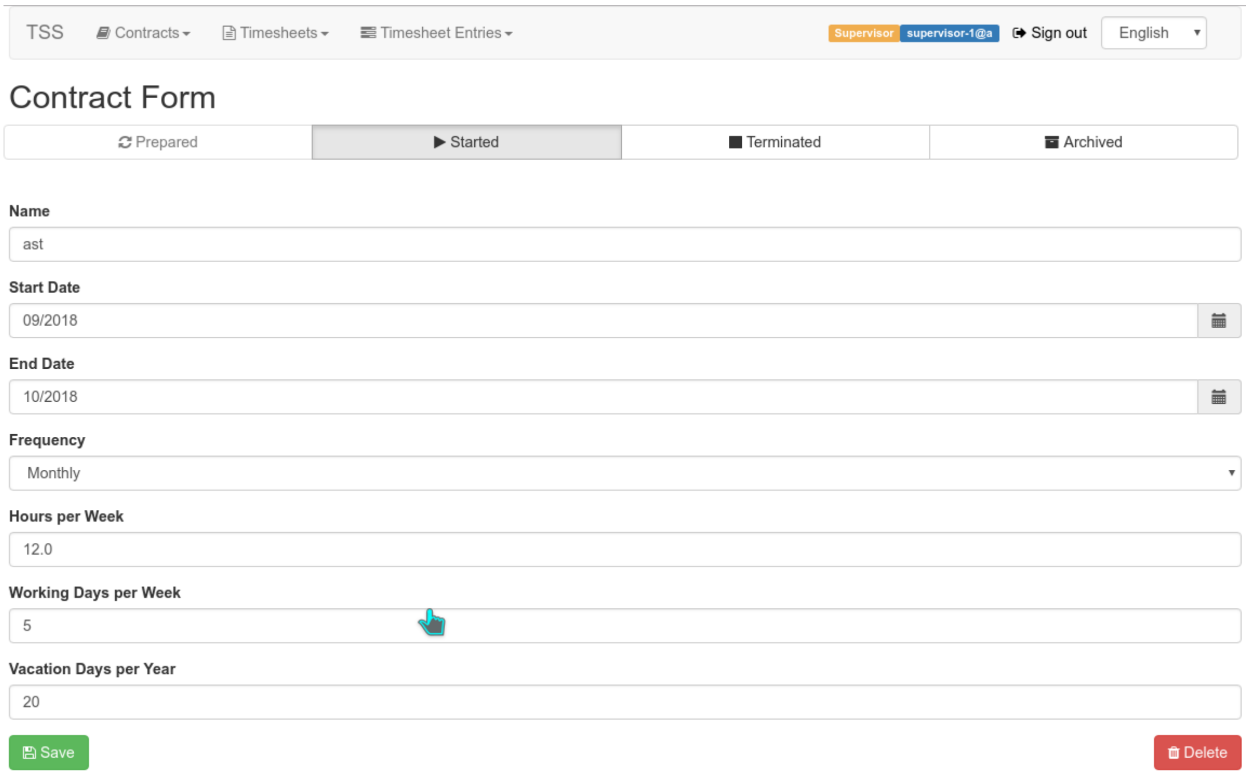


TSS | Contracts | Timesheets | Timesheet Entries | Supervisor supervisor-1@a | Sign out | English

Contract Form

Secretary	None selected
Employee	employee-1
Supervisor	supervisor-3
Assistant	None selected
Name	
Start Date	
End Date	
Frequency	Weekly
Hours per Week	
Working Days per Week	5
Vacation Days per Year	20

Save



TSS | Contracts | Timesheets | Timesheet Entries | Supervisor supervisor-1@a | Sign out | English

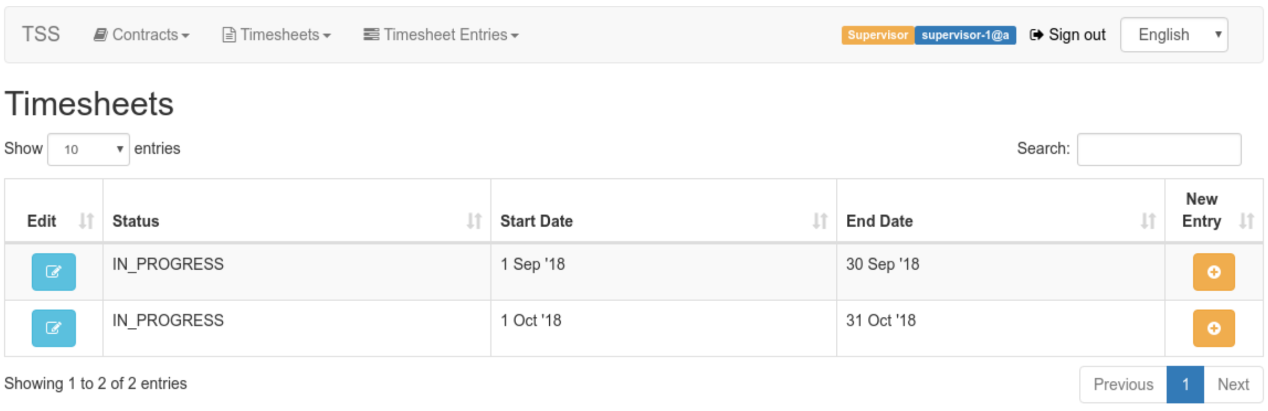
Contract Form

Prepared | Started | Terminated | Archived

Name	ast
Start Date	09/2018
End Date	10/2018
Frequency	Monthly
Hours per Week	12.0
Working Days per Week	5
Vacation Days per Year	20

Save | Delete

Timesheet

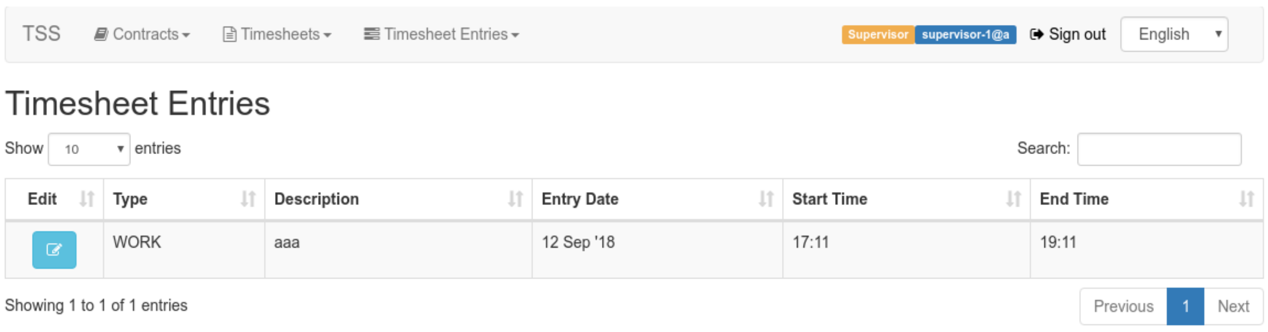


Timesheets

Show 10 entries　　Search:

Edit	Status	Start Date	End Date	New Entry
	IN_PROGRESS	1 Sep '18	30 Sep '18	
	IN_PROGRESS	1 Oct '18	31 Oct '18	

Showing 1 to 2 of 2 entries　　Previous 1 Next

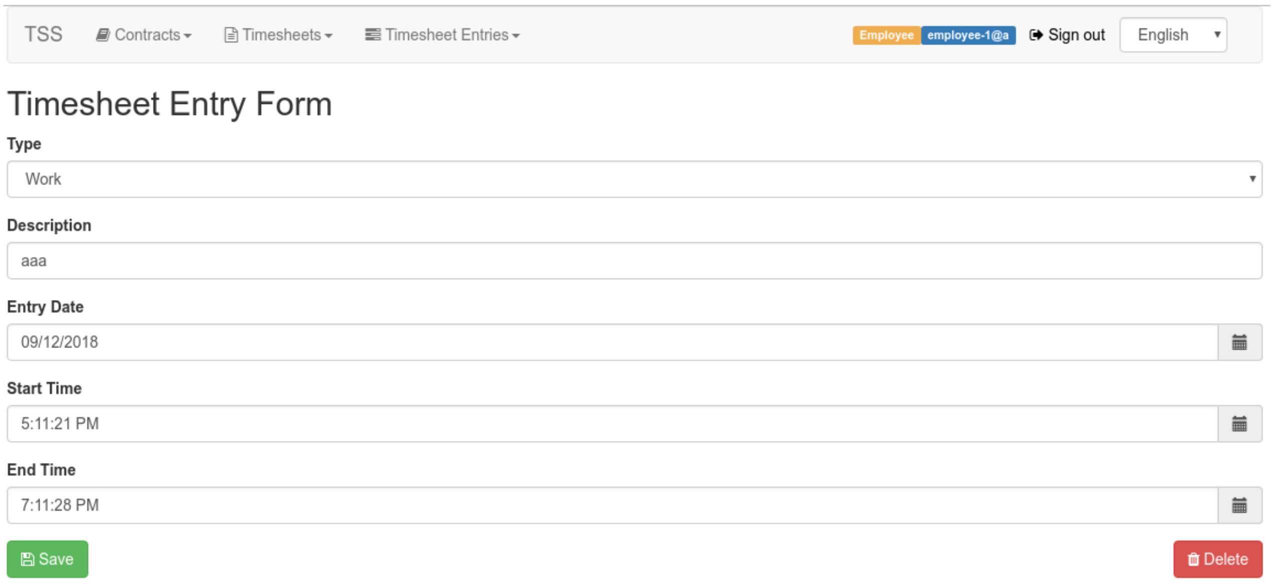


Timesheet Entries

Show 10 entries　　Search:

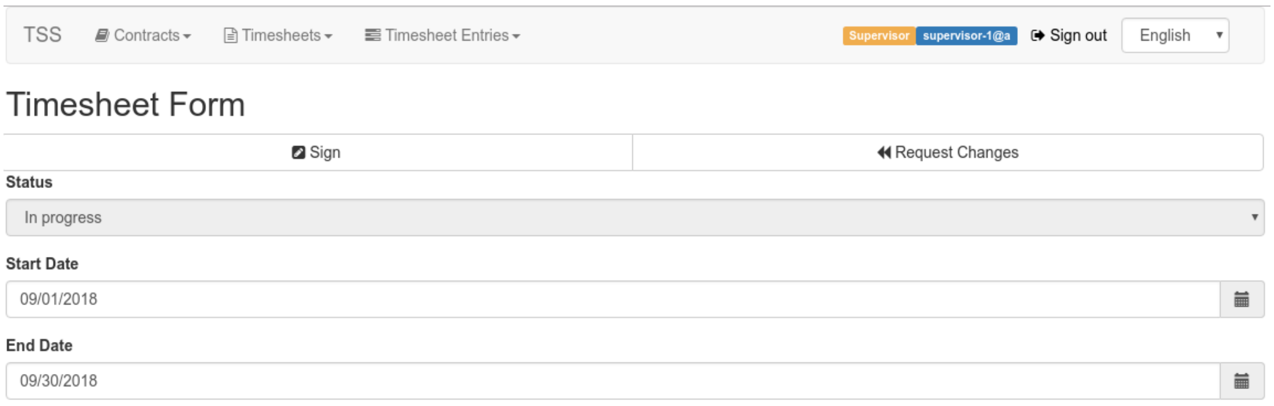
Edit	Type	Description	Entry Date	Start Time	End Time
	WORK	aaa	12 Sep '18	17:11	19:11

Showing 1 to 1 of 1 entries　　Previous 1 Next



Timesheet Entry Form

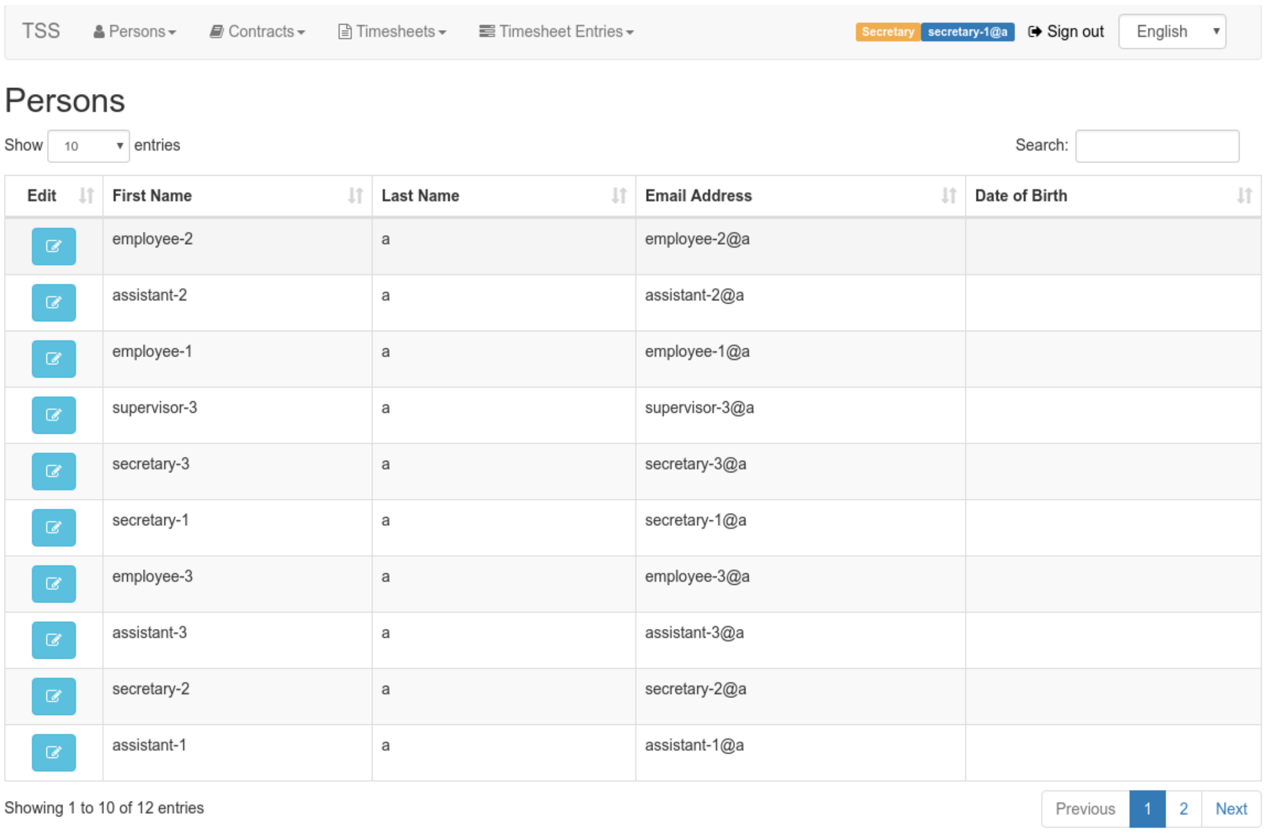
Type	Work
Description	aaa
Entry Date	09/12/2018
Start Time	5:11:21 PM
End Time	7:11:28 PM



Timesheet Form

Status	In progress
Start Date	09/01/2018
End Date	09/30/2018

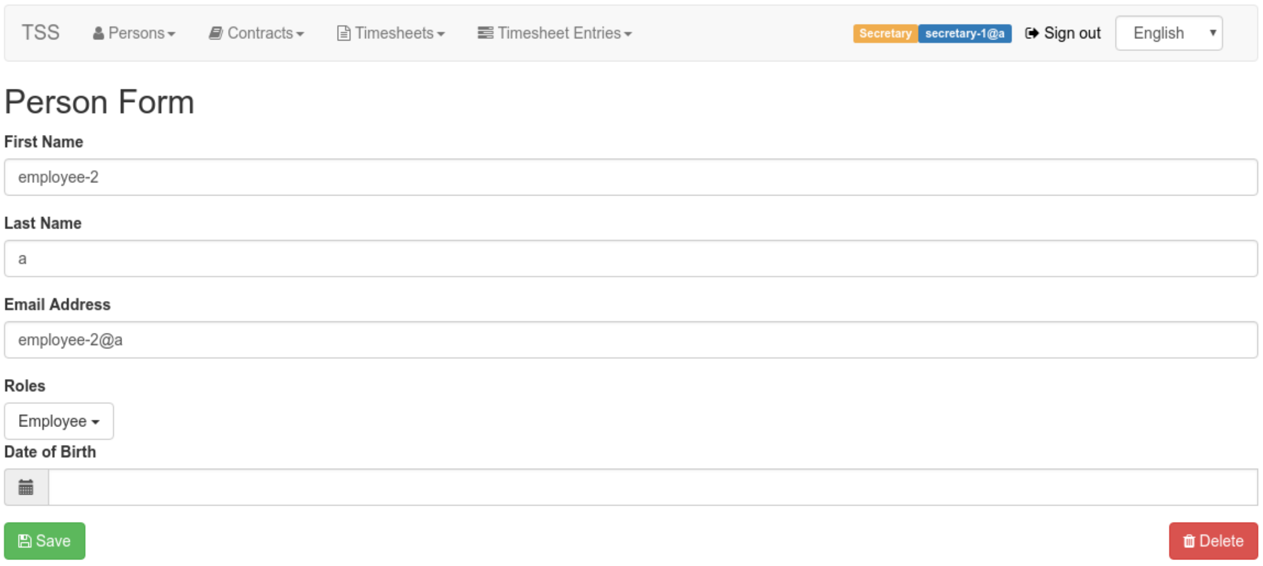
Person



Persons

Edit	First Name	Last Name	Email Address	Date of Birth
	employee-2	a	employee-2@a	
	assistant-2	a	assistant-2@a	
	employee-1	a	employee-1@a	
	supervisor-3	a	supervisor-3@a	
	secretary-3	a	secretary-3@a	
	secretary-1	a	secretary-1@a	
	employee-3	a	employee-3@a	
	assistant-3	a	assistant-3@a	
	secretary-2	a	secretary-2@a	
	assistant-1	a	assistant-1@a	

Showing 1 to 10 of 12 entries



Person Form

First Name

employee-2

Last Name

a

Email Address

employee-2@a

Roles

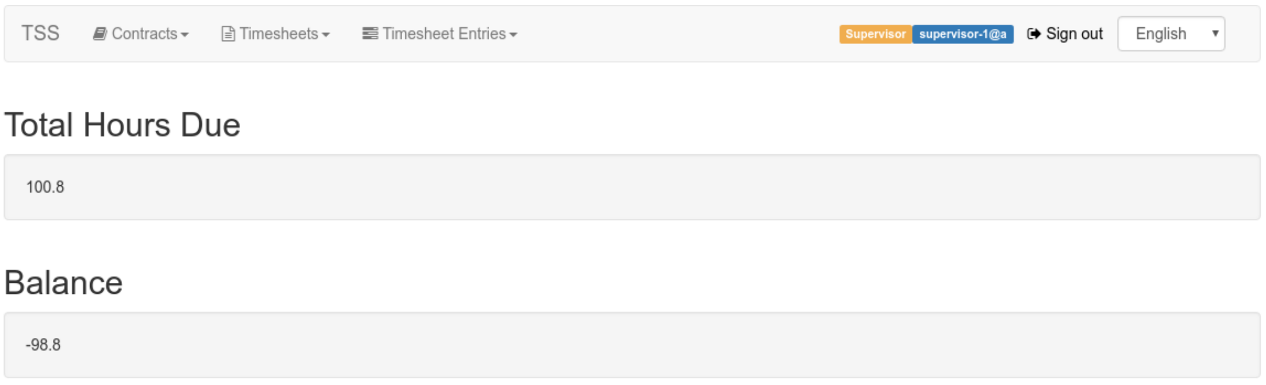
Employee

Date of Birth

Save

Delete

Total hours



Total Hours Due

100.8

Balance

-98.8

The problems and solutions

P1. Where could we start our project?

A1. Create JDBC resources, create entities and test the connection with DB

P2. Do we need to create each role entity class or just make role as an attribute of a person class?

A2. Yes. We need these different role entities, because each role has a different mapping relationship with the contract entity such as one-to-one, one-to-many, many-to-one.

P3. How to stay in touch during the holiday?

A3. Because we don't have a long break without lectures during August and September, it's difficult to sit down together to discuss our project. But our team members are active and use different ways to communicate online.

P4. How to deal with the problem: Entity name must be unique in a persistence unit?

A4. Because of the changes of version for one project, some new packages were created with new entities. New entities have the same name with the former entities, but we just want to use new entities. Even if delete the old entities source code, when deploy the project there is an error： Entity name must be unique in a persistence unit. Configure the persistence.xml file and just make the new entities included such as <class>New entity</class>

```
<exclude-unlisted-classes>true</exclude-unlisted-classes>
```

Timeline

06.2018	Set up team and find people
13.08.2018	Starting from the entities
20.08.2018	Simple “CRUD” operation
	Bugs fix
16.09.2018	Integration public Calendar
10.09.2018	Person\Contract Management and other roles
12.09.2018	Timesheet main function
22.09.2018	check other unfinished functions
25.09.2018	Bug fix
29.09.2018	Project manual

Time Spent

Description	**Hours spent**
Understanding Java EE. This included watching courses from Pluralsight and YouTube	50 hours
Going through the application provided from teacher.	9 hours
Working on application.	100 approx. (Around 20 days, 2-3 hours a day) including time for documentation

Subsystems in TSS Application

Following subsystems exists in TSS application.

1. **Contract and timesheet system**

   The major business functionalities are handled by this module.

   With majority of the functional requirements handled by this subsystem makes it an integral part of TSS application.

2. **Scheduler**

   The scheduler subsystem takes care of all scheduling related activities like archiving of timesheets after 2 years, scheduling of email reminders etc.

3. **Crud**

   This module does the job of some basic "CRUD" operations to test the connection with DB.

4. **Holiday system**

   Calculates public holidays in specified period.

5. **Person system**

   Keeps record (name, email, role, DOB) of employees, assistants, supervisors, secretary and administrators.

6. **Role system**

   Keeps track of roles information regarding employees, assistants, supervisors, secretary and administrators.

Glossary

Term	Explanation
Administrator	An Administrator is a university staff member. Administrators are responsible to install, configure, and operate the TSS.
Secretary	A secretary is a university staff member. Secretaries are responsible for printing the time sheets
Supervisor	A supervisor is a university staff member. The supervisor is the contractual boss of an employee.
Application server	Software framework that provides both facilities to create web applications and a server environment to run them.
Assistant	An assistant is a university staff member. Assistants are responsible for the concrete tasks assigned to employees.
Contract	A written or spoken agreement
Employee	An employee can be a student or a university staff member
Guest	Guests may only view public information and documentation about the TSS.