

# **Time Sheet System (TSS)**

JavaEE Web application course WS17.

## **Project Manual**

### **Team Yankee**

Biplov K.C

Madhu Rakhal Magar

Pradip Giri

Sabin Bhattarai

Shriharsh Ambhore

# Contents

Contents .....	2
Introduction.....	3
System Setup .....	4
Deploying Application on server.....	5
Launching the Application .....	12
Problems faced.....	13
Assumptions .....	16
Time Spent.....	17
Initial Project Planning .....	19
List of Completed and Missed Requirements.....	19
Subsystems in TSS Applications .....	24
<u>Using the application</u> .....	26
<u>Glossary</u> .....	41

# 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 have to 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) will keep track of all time sheet related business activities.

This user manual will provide instructions on how to access the TSS and installing it on Glassfish web application server.

# 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 of the client.

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

## 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 MYSQL database server is required to run TSS. The latest version of the MYSQL database server is available at <https://dev.mysql.com/downloads/installer/>.

### 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 ear

The latest version of the application in ear format can be obtained from the group svn repository.

## Server-side Hardware requirements

System with min 2 GB RAM.

# Deploying Application on server

For an application to run, it must be deployed on the application server like Payara to start serving to the user requests. But before that we must complete few steps to complete the process of installing the application on the server.

## Installing MYSQL

After Downloading the MYSQL database server open the file and follow the instructions to install the MYSQL server.

Create a database schema with name “**yankee\_tss**” and desired **username** and **password** to access it.

## Creating JDBC resource and JDBC Connection pools

After installing the payara server to the desired location, start the server domain by executing following command to start the domain of your choice.

```
$ {PAYARA_INSTALL_DIR}/bin/asadmin start-domain your_domain_name
```

```
asadmin> start-domain D1
Waiting for D1 to start .....
.....
Successfully started the domain : D1
domain Location: I:\software\payara-4.1.1.171.1\payara41\glassfish\domains\D1
Log File: I:\software\payara-4.1.1.171.1\payara41\glassfish\domains\D1\logs\server.log
Admin Port: 28933
Command start-domain executed successfully.
asadmin>
```

Once the server is up and running, navigate to URL <http://localhost:28933>. Please note that the admin port may vary. Refer to the port number displayed in the message after the server is started.

Navigate to the JDBC Connection pool option.

Click on New button to create a new JDBC connection pool. Fill in the details as show in the screenshot shown below.

## New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

### General Settings

Pool Name: \*

Resource Type:    
 Must be specified if the datasource class implements more than 1 of the interface.

Database Driver Vendor:    
 Select or enter a database driver vendor

Introspect: ☐ Enabled   
 If enabled, data source or driver implementation class names will enable introspection.

Click on next to mention the desired values for database username, password and the URL.

Pool Name: yankee\_tss

Additional Properties (3)			
<input checked="" type="checkbox"/>	<input type="text" value="Add Property"/>	<input type="text" value="Delete Properties"/>	
Select	Name	Value	Description
<input type="checkbox"/>	password	root	
<input type="checkbox"/>	user	root	
<input type="checkbox"/>	URL	jdbc:mysql://localhost:3306/yankee_tss?zeroD	

Click finish to save the Connection pool settings. Navigate to JDBC Resources to create a new Resource. Select the pool created in previous step.

### 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: \*

Pool Name:    
 Use the [JDBC Connection Pools](#) page to create new pools

Description:

Status: ☒ Enabled

Additional Properties (0)			
<input type="text" value="Add Property"/>	<input type="text" value="Delete Properties"/>		
Select	Name	Value	Description
No items found.			

Please note that a JDBC driver for MYSQL is required. This can be obtained from <https://dev.mysql.com/downloads/connector/j/>. The jar file obtained needs to be placed in side `${PAYARA_INSTALL_DIR} \payara41\glassfish\lib`

## Deploying the application

Before we start the deployment, we should ensure that the domain and the database server is up and running.

To see which domains are started execute the following command.

```
$ {PAYARA_INSTALL_DIR}/bin/asadmin list-domains
```

```
asadmin> list-domains
D1 running
domain1 not running
payaradomain not running
yankeeDomain running
Command list-domains executed successfully.
asadmin>
```

To deploy the EAR file, use the **deploy** option and mention the path to the application which needs to be deployed.

```
$ {PAYARA_INSTALL_DIR}/bin/asadmin deploy yankee_tss.ear
```

Please note above command will deploy in the default domain. If any other domain is selected then use following command

```
$ {PAYARA_INSTALL_DIR}/bin/asadmin deploy --port DomainPort --host localhost yankee_tss.ear
```

Where **DomainPort** represents the port number specified for the selected domain and localhost is the hostname.

```
asadmin> deploy --port 28933 --host localhost C:\\deploy\\yankee_TSS.ear
Deprecated syntax, instead use:
asadmin --port 28933 --host localhost deploy [options] ...
Application deployed with name yankee_TSS.
Command deploy executed successfully.
asadmin>
```

## Deploying Directory Lookup application

To support the working of TSS application, additional “DirectoryLookup” application needs to be deployed in the same domain in which the yankee\_TSS.ear file will be deployed. The latest version of the “DirectoryLookup.ear” can be obtained from the svn repository. To deploy execute the following command.

```
$ {PAYARA_INSTALL_DIR}/bin/asadmin deploy -port 28933 {path to application}
asadmin> deploy --port 28933 C:\\deploy\\DirectoryLookup.ear
Application deployed with name DirectoryLookup.
Command deploy executed successfully.
asadmin>
```

## Installing the file realm on server

To ensure that only authenticated users can access a security policy “realm” is defined. Users are assigned to user groups and are given specific roles.

To install UNI-KO file realm please follow below mentioned steps.

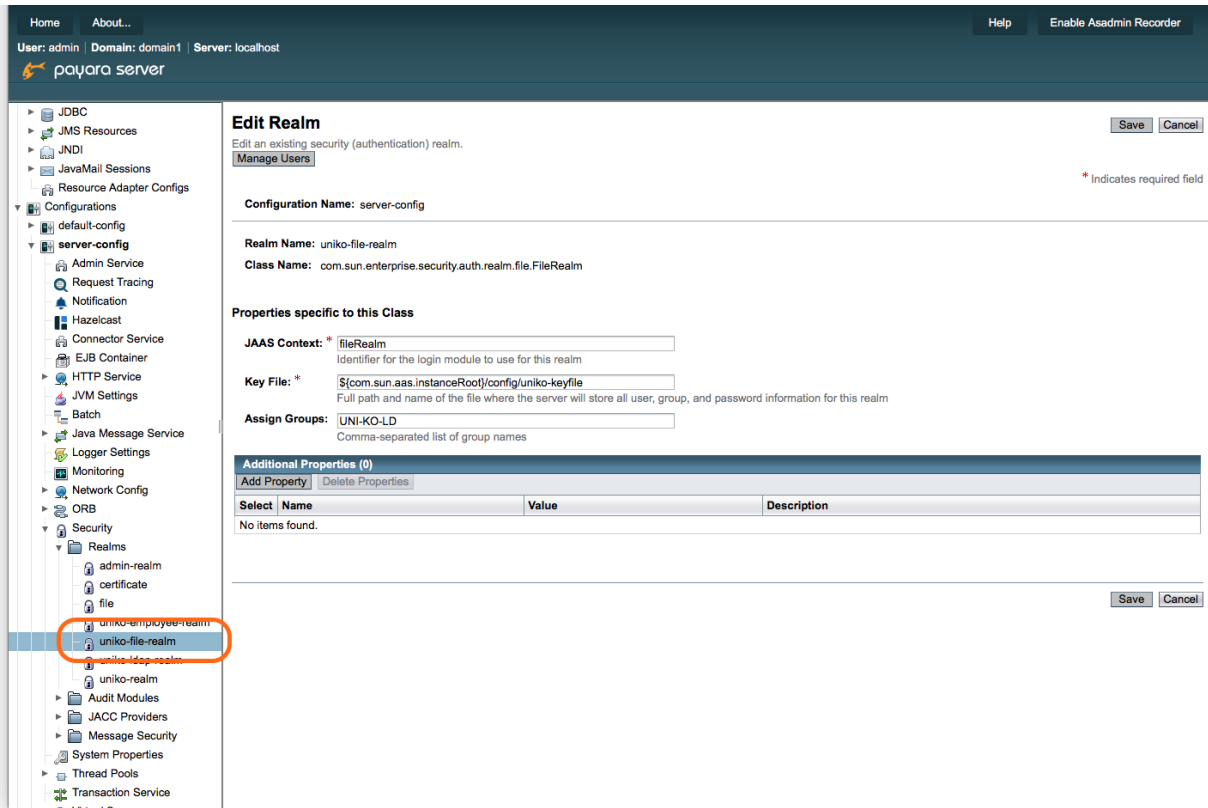
1. Copy the file "uniko-keyfile" to your Payara domain config folder.

In general, you'll find that folder from Payara's installation directory:

```
$ {PAYARA_INSTALL_DIR}/payara41/glassfish/domains/D1/config
```

2. Using the domain admin console, create a Security Realm by following below mentioned path. Configurations > server-config > security > Realms.



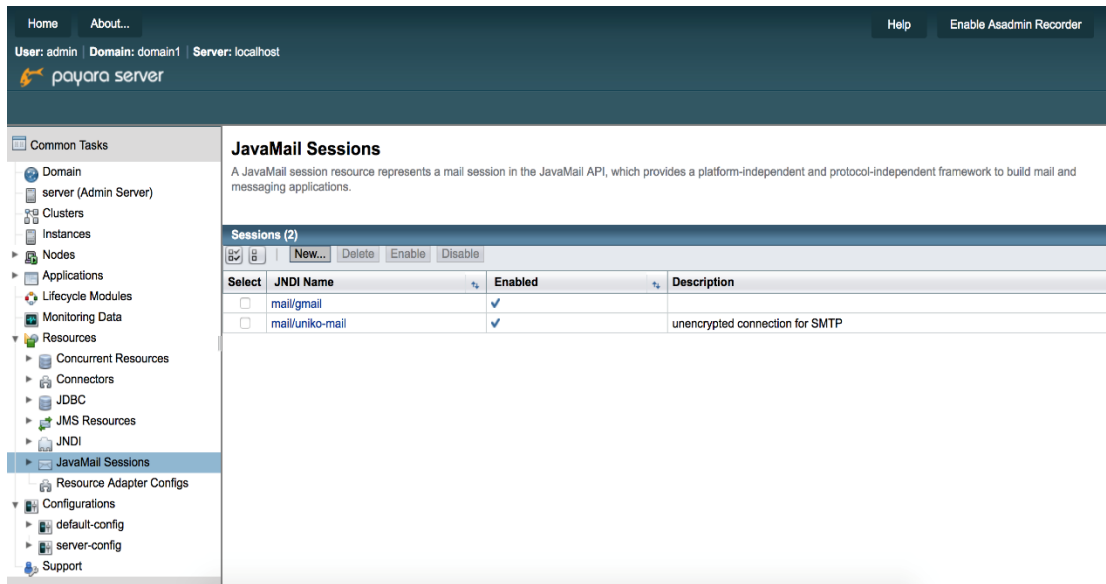


3. Restart your Payara server.

## Installation Guide of TestSMTP Server

To enable email service in the application we need to set up SMTP server to send email reminders. Please follow following steps to setup SMTP server.

1. Go to Payara Admin Console by right clicking to the Payara Server and click on “View Admin Console”. Before going there, make sure that Payara Server is running. (You can start the Payara Server from Project Explorer -> Services tab -> Servers -> Payara).
2. From Admin Console click on Resource Tab, after that click on JavaMail Sessions.



3. Click New Button and Fill the following information from the image below:

The screenshot shows the JBoss Administration Console interface. On the left is a 'Tree' view with a 'Common Tasks' section and a 'Resources' section. The 'Resources' section is expanded, showing 'JavaMail Sessions' and the selected session 'mail/uniko-mail'. The main area is titled 'Edit JavaMail Session' and contains the following configuration fields:

- JNDI Name:** mail/uniko-mail
- Mail Host:** localhost (DNS name of the default mail server)
- Default User:** nobody (User name to provide when connecting to a mail server; must contain only alphanumeric, underscore, dash, or dot characters)
- Default Sender Address:** nobody@localhost (E-mail address of the default user)
- Deployment Order:** 100 (Specifies the loading order of the resource at server startup. Lower numbers are loaded first.)
- Description:** Unencrypted connection to test SMTP server (Makes it easier to find this session later)
- Status:** ☒ Enabled

Below these fields is the 'Advanced' section:

- Store Protocol:** imap (Either IMAP or POP3; default is IMAP)
- Store Protocol Class:** com.sun.mail.imap.IMAPStore (Default is com.sun.mail.imap.IMAPStore)
- Transport Protocol:** smtp (Default is SMTP)
- Transport Protocol Class:** com.sun.mail.smtp.SMTPTransport (Default is com.sun.mail.smtp.SMTPTransport)
- Debug:** ☐ Enabled (Select the Debug checkbox to enable extra debugging output for this mail session, including a protocol trace.)

At the bottom is the 'Additional Properties (1)' section, which contains a table with one property:

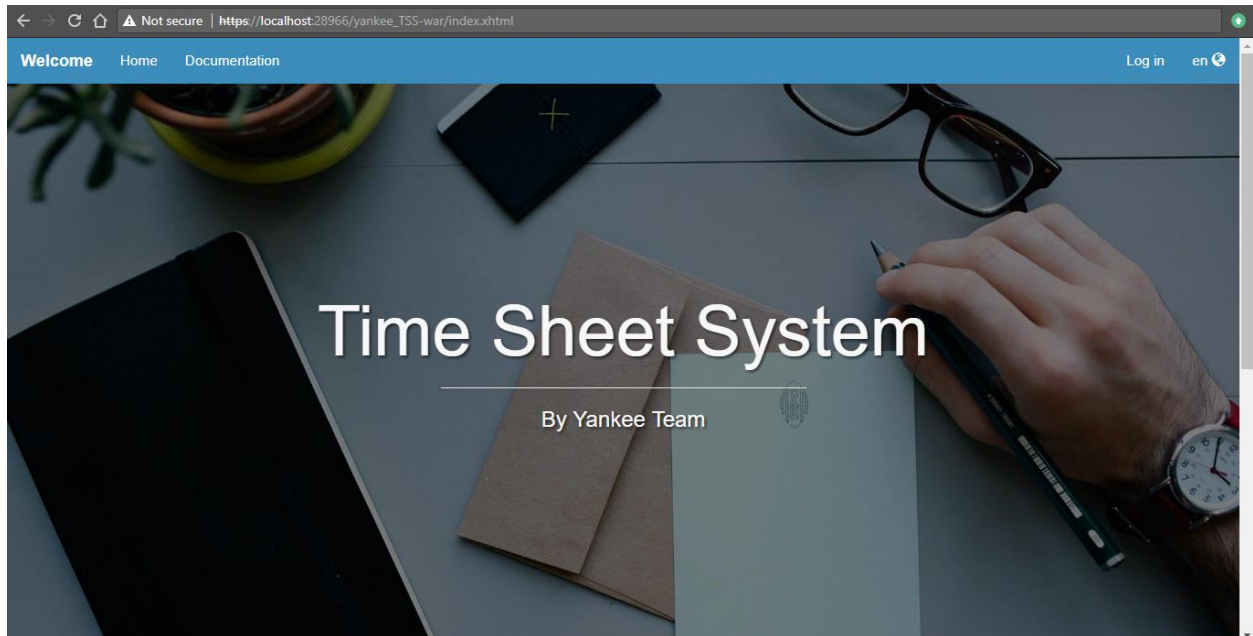
Select	Name	Value	Description
<input type="checkbox"/>	mail.smtp.port	2500	

4. Click save. Now you can see the listing of your JavaMail Session settings.
5. Deploy the TestSMTPServer.ear application on the same domain and the TSS application is ready to send emails.

# Launching the Application

Once the deployment process is complete you can access the application by entering following URL <http://localhost:28966/yankee/> . **Please note that the port number and the host name may vary.**

The application is up and running once the homepage of the application is displayed.



# Problems faced

## Unit Testing

Cannot start glassfish server error:

For testing we started with some simple application which did not include fetching data from database. This was just to get acquainted with Junit, creating test cases and working with basic features like assertEquals, assertTrue, assertFalse, etc.

Things worked fine for simple application. This sample application can be provided if need be. A screenshot from sample application for unit testing can be seen below:

```

8      public String getName() { return name; }
9
10     public boolean isHappy() {
11         return happy;
12     }
13
14     public void playWithRock() {
15         happy = true;
16     }
17 }
18
19 @Test
20 public void testHappyAfterPlay() throws Exception {
21     PetRock rocky = new PetRock(name: "rocky");
22     rocky.playWithRock();
23     assertTrue(rocky.isHappy());
24 }

```

Run PetRockTest

Test Case	Duration	Status
PetRockTest	4ms	Passed
testHappyAfterPlay	4ms	Passed
getName	0ms	Passed
testUnhappyAtStart	0ms	Passed

All 3 tests passed - 4ms

"C:\Program Files\Java\jdk1.8.0\_131\bin\java" ...

Process finished with exit code 0

However, for our application when it came to working with database we could not get it running. We kept getting error saying cannot start glassfish server. Even when the server was not running we kept getting same error.

Due to time limit we could not dive further into the problem. However, we have created test classes for Business logic and it exists in unit testing branch. The test classes include the test methods for business logic class which we think should work if we could solve the glassfish server problem.

## Tab View Refresh

Problem occurred when trying to refresh page with actions on tab view e.g. Sign Timesheet or Archive Timesheet using command button. When we implemented similar tab related tasks in few other pages, there seems to be no problem. Work around was performed on few pages using refresh window using JavaScript. It was also later realized that annotation @Postconstruct was causing issues on tabs to be refreshed. It was later resolved by not making use of such annotation. A research on code optimization will be realized in next version.

## JSF Communication Issue

During Contract creation we were to achieve a functionality of being able to change supervisor on behalf of another staff. This was to be achieved when a person as employee is selected from the drop down, a list of applicable supervisor for that employee were to be obtained. This meant Ajax rendering of content which contained another form. The issue was not solved as we figured it was a JSF spec issue 790. The issue is detailed in following link.

<http://balusc.omnifaces.org/2011/09/communication-in-jsf-20.html#AjaxRenderingOfContentWhichContainsAnotherForm>

## JDBC Ping Issue

We faced issues when trying to setup the database using the jdbc pool and jdbc resource as described above in system setup. Some of us could ping to the database and obtain successful connection. But few of us had issued connecting.

Solution mentioned below solved the issue after few research using google.

Firstly we located the domain.xml file in the payara41 folder inside our domain, which for us was - for example in following location

*C:\Users\Sabs\GlassFish\_Server\payara41\glassfish\domains\tss\_domain\config\domain.xml*

We had to make sure the following existed in the following xml file. We were missing value as localhost "jdbc:mysql://localhost:3306/yankee\_tss"

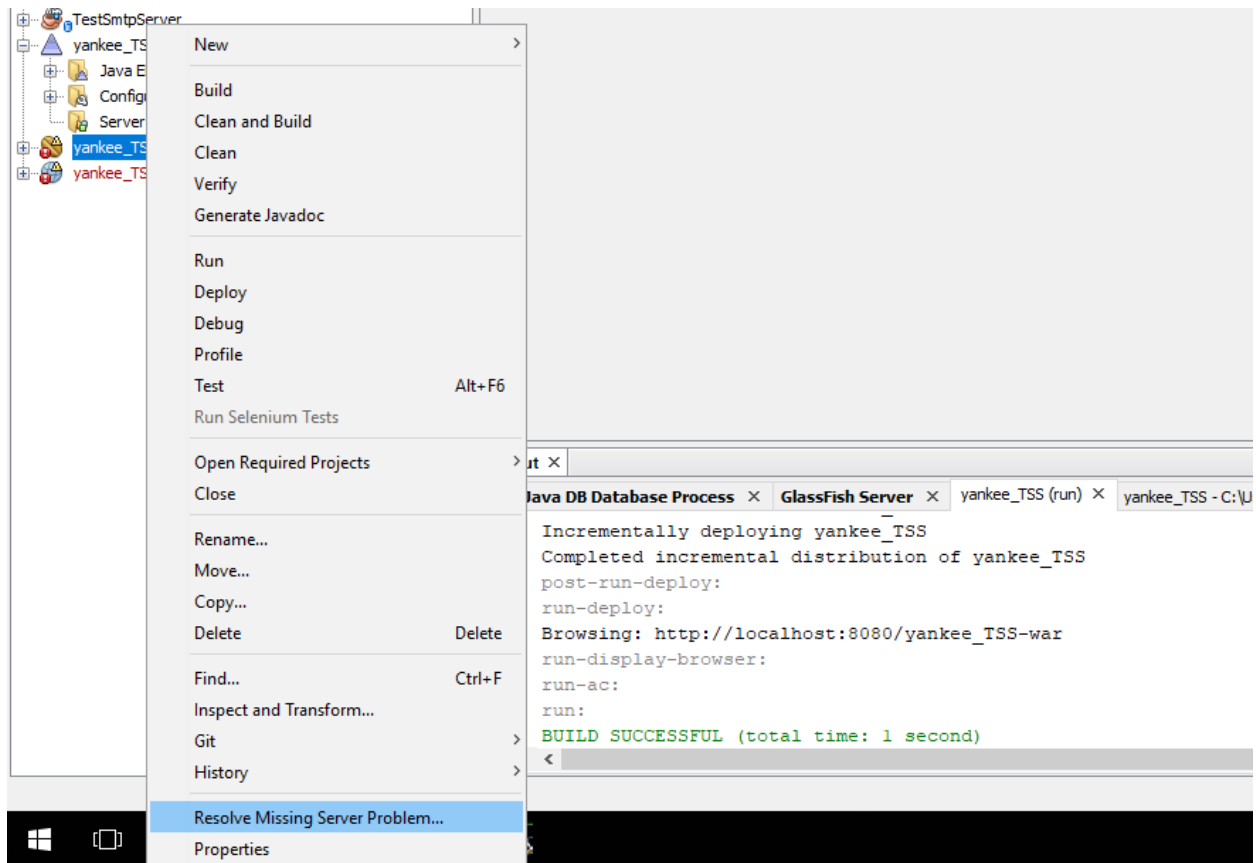
```
<jdbc-connection-pool driver-classname="com.mysql.jdbc.Driver" name="yankee_tss"
res-type="java.sql.Driver">
  <property name="password" value="root"></property>
  <property name="user" value="root"></property>
  <property
value="jdbc:mysql://localhost:3306/yankee_tss"></property>
</jdbc-connection-pool>
```

*name="URL"*

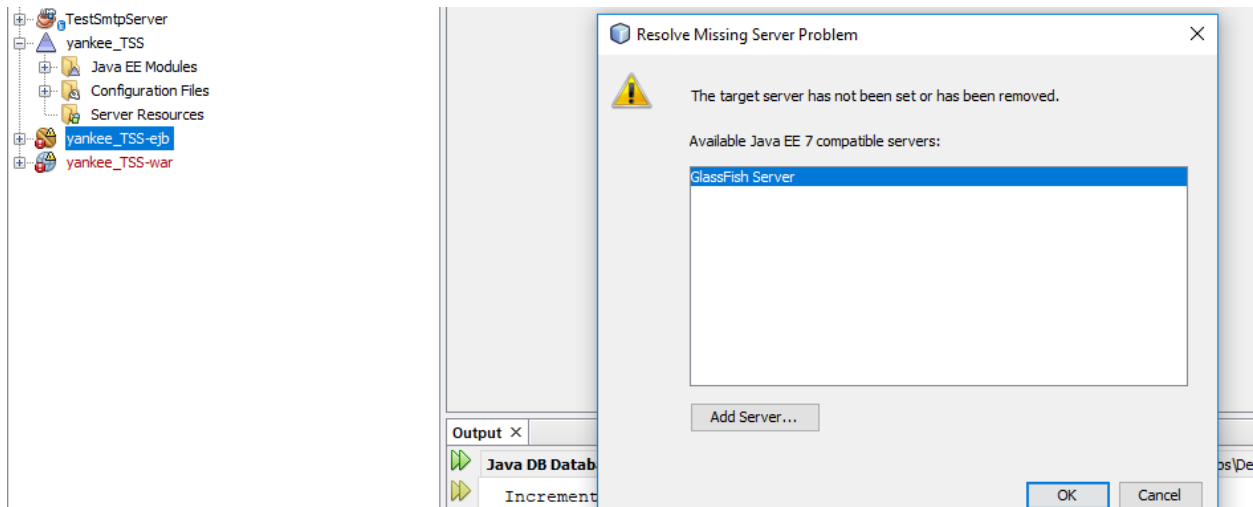
## Missing Server Problem

Due to .private and .properties files created by NetBeans during build, we ran into problems where certain libraries were not detected and error were thrown when we tried to pull the pushed code in via GitHub. We found a solution to exclude those properties files from git using .gitignore file. However we kept on getting the problem with properties files missing. Thus the following had to be done when pulling from git.

Right click on the project file and click on Resolve missing Server Problem



Next is to select the Glassfish server from the list as shown below.



# Assumptions

We made few assumptions during the project implementation as described below.

## Working Days Per Week

Assumption was made during hours due calculation where public holidays were to be considered. The calculation involved use of working hours per week which user could edit during contract creation. The assumption made was when a user enters working days per week as 3, the days taken were Monday Tuesday and Wednesday. If working days per week was given as 2, the days taken were Monday and Tuesday. Meaning the working days started from Monday.



# Time Spent

Following are some of the course and videos followed prior to project implementation.

## **Pluralsight course**

Java EE : Getting started with Javasever Faces – by Jesper De Jong.

Java EE 7: The fundamentals – by Antonio Goncalves

Java Persistence API 2.1 – by Antonio Goncalves

## **Biplov K.C.**

Description	Hours spent
Understanding Java EE. This included watching courses from Pluralsight and YouTube	40 approx.
Going through the application provided from teacher.	8 approx.
Working on application.	50 approx. (Around 20 days, 2-3 hours a day)

## **Pradip Giri.**

Description	Hours spent
Studying java scheduler	8 approx.
Implementing reminder service	24 hrs
Deleting Archive Timesheet	8 hrs
Research on finding public holiday on Germany	8 hrs

**Sabin Bhattarai.**

Description	Hours spent
Understanding Java EE. This included watching courses from Pluralsight and YouTube	45 approx. (inclusive with working on application)
Going through the application provided from teacher.	12 approx.
Working on application.	112 approx. (Beginning 25 days, 4 – 5 hours a day)  100 approx. (Last 10 Days (8 - 12 hours a day))

**Madhu Rakhal Magar.**

(22 \* 5 hour working on application + 50 hour study and research.)

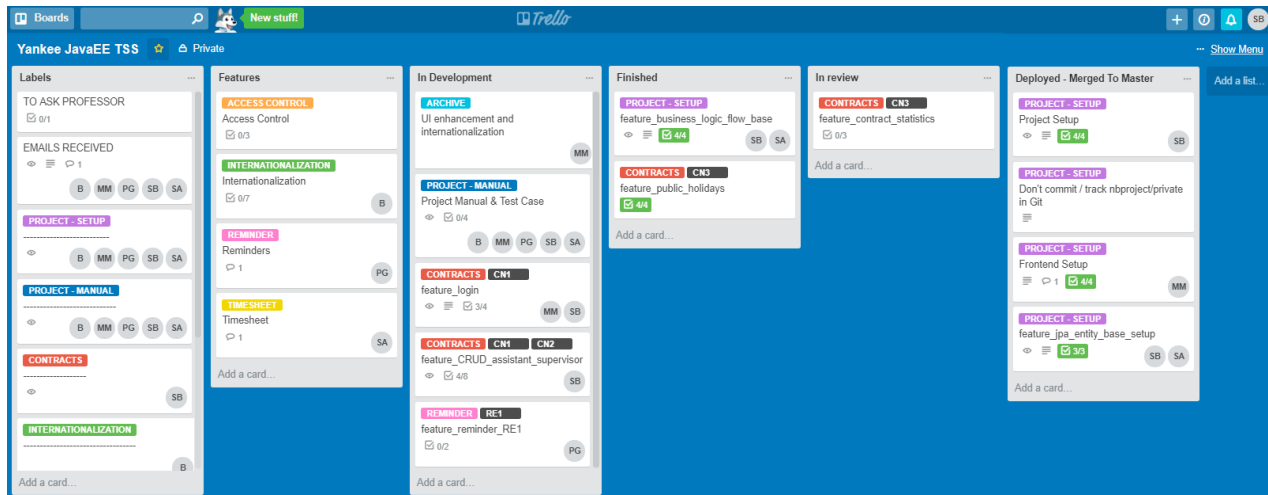
**Shriharsh Ambhore.**

(40hr study and research + 20 (days) \* 4 hours)

# Initial Project Planning

Project planning was done using **Trello** where all requirements were listed and divide among the group.

A screenshot of the initial plan is shown below.



## 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
CN2	Ability to print contracts	Completed
CN3	View contract Statistics	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	Completed

	hours due of individual timesheet	
CN4c	Calculation of hours due	Completed
CN4d	Determine the public holidays in Rheinland-Platinat	Completed
CN4e	Determine the public holidays in Germany from 01-January-2017 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
CN6a	Ensure that start date and end date of a contract always denote complete months	Completed
CN7	Provide assistants and supervisors with the ability to start a contract.	Completed
CN8	Provide assistants and supervisors with the ability to terminate a started contract	Completed
CN9	Ensure that only contracts can be aborted whose time sheets are in status SIGNED_BY_SUPERVISOR or IN_PROGRESS.	Completed
CN10	Record the date of termination of a contract	Completed
CN11	Warn the user if there are time sheets in state IN_PROGRESS that have entries, before terminating the contract	Completed
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.

<b>Requirement No</b>	<b>Short Description</b>	<b>Status</b>
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
TS4	Don't delete time sheets that are in the SIGNED_BY_EMPLOYEE state	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

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

<b>Requirement No</b>	<b>Short Description</b>	<b>Status</b>
SG1	Provide employees with the ability to sign a time sheet	Completed
SG1a	Provide employees with the ability to revoke signatures	Completed
SG2	Provide supervisors with the ability to sign a time sheet that is in status SIGNED_BY_EMPLOYEE	Completed
SG3	Ability to request changes to a time sheet that is in status SIGNED_BY_EMPLOYEE	Completed
SG4	Ability to digitally sign the time sheets	Incomplete

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

<b>Requirement No</b>	<b>Short Description</b>	<b>Status</b>
RE1	Send reminder mail to employee on last day of timesheet if time sheet is in IN_PROGRESS state	Complete
RE2	Send reminder mail to supervisor and assistant if time sheet is signed by employee	Complete
RE3	Send a reminder mail to the secretaries if the time sheet is in state SIGNED_BY_SUPERVISOR.	Complete
RE4	Repeat reminders every day.	Complete
RE5	Collect all reminders so that a person receives at most one e-mail per day.	Complete

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

<b>Requirement No</b>	<b>Short Description</b>	<b>Status</b>
AR1	Ability to archive time sheets that are in status SIGNED_BY_SUPERVISOR	Completed
AR2	ensure that archived time sheets can not be changed	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**.

<b>Requirement No</b>	<b>Short Description</b>	<b>Status</b>
AC1	authenticate users prior to giving access to any data	Completed
AC2	determine whether a person is university staff member	Completed
AC3	Allow only authorized users to manipulate data	Completed

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

Requirement No	Short Description	Status
UI1	Support mobile devices	Completed
UI2	Support different browsers	Completed
UI3	Use Bootstrap CSS library	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	Incomplete
PM5	Record time spent on project	Complete

# Subsystems in TSS Application

Following subsystems exists in TSS application.

1. Contract and timesheet subsystem.  
The core business functionalities are handled by this subsystem.  
With majority of the functional requirements handled by this subsystem makes it an integral part of TSS application.
2. Email reminder subsystem  
The email reminder subsystem is responsible for sending email reminder based on the configuration approved by the system administrator.
3. Scheduler subsystem.  
The scheduler subsystem is responsible for all scheduling related activities like archiving of timesheets after 2 years, scheduling of email reminders etc.



# Documentation

# Using the application

**Note:** When initially deployed, admin of the system should perform the loading public holidays task.

Initially the login is performed by admin and through Dashboard (shown in later steps) he/she will load all public holidays as shown below.

The image shows two screenshots of a web application. The top screenshot is a 'Sign in' form with fields for 'Email' (containing 'admin@uni-koblenz.de') and 'Password' (masked with dots). Below the password field is a checkbox labeled 'I accept terms and condition. \*' and a text box containing the disclaimer: '\*) I hereby declare that Time Sheet will store and use my personal information for official use only.' At the bottom of the form are two buttons: 'Login As Guest' and 'Log in'. The bottom screenshot shows the application's dashboard. At the top right is a breadcrumb trail 'Home > Dashboard'. Below this is a section titled 'Administration' containing a blue button 'Load Public Holidays To Database'. Further down is a section titled 'Admin Administrator' containing two buttons: 'My contracts' and 'Statistics'.

Button namely Load Public Holidays To Database is to be pressed prior to use of application. **Note** it takes few seconds to perform the task.

Similarly, Admin after loading application has access to choose state and also decide on whether to switch of Reminder service as shown below.

The image shows a screenshot of the 'Administration' settings page. It features a dropdown menu labeled 'Select German state' with 'RHINELANDPALATINATE' selected. Below this is a toggle switch for 'Set Reminder Service' which is currently set to 'OFF'. At the bottom right of the page is a green 'Update' button.

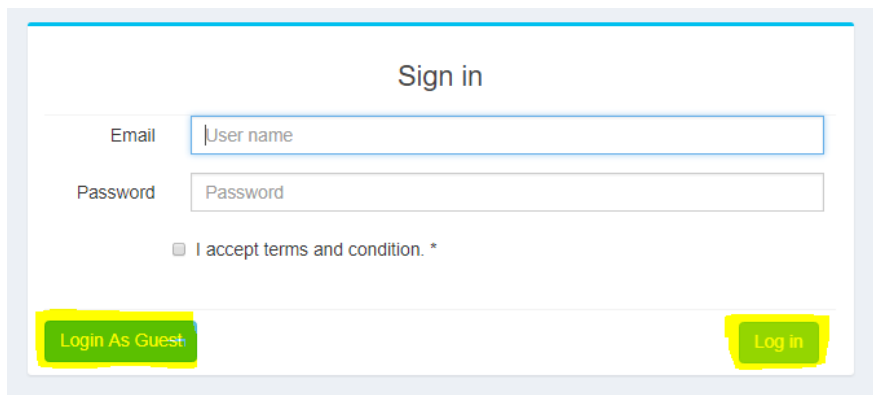
Now Let us delve into all the allowed actions and basic set of business activities that can be performed by the logged in user.

The view for the user will vary as per the roles assigned to that user.

## 1. Login

Application allows login to users listed in the realm file. No registration allowed. Login page appears when clicked on Login on top right of the main page shown while launching the application. There are two options with Guest Login as well as user Login.

Guest Login can only view documentation while the user login gets all the application features that will be mentioned below.

A screenshot of a web application's sign-in page. The page has a light blue border. At the top center, the text "Sign in" is displayed. Below this, there are two input fields: the first is labeled "Email" and contains the placeholder text "User name"; the second is labeled "Password" and contains the placeholder text "Password". Below the password field, there is a checkbox with the text "I accept terms and condition. \*". At the bottom of the form, there are two buttons: "Login As Guest" on the left and "Log in" on the right. Both buttons are highlighted with a yellow rectangular box.

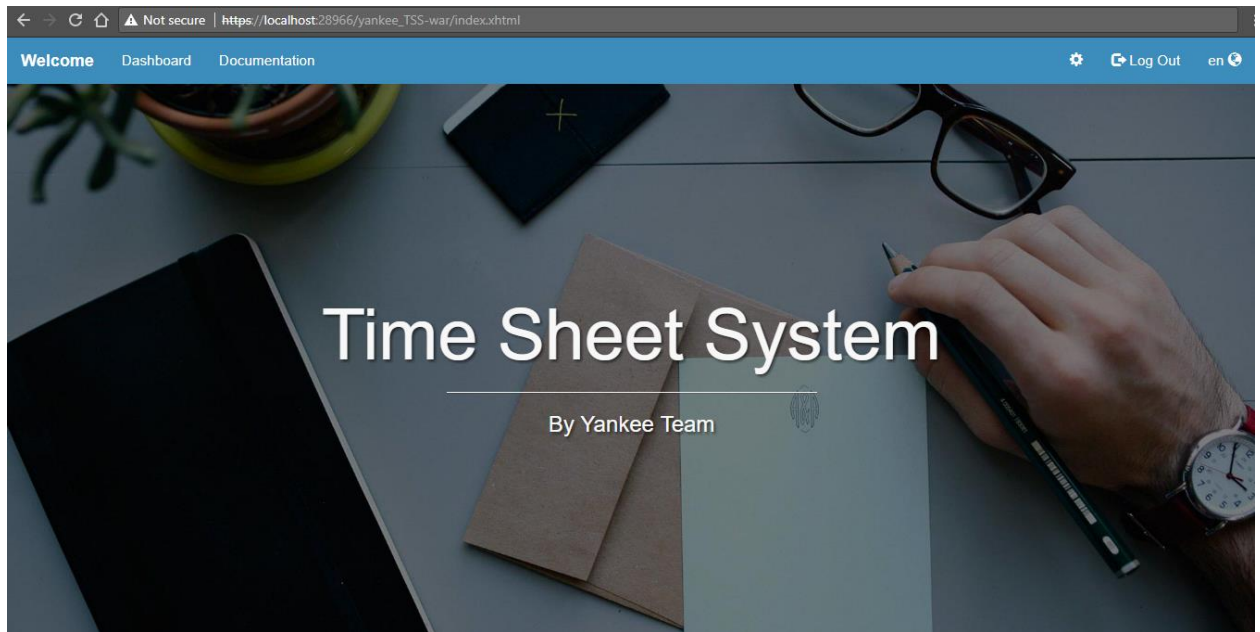
## 2. Create contract

As per the requirements a contract can be created by a person who has a role of Supervisor, Secretary or Assistant. i.e. STAFF.

Steps to create a contract are mentioned below.

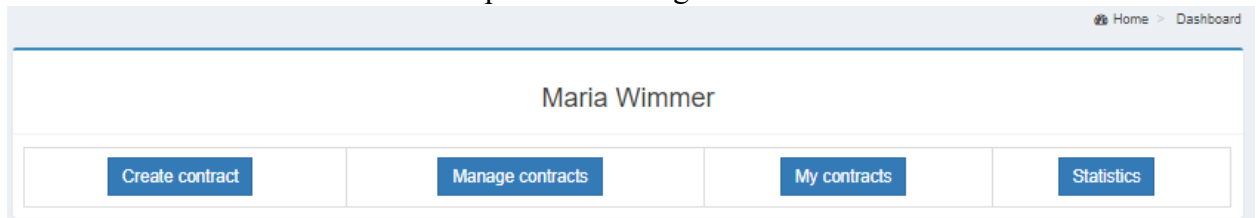
### Step 1:

To create a contract a person with role STAFF must log in with correct credentials. On successful validation home page is displayed to user as shown in below screenshot.



Step 2:

Click on the “Dashboard” button to open the “Manage contract” view.



Step 3:

Click on the “Create Contract” button to view the Create Contract form.

Home > Dashboard > Create contract

### Create Contract For

Select Person

On Behalf of Supervisor

First name

Last name

Email

Working days per week

Vacation days per year

Hours per week

Archive Duration

Time sheet frequency

Contract start date

Contract end date

[Cancel](#) [Create contract](#)

Following fields are mandatory to create contract.

Select Person- List of University staff members for whom the contract needs to be created.

Hours per week- Hours to work per week.

Time Sheet Frequency- Denotes the frequency of time sheets. It can be either weekly or monthly.

Contract Start Date- Denotes the start date of the contract.

Contract End Date- Denotes the end date of the contract.

Click “Create Contract” button to create the contract.

### 3. Manage contract

Manage contract screen shows the list of Contract that logged in user is assigned to as either a Supervisor or as Secretary or as Assistant. The Terminated Contract is also viewable in the tabs shown.

Home > Dashboard > Manage contracts

As supervisor As assistant As Secretary Terminated

Maria Wimmer (As Supervisor)

Show 10 entries Search:

Employee	Status	Action
Sabin Bhattarai (sbhattarai@uni-koblenz.de)	PREPARED	<a href="#">▶</a> <a href="#">▶</a> <a href="#">▶</a>

Showing 1 to 1 of 1 entries

Previous 1 Next

For each contract prepared following action buttons are enabled.

Start Contract- On clicking this button the selected contract is started. Once started no changes can be done to the contract. Timesheets are created at this stage.

View Contract- On clicking this button the details of selected contract are displayed as shown.

Home > Dashboard > Employee Contracts > View contract details

Contract for : Sabin Bhattarai (sbhattarai@uni-koblenz.de)

Supervisor	Maria Wimmer	Hours per week	20.0
Contract start date	1 OCTOBER 2017 (SUNDAY)	Total Hours Due	80.0
Contract end date	31 OCTOBER 2017 (TUESDAY)	Vacation Hours	6.67
Time sheet frequency	WEEKLY		

Current Previous Signed by employee Archived All

Show 10 entries Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	IN_PROGRESS	<a href="#">▶</a>

Showing 1 to 1 of 1 entries

Previous 1 Next

Edit Contract- On clicking this button the details entered for contract can be edited as shown where assistants can be added, supervisor can be changed, secretaries added.

Contract For : Anne Brucker (abrucker@uni-koblenz.de)

Contract start date

Contract end date

Time sheet frequency

Working days per week

Vacation days per year

Hours per week

Archive Duration

Change Supervisor

Edit assistants

☐ Silke Allmann

Current assistants

Edit secretaries

☐ Silke Allmann

Delete Contract- On clicking this button the selected contract can be deleted only in prepared state. As no delete is allowed after Started as shown below.

As supervisor As assistant As Secretary Terminated

Maria Wimmer (As Supervisor)

Show  entries Search:

Employee	Status	Action
Sabin Bhattarai (sbhattarai@uni-koblenz.de)	STARTED	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Anne Brucker (abrucker@uni-koblenz.de)	PREPARED	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Showing 1 to 2 of 2 entries

Previous **1** Next

Confirmation

Confirm delete?

#### 4. Adding time sheet entry

When viewing the contract details as shown in view contracts above. You have an option to view timesheet entries for that contract as shown.

Home > Dashboard > Employee Contracts > View contract details

Contract for : Sabin Bhattarai (sbhattarai@uni-koblenz.de)

Supervisor	Maria Wimmer	Hours per week	20.0
Contract start date	1 OCTOBER 2017 (SUNDAY)	Total Hours Due	80.0
Contract end date	31 OCTOBER 2017 (TUESDAY)	Vacation Hours	6.67
Time sheet frequency	WEEKLY		

Current Previous Signed by employee Archived All

Show 10 entries Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	IN_PROGRESS	

Showing 1 to 1 of 1 entries

Previous 1 Next

Note that we are looking this under Staff who is not allowed to add entries for the employee. So when logged in as employee the similar page appears with further possible action. Thus we will now delve into Employee view when they login.

Sabin Bhattarai

My contracts Statistics

Can now view my contracts and following appears

Sabin Bhattarai - As Employee

Started Prepared To sign time sheets

Show 10 entries Search:

Supervisor	Start date	End date	Frequency	Statistics
Maria Wimmer (wimmer@uni-koblenz.de)	2017-10-01	2017-10-31	WEEKLY	View

Showing 1 to 1 of 1 entries

Previous 1 Next



When View button pressed. He/She will be able to perform timesheet entries. i.e. further actions are introduced as shown.


Home > Dashboard > Employee Contracts > View contract details

Contract for : Sabin Bhattarai (sbhattarai@uni-koblenz.de)

Supervisor	Maria Wimmer	Hours per week	20.0
Contract start date	1 OCTOBER 2017 (SUNDAY)	Total Hours Due	80.0
Contract end date	31 OCTOBER 2017 (TUESDAY)	Vacation Hours	6.67
Time sheet frequency	WEEKLY		

Current Previous Signed By Supervisor

Show 10 entries Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	IN_PROGRESS	

Showing 1 to 1 of 1 entries

Previous 1 Next

As you can see the Sign timesheet action, view timesheet entries action, and add entry action appears.

Time sheet entry can be added by an employee for whom the contract has been created and the contract is in “STARTED” state.

The contract is visible under Home > Dashboard > Your Contracts.

A Time sheet is created based on the **Start** and **End date** of the contract and as per the **Time sheet frequency** mentioned in the contract.

An entry can be added to log the working hours by clicking the “Add Timesheet Entry” action button for the contract.

Home > Dashboard > Employee Contracts > View Contract Details > Edit Entries

Time Sheet entries for : 2017-10-01 - 2017-10-31

2017-10-01 Logged Hours: 0.0 +	2017-10-02 Logged Hours: 0.0 +	2017-10-03 Logged Hours: 0.0 +	2017-10-04 Logged Hours: 0.0 +	2017-10-05 Logged Hours: 0.0 +	2017-10-06 Logged Hours: 0.0 +	2017-10-07 Logged Hours: 0.0 +
2017-10-08 Logged Hours: 0.0 +	2017-10-09 Logged Hours: 0.0 +	2017-10-10 Logged Hours: 0.0 +	2017-10-11 Logged Hours: 0.0 +	2017-10-12 Logged Hours: 0.0 +	2017-10-13 Logged Hours: 0.0 +	2017-10-14 Logged Hours: 0.0 +
2017-10-15 Logged Hours: 0.0 +	2017-10-16 Logged Hours: 0.0 +	2017-10-17 Logged Hours: 0.0 +	2017-10-18 Logged Hours: 0.0 +	2017-10-19 Logged Hours: 0.0 +	2017-10-20 Logged Hours: 0.0 +	2017-10-21 Logged Hours: 0.0 +
2017-10-22 Logged Hours: 0.0 +	2017-10-23 Logged Hours: 0.0 +	2017-10-24 Logged Hours: 0.0 +	2017-10-25 Logged Hours: 0.0 +	2017-10-26 Logged Hours: 0.0 +	2017-10-27 Logged Hours: 0.0 +	2017-10-28 Logged Hours: 0.0 +
2017-10-29 Logged Hours: 0.0 +	2017-10-30 Logged Hours: 0.0 +	2017-10-31 Logged Hours: 0.0 +				

Timesheet Entry for: 2017-10-02

Start time: 10:00

End time: 15:00

Description: Project setup

Save Reset

## 5. Sign Time sheet.

Once all time sheet entries are added the user can choose to Sign the time sheet on the click of “Sign timesheet” action button.

Welcome Dashboard Documentation Anja Log Out en

Home > Dashboard > Employee Contracts > View Contract Details

Contract For : Anja Voigt (aanja@uni-koblenz.de)

Supervisor	Maria Wimmer
Contract Start Date	1 OCTOBER 2017 (SUNDAY)
Contract End Date	31 OCTOBER 2017 (TUESDAY)
Time Sheet Frequency	WEEKLY
Hours Per Week	35.0
Total Hours Due	140.0
Vacation Hours	11.67

Current Previous Signed By Supervisor

Show 25 entries Search:

Start Date	End Date	Hours Due	Hours Entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	35.0	0.0	IN_PROGRESS	Sign TimeSheet

Showing 1 to 1 of 1 entries Previous 1 Next

## 6. View entered Time sheet entries

Employee, Supervisor, Assistant and Secretary can view the time sheet entries entered by the employee.

Home > Dashboard > Manage Contracts > View Contract Details > View Entries

Show  entries Search:

Date	Hours	Description	StartDateTime	EndDateTime
2017-10-01	0.0			
2017-10-02	9.0	Java User interface design.	06:00:00	15:00:00
2017-10-03	0.0			
2017-10-04	0.0			
2017-10-05	0.0			
2017-10-06	0.0			
2017-10-07	0.0			
2017-10-08	0.0			
2017-10-09	0.0			
2017-10-10	0.0			

Showing 1 to 10 of 31 entries


Previous **1** 2 3 4 Next

## 7. Secretaries can print contract

Home > Dashboard > Manage contracts

As supervisor As assistant **As Secretary** Terminated

Show  entries Search:

Employee	Supervisor	Status	Action
Anne Brucker (abrucker@uni-koblenz.de)	Maria Wimmer	STARTED	

Showing 1 to 1 of 1 entries

Previous **1** Next

Following print view shows

**Employment Contract**

---

This Employment Contract states the terms and conditions that govern the contractual agreement between  
**Supervisor : Maria Wimmer**  
 having its principal place of employment at [University Koblenz Landau], and  
**Employee : [Anne Brucker]**  
 who agrees to be bound by this Contract.

---

NOW, Therefore, in consideration of the mutual covenants and promises made by the parties hereto, the Company and the Temp (individually, each a "Party" and collectively, the "Parties") covenant and agree as follows:

**TERM.** The term of this Temporary Employment Contract shall commence on **1 OCTOBER 2017 (SUNDAY)** and end on **31 OCTOBER 2017 (TUESDAY)**.

**TIMESHEET FREQUENCY.** The time sheets frequency is set to **WEEKLY**

**SECRETARIES.** The Contract will be bound to have following member as secretary  
**Silke Allmann**

**HOURS OF WORK.** The hours of work shall be on working days except public holidays at hours determined by the employer provided that ordinary working hours shall not exceed **20.0** hours per week.

---

IN WITNESS WHEREOF, each of the Parties has executed this Temporary Employment Contract, both Parties by its duly authorized officer, as of the day and year set forth below.

\_\_\_\_\_  
Supervisor SIGN/DATE

\_\_\_\_\_  
Employee SIGN/DATE

[Print](#)

## 8. Secretaries can print timesheet

**Contract for : Anne Brucker (abrucker@uni-koblenz.de)**

---

Supervisor	Maria Wimmer	Secretaries	-> Silke Allmann
Contract start date	1 OCTOBER 2017 (SUNDAY)	Hours per week	20.0
Contract end date	31 OCTOBER 2017 (TUESDAY)	Total Hours Due	80.0
Time sheet frequency	WEEKLY	Vacation Hours	6.67

[Current](#)
[Previous](#)
[Signed By Supervisor](#)
[Archived](#)
[All](#)

Show 10 entries

Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	IN_PROGRESS	

Showing 1 to 1 of 1 entries

[Previous](#)
1
[Next](#)

The following print preview for timesheet appears

Home > Dashboard > Manage contracts > View contract details > Print time sheet

---

**TimeSheet For : Anne Brucker (abrucker@uni-koblenz.de)**

---

Supervisor	Maria Wimmer	Hours per week	20.0
Time sheet frequency	WEEKLY	Total Hours Due	20.0
		Total Hours Entered	0.0

---

Show  entries Search:

Date	Hours	Description	StartDateTime	EndDateTime
2017-10-14	0.0			
2017-10-15	0.0			
2017-10-16	0.0			
2017-10-17	0.0			
2017-10-18	0.0			
2017-10-19	0.0			
2017-10-20	0.0			

Showing 1 to 7 of 7 entries Previous **1** Next

[Print](#)

## 9. Sign timesheets By Employee

Prompts user to sign in and when signed in.

**Contract for : Anne Brucker (abrucker@uni-koblenz.de)**

---

Supervisor	Maria Wimmer	Secretaries	-> Silke Allmann
Contract start date	1 OCTOBER 2017 (SUNDAY)	Hours per week	20.0
Contract end date	31 OCTOBER 2017 (TUESDAY)	Total Hours Due	80.0
Time sheet frequency	WEEKLY	Vacation Hours	6.67

---

Current Previous Signed By Supervisor

**Confirmation** ✕

⚠ Confirm sign?

Yes No

Show  entries Search:

Start date	End date	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	IN_PROGRESS <div style="float: right; text-align: center;"> <div style="background-color: #007bff; color: white; padding: 2px 5px;">+</div> <div style="background-color: #6c757d; color: white; padding: 2px 5px;">-</div> </div>

Showing 1 to 1 of 1 entries Previous **1** Next

Status changes to Signed by Employee and further allows supervisor to either revoke

Contract for : Anne Brucker (abrucker@uni-koblenz.de)

Supervisor **Maria Wimmer** Secretaries -> Silke Allmann


Contract start date **1 OCTOBER 2017 (SUNDAY)** Hours per week **20.0**

Contract end date **31 OCTOBER 2017 (TUESDAY)** Total Hours Due **80.0**

Time sheet frequency **WEEKLY** Vacation Hours **6.67**

Current Previous Signed by employee Archived

Show 10 entries Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	SIGNED_BY_EMPLOYEE	

Showing 1 to 1 of 1 entries Previous 1 Next

Revoke

⚠ Confirm revoke signature?

Yes No

or approve the timesheet entries by signing the timesheet as Supervisor.

Contract for : Anne Brucker (abrucker@uni-koblenz.de)

Supervisor **Maria Wimmer** Secretaries -> Silke Allmann


Contract start date **1 OCTOBER 2017 (SUNDAY)** Hours per week **20.0**

Contract end date **31 OCTOBER 2017 (TUESDAY)** Total Hours Due **80.0**

Time sheet frequency **WEEKLY** Vacation Hours **6.67**

Current Previous Signed by employee Archived All

Show 10 entries Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	SIGNED_BY_SUPERVISOR	

Showing 1 to 1 of 1 entries Previous 1 Next

And the status changes to Signed by supervisor.

Now the secretary will be able to archive the timesheet

Contract for : Anne Brucker (abrucker@uni-koblenz.de)

Supervisor	Maria Wimmer	Secretaries	-> Silke Allmann
Contract start date	1 OCTOBER 2017 (SUNDAY)	Hours per week	20.0
Contract end date	31 OCTOBER 2017 (TUESDAY)	Total Hours Due	80.0
Time sheet frequency	WEEKLY	Vacation Hours	6.67

Current

Previous




Signed By Supervisor

Archived

All

Show 10 entries

Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	SIGNED_BY_SUPERVISOR	  

Showing 1 to 1 of 1 entries

Previous

1

Next

Which when performed changes the status to Archived.

Contract for : Anne Brucker (abrucker@uni-koblenz.de)

Supervisor	Maria Wimmer	Secretaries	-> Silke Allmann
Contract start date	1 OCTOBER 2017 (SUNDAY)	Hours per week	20.0
Contract end date	31 OCTOBER 2017 (TUESDAY)	Total Hours Due	80.0
Time sheet frequency	WEEKLY	Vacation Hours	6.67

Current

Previous




Signed By Supervisor

Archived

All

Show 10 entries

Search:

Start date	End date	Hours due	Hours entered	Status	Action
14 OCTOBER 2017 (SATURDAY)	20 OCTOBER 2017 (FRIDAY)	20.0	0.0	ARCHIVED	  

Showing 1 to 1 of 1 entries

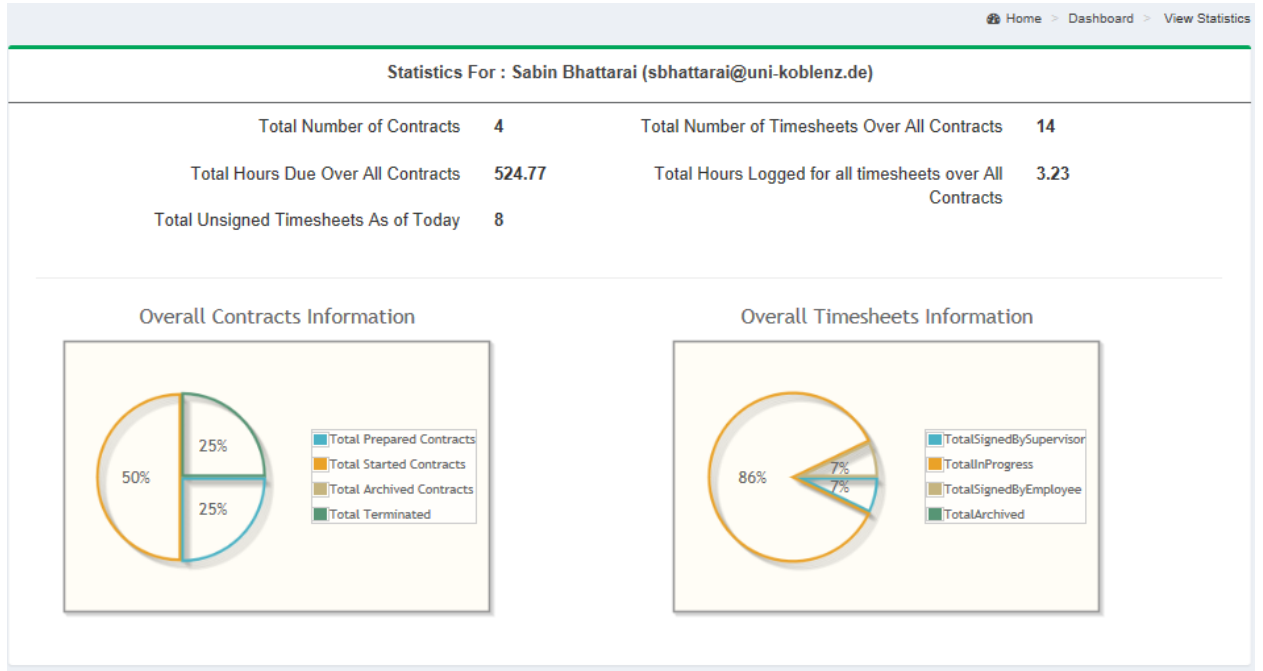
Previous

1

Next

### 10. Contract Statistics for Employee.

Employees will be able to view statistics for all the contracts they have with more detailed view about the contracts and timesheets as shown.





# Glossary

<b>Term</b>	<b>Definition</b>
Administrator	An Administrator is a university staff member. Administrators are responsible to install, configure, and operate the TSS.
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.
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.