Time Sheet System (TSS)

JakartaEE Web Applications 2025 Examination Project

System Vision and Requirements Version 1.5 (May 22nd 2025)

TABLE OF CONTENTS

TIME SHEET SYSTEM (TSS)

SYSTEM VISION

The German federal law for minimum wages („Mindestlohngesetz") requires that employees with a monthly income below a certain amount 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 have to 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 deleted.

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 TSS shall support the time sheet business. The supervisor and/or a secretary can enter contract data (name of supervisor, name of employee, duration and hours per week) for the student assistants. Employees can view their contracts.

The system keeps track of the hours due and the hours completed. To get an overview, a short description of the work content can be stored. An ideal solution would also support working hour reports via mobile devices.

The TSS shall provide basic statistics for supervisors as well as for employees.

For practical reasons, the frequency of printed reports should be configurable, e.g. once a week, or at the end of a month. The TSS shall send reminders by email in case of missing entries, missing reports, missing signatures, etc.

By the nature of the problem, the system records sensitive personal information that is subject to European and federal data protection acts. Wherever possible, the system shall employ technical measures to guarantee privacy and confidentiality (e.g. encrypted transfer) and ensure that only authorized people can access the data.

Prior to using the system, users have to give consent store their data, and confirm that they understand the regulations and the protection measures that protect their data. It is assumed that the operating environment of the system servers is secured in a way that only trusted personnel can gain access to hardware, server software and databases.

TSS REQUIREMENTS

GENERAL CONVENTIONS

TSS requirements contain an *obligation keyword*. This is one of:

- SHALL - denotes a mandatory requirement that has to be realized

- SHOULD - denotes an optional requirement that has a certain value - it may be skipped if serious obstacles impede the realization

- WILL - denotes an interface requirement that has to be taken into account but the realization of the functionality is outside of the system

Regarding the examination and grading of the course project, there is a minimum set of requirements that have to be realized to pass (grade 4.0 „sufficient"). Those requirements are labeled with MIN.

ROLES AND USE CASES

Roles

Employee

An employee in the TSS is a person that works under one or more contracts. An employee can be a student or a university staff member.

Supervisor

A supervisor is a university staff member. The supervisor is the contractual boss of an employee. Supervisors have to sign the time sheets prior to archiving.

Assistant

An assistant is a university staff member. Assistants are responsible for the concrete tasks assigned to employees. Assistants may as well manage contracts in the TSS.

Secretary

A secretary is a university staff member. Secretaries are responsible for printing the time sheets. They forward the time sheets to the supervisor in order to sign them. Secretaries are also responsible for archiving the signed time sheets.

Guest

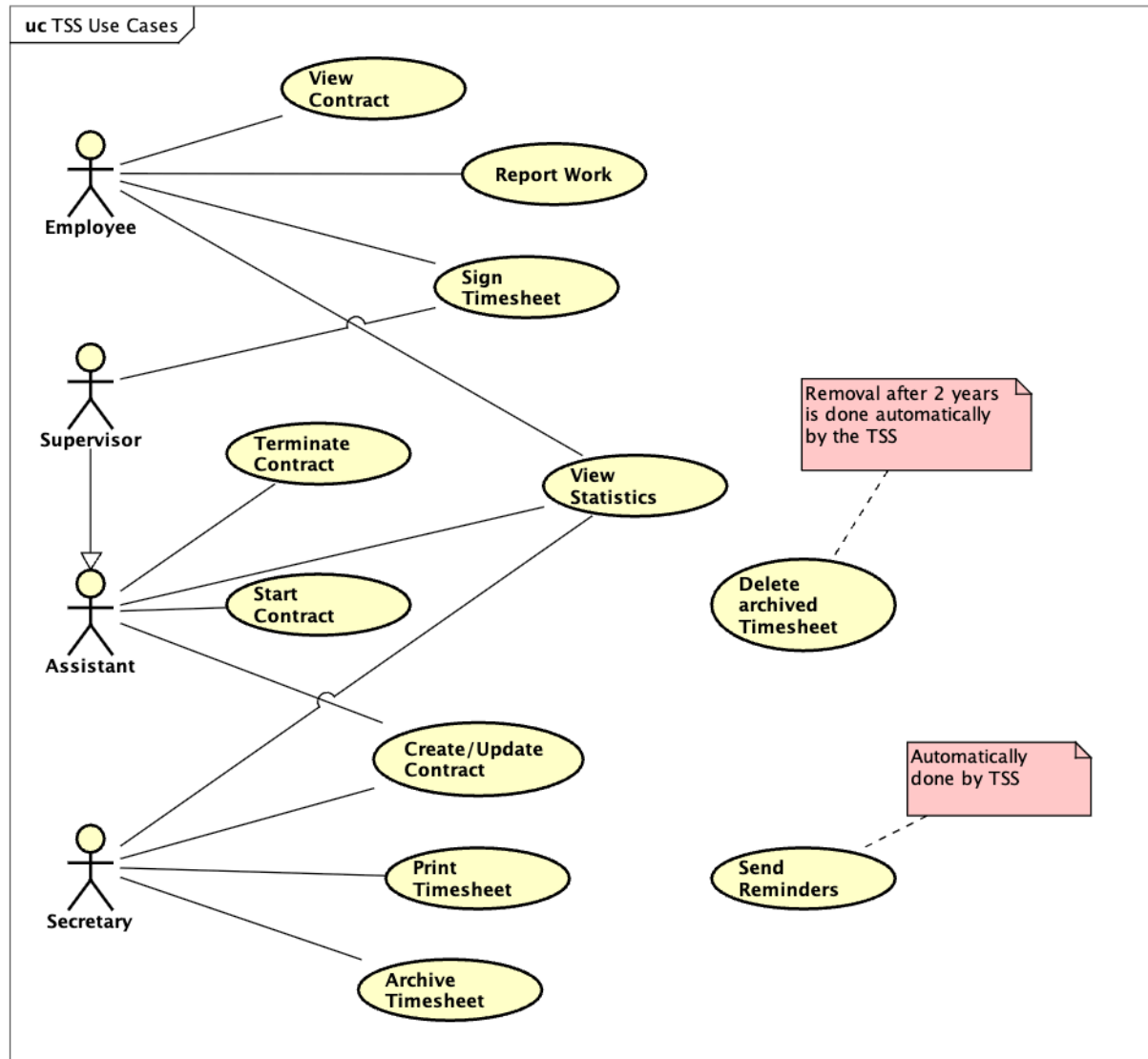Guests are unregistered system users. Guests may only view public information and documentation about the TSS.
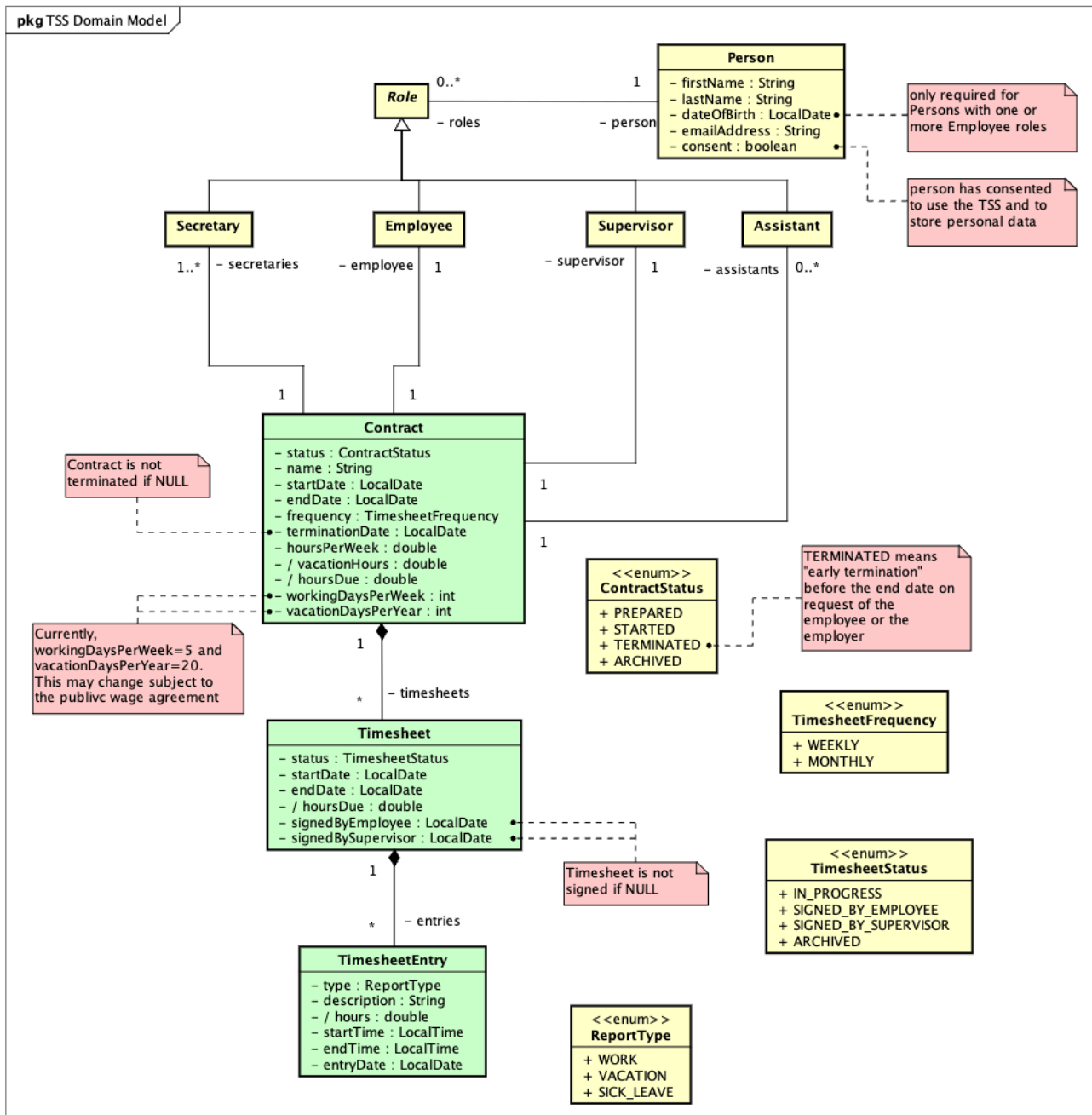
Administrator

An Administrator is a university staff member. Administrators are responsible to install, configure, and operate the TSS. As TSS users, they may get additional responsibilities, e.g. deleting contracts of former employees or database management tasks.
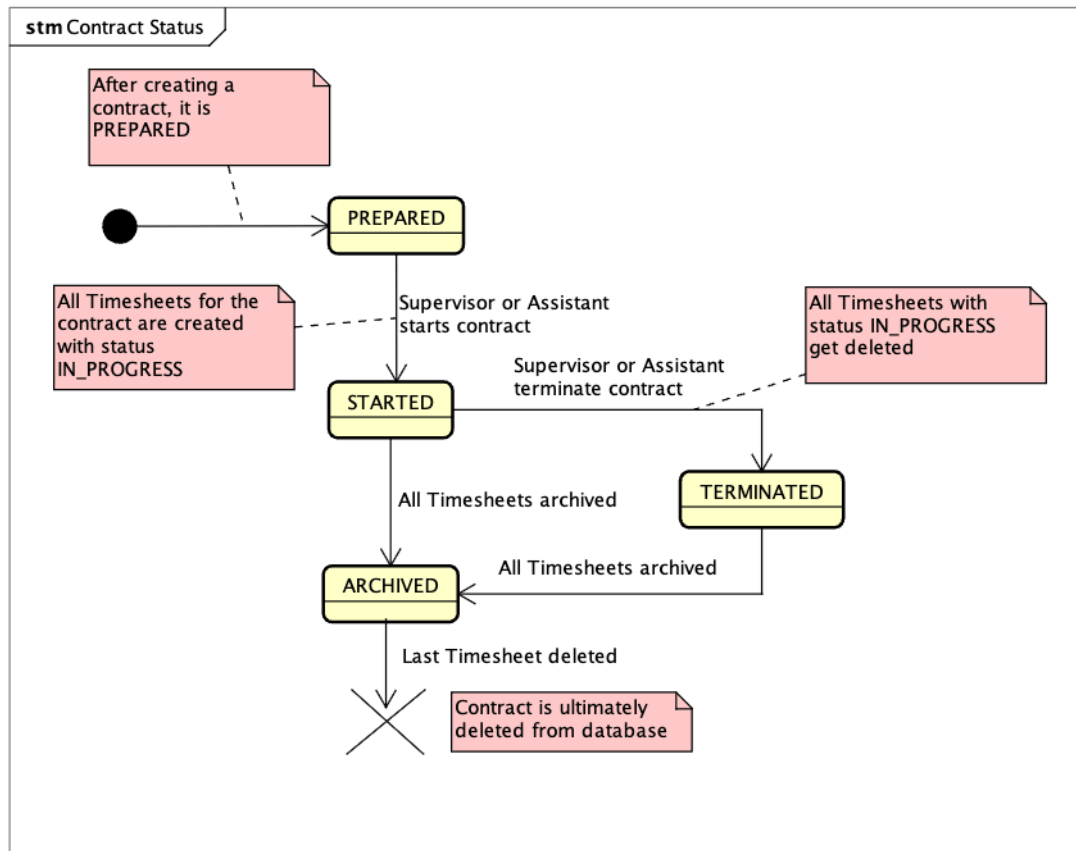
TSS Use Cases

DOMAIN MODEL

pkg TSS Domain Model

Person
- firstName : String
- lastName : String
- dateOfBirth : LocalDate
- emailAddress : String
- consent : boolean

Role   0..*   1

– roles   – person

only required for Persons with one or more Employee roles

person has consented to use the TSS and to store personal data

Secretary   **Employee**   **Supervisor**   **Assistant**

1..* – secretaries   – employee 1   – supervisor 1   – assistants 0..*

1   1

Contract
- status : ContractStatus
- name : String
- startDate : LocalDate
- endDate : LocalDate
- frequency : TimesheetFrequency
- terminationDate : LocalDate
- hoursPerWeek : double
- / vacationHours : double
- / hoursDue : double
- workingDaysPerWeek : int
- vacationDaysPerYear : int

Contract is not terminated if NULL

Currently, workingDaysPerWeek=5 and vacationDaysPerYear=20. This may change subject to the publivc wage agreement

1

1

1

**<<enum>>
ContractStatus**
+ PREPARED
+ STARTED
+ TERMINATED
+ ARCHIVED

TERMINATED means "early termination" before the end date on request of the employee or the employer

**<<enum>>
TimesheetFrequency**
+ WEEKLY
+ MONTHLY

– timesheets

*

Timesheet
- status : TimesheetStatus
- startDate : LocalDate
- endDate : LocalDate
- / hoursDue : double
- signedByEmployee : LocalDate
- signedBySupervisor : LocalDate

1

Timesheet is not signed if NULL

**<<enum>>
TimesheetStatus**
+ IN_PROGRESS
+ SIGNED_BY_EMPLOYEE
+ SIGNED_BY_SUPERVISOR
+ ARCHIVED

*   – entries

TimesheetEntry
- type : ReportType
- description : String
- / hours : double
- startTime : LocalTime
- endTime : LocalTime
- entryDate : LocalDate

**<<enum>>
ReportType**
+ WORK
+ VACATION
+ SICK_LEAVE

REQUIREMENTS

Contract State Diagram

Contracts

CN1 MIN   The TSS SHALL provide assistants and supervisors with the ability to manage contracts. Manage means „CRUD" (Create, Read, Update, Delete). Update and deletion of contracts is only possible in state PREPARED.

CN2   The TSS SHALL provide secretaries with the ability to print contracts. This printout is intended to be used as cover page for the paper archive of the printed time sheets.

CN3 MIN   The TSS SHALL provide employees, assistants, supervisors, and secretaries with the ability to view contract statistics (e.g. sum of hours due, balance).

CN4 MIN   The TSS SHALL calculate the total hours due, the vacation hours, the remaining hours due and take into account weekends and public holidays.

CN4a MIN   The TSS SHALL calculate the vacation hours by the following formula:

$$vacationHours = vacationDaysPerYear * durationOfContract / 12 * hoursPerWeek / workingDaysPerWeek$$

The duration of the contract is counted in months.

CN4b MIN   The TSS SHALL calculate the total hours due for the contract as the sum of the hours due of the individual Timesheets.

CN4c MIN   The TSS SHALL calculate the hours due for a Timesheet by the following formula:

hoursDue = (workingDaysInPeriod - publicHolidaysInPeriod) * hoursPerWeek / workingDaysPerWeek

The period can be a week or a month, depending on the TimesheetFreqeuency of the contract.
E.g., if workingDaysPerWeek = 5, Monday to Friday count as working days, Saturday and Sunday as non-working days.
Only public holidays have to be counted that are on working days.

CN4d MIN   The TSS SHALL be able to determine the public holidays in Rhineland-Palatinate starting from January 01 2025 until December 31 2030 (at least).

CN4e   The TSS SHALL be able to determine the public holidays in Germany from January 01 2025 until December 31 2030 (at least).
Then, the federal state must be configurable since there are different regulations for public holidays in the various parts of the country.

CN5 MIN   The TSS SHALL set a contract to PREPARED status as soon as it is created.

CN6 MIN   The TSS SHALL 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.

CN6a   The TSS SHALL ensure that start date and end date of a contract always denote complete months (i.e. the start date must be the first day of a month, the end date the last day).

CN7 MIN   The TSS SHALL provide assistants and supervisors with the ability to start a contract. Then, the contract changes from PREPARED to STARTED status. All Timesheets are created in status IN_PROGRESS.

CN8   The TSS SHALL provide assistants and supervisors with the ability to terminate a started contract. Then, the contract changes from STARTED to TERMINATED status.

CN9 MIN   The TSS SHALL ensure that only contracts can be aborted whose time sheets are in status SIGNED_BY_SUPERVISOR or IN_PROGRESS or ARCHIVED.

CN10   The TSS SHALL record the date of termination of a contract.

CN11   Before terminating a contract, the TSS SHOULD warn the user if there are time sheets in state IN_PROGRESS that have entries. The user may then decide to not terminate the contract yet.

CN12 MIN   The TSS SHALL set a contract to status ARCHIVED as soon as all time sheets of that contract are in status ARCHIVED.

Time Sheet State Diagram
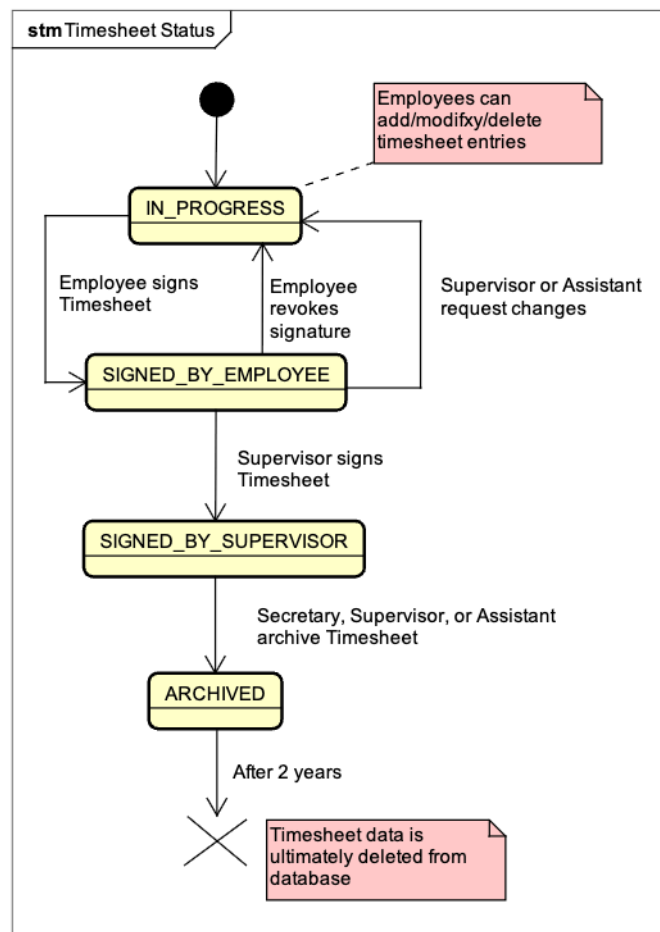
Time Sheets

TS1 MIN   The TSS SHALL create all time sheets for a contract based on the time sheet frequency and the start and end dates of the contract as soon as the contract enters the STARTED status.

TS2 MIN   The TSS SHALL ensure that time sheet entries can only be added, changed, and removed when the time sheet is in IN_PROGRESS status and the contract of the time sheet is in STARTED status.

TS3 MIN   The TSS SHALL ensure that the total hours reported in entries of type VACATION can not exceed the total vacation hours (see requirement CN4a).

TS4   The TSS SHALL delete time sheets in status IN_PROGRESS as soon as the contract status changes to TERMINATED.

TS5 MIN   The TSS SHALL not delete time sheets that are in the SIGNED_BY_EMPLOYEE state.

TS6 MIN   The TSS SHALL not delete time sheets that are in the SIGNED_BY_SUPERVISOR state.

TS7 MIN   The TSS SHALL provide employees, assistants, supervisors, and secretaries with the ability to view time sheets.

TS8   The TSS SHALL provide secretaries with the ability to print time sheets.

TS9 MIN   The TSS SHALL provide employees with the ability to manage their time sheet entries. Changes to entries are only allowed when a Timesheet is IN_PROGRESS.

Signatures

SG1 MIN   The TSS SHALL provide employees with the ability to sign a time sheet. Then, the time sheet changes from IN_PROGRESS to SIGNED_BY_EMPLOYEE status.

SG1a   The TSS SHALL provide employees with the ability to revoke the signature. Then, the time sheet changes from SIGNED_BY_EMPLOYEE to IN_PROGRESS status.

SG2 MIN   The TSS SHALL provide supervisors with the ability to sign a time sheet that is in status SIGNED_BY_EMPLOYEE. Then, the time sheet changes to SIGNED_BY_SUPERVISOR status.

SG3   The TSS SHALL provide assistants and supervisors with the ability to request changes to a time sheet that is in status SIGNED_BY_EMPLOYEE. Then, the time sheet changes to IN_PROGRESS status.

SG4   The TSS SHOULD provide employees and supervisors with the ability to digitally sign the time sheets.

Reminders

RE1 MIN   On the last day of a time sheet (either end of week or end of month), the TSS SHALL send a reminder mail to the employee if the time sheet is in state IN_PROGRESS.

RE2   The TSS SHALL send a reminder mail to the supervisor and the assistants if the time sheet is in state SIGNED_BY_EMPLOYEE. (then, the supervisor may reject or sign the time sheet)

RE3   The TSS SHALL send a reminder mail to the secretaries if the time sheet is in state SIGNED_BY_SUPERVISOR.

RE4   The TSS SHALL repeat reminders every day.

RE5   The TSS SHOULD collect all reminders so that a person receives at most one e-mail per day.

Archiving

AR1 MIN   The TSS SHALL provide secretaries with the ability to archive time sheets that are in status SIGNED_BY_SUPERVISOR. Then, the time sheet changes to status ARCHIVED.

AR2 MIN   The TSS SHALL ensure that archived time sheets can not be changed.

AR3   The TSS SHALL delete time sheets 2 years after the signature of the supervisor. When all time sheets of a contract are deleted, the contract has to be deleted as well.

AR4	The TSS SHOULD support variable archive durations. In this case, the 2 years duration is not fixed, but must be stored per contract. The duration should have a default value of 2 years.

Access Control

AC1 MIN	The TSS SHALL authenticate users prior to giving access to any data.
AC2	The TSS SHALL be able to determine whether a person is university staff member.
AC3 MIN	The TSS SHALL be designed in a way that only authorized users may view/change/delete data. Access rules are based on the users affiliation (staff, student) as well as on the roles a user owns with respect to a contract.

User Interface

UI1 MIN	The TSS SHALL be designed in a way that the most frequent task for employees (report work) is accessible immediately after successful login.
UI2	The TSS SHOULD support mobile devices.
UI3	The TSS SHALL be designed in a way that it can be used with different browsers. At least, FireFox, Safari, Chrome shall be supported.
UI4	The TSS SHOULD use the PrimeFaces Component library.

Internationalization

IN1 MIN	The TSS SHALL be designed in a way that the user interface language can be switched.
IN2	The TSS SHALL provide the users with the ability to choose their preferred language.
IN3 MIN	The TSS SHALL support at least two user interface languages.
IN4 MIN	The TSS SHALL support English as user interface language.
IN5	The TSS SHALL support German as user interface language.
IN6	The TSS SHOULD send reminders to users in their preferred language.

Software Architecture

SA1 MIN	The TSS SHALL be implemented according to the layered architecture (see tss-architecture document).
SA2 MIN	The TSS SHALL contain (at least) two modules, the web module and the EJB module.
SA3 MIN	The TSS project team SHALL ask for consent prior to using third-party libraries that were not presented in the lecture/lab. Usage is permitted only after negotiation with the supervisor. The Rewrite and PrimeFaces libraries presented in the lecture and tutorial examples may be used without prior notice.
SA4 MIN	The TSS SHALL use one of the database servers MariaDB (preferred), PostgreSql, or JavaDB/Derby.

SA5 MIN   The project team SHALL document architectural decisions, e.g., which subsystems do exist, how is the source code structured, etc. in the project manual.

SA6 MIN   The project team SHALL prefix all global names with its team name. Such names are, e.g., project name, context path (root URL), database name, persistence units, server resources, etc.
             This is to avoid naming conflicts, since the customer has to install the systems of many teams side-by-side.

Project Manual

PM1 MIN   The deployment process for the TSS SHALL be documented in the project manual (can be online, e.g. GitLab Wiki) such that the customer can install the system on a fresh Glassfish server.

PM2 MIN   The project manual SHALL contain a list of completed/missed requirements.

PM2a MIN   If there are missed requirements, the project team SHALL write a short explanation why the requirement could not be fulfilled.

PM3 MIN   The project manual SHOULD contain descriptions of problems that occurred during the implementation and their solutions.

PM4 MIN   Decisions made to detail and/or change requirements SHALL be documented in the project manual.

PM5   The project participants SHALL record the individual time spent on the project and their specific tasks. One possibility to do so is to use GitLab's issue tracking feature.

ARCHITECTURE SKETCHES

Presentation	Visualization of values, Input areas, graphics, interaction with I/O devices
Application Logic	Control of sequence of actions, navigation, validation of input values, prevention of unauthorized access, conversion of values
Business Logic	Realization of basic application functions, interfaces to other systems, prevention of unauthorized access
Database	Persistent storage of data, backup, transaction management, efficient access…

For simple systems, the facade implementation may also contain the concrete Business Functions

There can be multiple facades for different clients, as well as for different subsystems. Each subsystem also could have its own facades.

In our architecture, Entities are allowed in Business Functions.

This is not advisable in really big applications, instead business (transfer) objects should be used.

Java Client

SOAP/REST Client

Transfer Objects   Transfer Objects

EJB Tier

@Remote Interface Facade

Business Logic

SOAP/REST Service Facade

Facades only offer stateless services

Transfer Objects

Facade Implementation

Facade Implementation

NO Entities above this line

Subsystem Business Functions

Subsystem Business Functions

Subsystem Business Functions

NO EntityManager operations and/or queries above this line

Data Access Objects

Entities

EntityManager

JPA

Persistence Provider (EclipseLink)

JDBC

DB

Presentation Layer (Browser)	Client Tier

HTML document

HTTP request (GET/POST)   HTTP response

Web Tier

XHTML „page"

EL expressions

JSF Render Kit

Faces Servlet

executes JSF Lifecycle

Application Logic Layer

properties, action methods

Backing Beans

@EJB reference

BB instances created based on scope (request/view/session/application)

transfer objects

Facades only offer **stateless** services

@Remote **Interface Facade**

Business Logic Layer

Business Logic   EJB Tier

JPA via JDBC

EIS Layer

DB   DB Tier