Scheduling and Time Management Application - Requirements

Group 20:
Sunghoon Cho
Michael Czapary
Christopher Elliott
Alex Kolstad
Zachary Wetekamm

Requirements Definition

Functional Requirements

- The system shall authenticate the employee's credentials before logging them in
- The system shall allow employees to check their schedules and work history from anywhere
- The system shall allow employees to use sick days and vacation days
- The system shall allow employees to communicate with management to apply for paid time off
- The system shall, if allowed by management, allow employees to create their own schedule
- The system shall be universal for all jobs in the workplace
- The system shall track employee's lateness and find trends
- The system shall help to troubleshoot issues within culture of workplace
- The system shall identify employees who may be overworked and burnt out
- The system shall use multiple data parameters to help management identify why individual employees are late
- The system shall not allow employees to clock in from any location other than the work site
- The system shall allow employees to clock in and out, clock in and out to take their lunch, and clock in and out for their break time

Non-Functional Requirements

- The system shall log users in within 20 seconds
- The system shall log the user out after 5 minutes of inactivity
- The system shall not allow employees to clock in after a specified period of "lateness"
- The system shall notify management when someone is late

Use Case 1: Employee Timestamp

Actors:

- Non-admin Employees
- Admin Employees

Preconditions:

- User is onsite at company office.
- User is registered as an authorized company user.
- User is registered to the application system.
- User logs into the system using username and password credentials.

Postconditions:

- User has successfully recorded timestamp for either:
    - Clock In/Out
    - Break
    - Lunch
- User receives a success or failure notification for any timestamp query.
- Appropriate Admins are notified if an employee is tardy or absent.
    - Frequency of notifications are determined by user preferences.
- User can access a view of their recorded timestamps from a customizable date range.
    - Admins can edit recorded timestamps.
        - Any edited timestamp is associated with the admin that edits that timestamp.
        - Admins cannot edit their own timestamps.
    - Non-admins cannot edit recorded timestamps.

Flow of Events:

- User is onsite and logs into an authorized company computer.
- User logs into the application with username and password credentials.
- User records a timestamp for the appropriate condition:
    - Clock In/Out
    - Break
    - Lunch
- If user is Admin, user may edit an existing timestamp for an employee.
- User receives confirmation that timestamp was recorded.

Use Case 2: Notification of Tardy or Absent Employees

Actors:

- Non-admin Employees
- Admin Employees

Preconditions:

- A registered and authorized user logs into the system using appropriate credentials.
- User has successfully records appropriate timestamps
- System uses employee time schedule and check-in status to determine employee presence
- Admin employees have opted to receive notification for tardy or absent employees

Postconditions:

- Admin employees are notified of tardy or absent employees based on their settings
    - Notification type and frequency can be set based on preference
- Tardy or absent employees are notified of their status at check-in or at set times (e.g., 10 minutes past starting time) based on preferences set by respective admin employees
- Admin/supervisor or employee can choose to document reason for tardiness or absence
- Admin employees can check whether tardiness or absence was pre-approved or notified in advance
- Separate notifications for persistent tardiness can be sent (conditions for persistence determined by admin employee, e.g., based on repeated reasons)

Flow of Events:

- System records timestamp by employee
- System sends employee and admin notification when timestamp is recorded based on previously set preference
    - Notifications can be sent only at late check-in
    - Notifications sent after set time period past beginning of shift
- User or admin/supervisor can choose to document reason for tardiness or absence
- Based on preferences, system sends notification to admin/supervisor for persistent tardiness

Use Case 3: Request Paid Time Off

Actors:

- All employees up through management should have access to this use case

Preconditions:

- The employee is registered in the system
- The employee is logged into the system (from anywhere)
- The employee has sufficient vacation or sick days remaining to cover request
- The employee has given sufficient time in advance of request so that their shifts can be covered
    - If a request comes after the predetermined advanced notice time then a notification should be sent to the on duty supervisor for approval

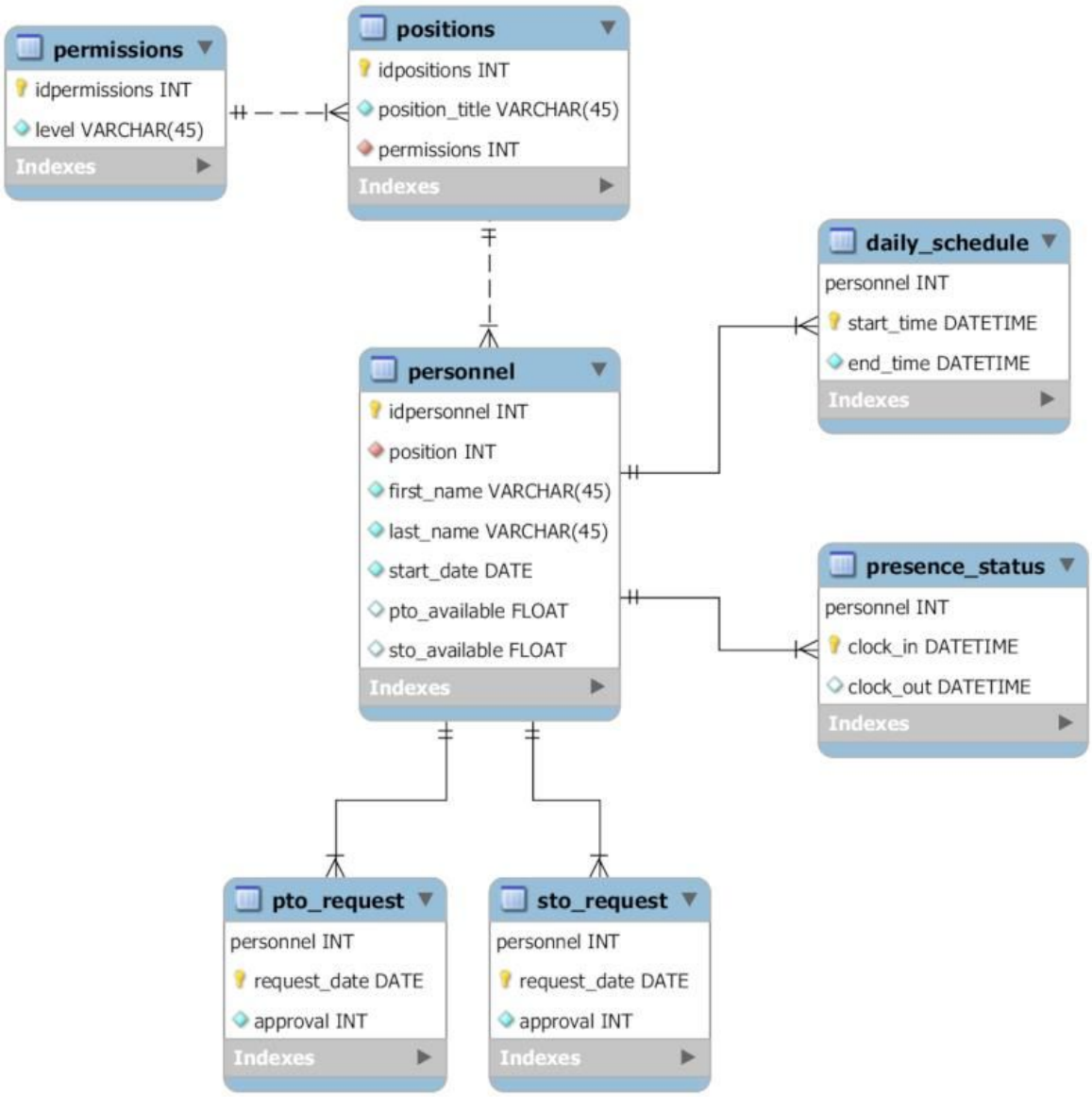
Postconditions:

- *If the request is granted then the scheduler is notified of the days and time the employee is unavailable
- Remaining sick/vacation/personal time off is reduced by the amount being used
- An approval message is sent to the employee so they know the request was granted.
- *If the request come later then the predetermined advanced notice time then the employee will get a message informing them that their request is being reviewed by a supervisor. After if is accepted or denied it will follow the regular postcondition route.
- *If a request is denied a message will be sent to the employee informing them why (not enough advanced notice, not enough remaining vacation days etc.) The employee will then be given the option to make a special request to their supervisor if they have special circumstances to be considered.

Flow of Events:

- Employee logs into system with their username and password from anywhere
- User fills out and submits request form
- *If necessary, message is immediately sent to supervisor for review.
- A notification is sent the employee either confirming request was granted, informing that it is being reviewed, or informing the employee why it was denied and allowing them the option to contact an active supervisor if they have special circumstances.
- *if the request was granted then the employee's remaining sick days, vacation days or PTO is immediately reduced.
- *if the request was granted a notification is sent to scheduling so that they know when the employee will be unavailable

Entity Relationship Diagram:



Requirements Specification

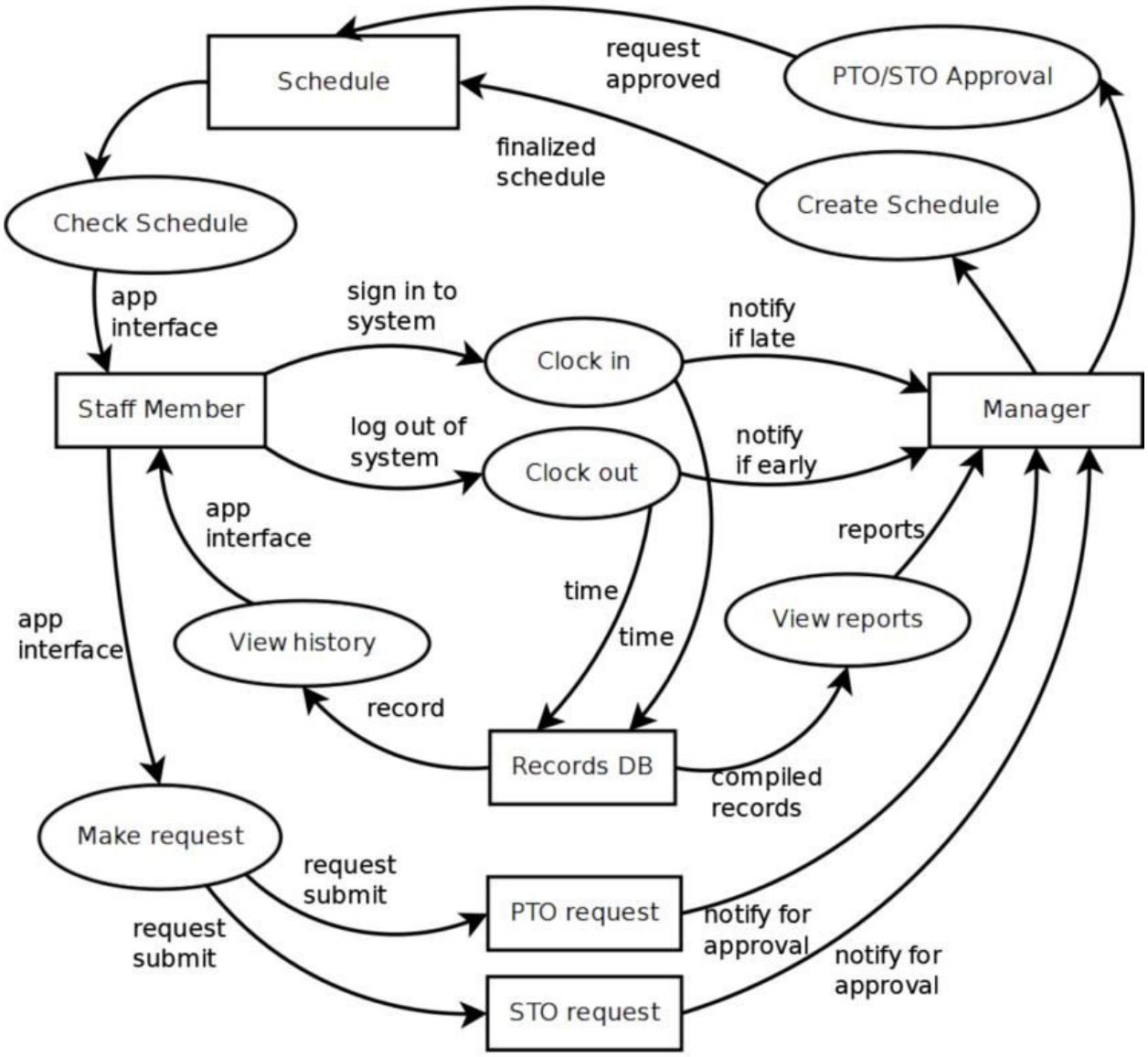
Functional Requirements

- User details with be validated with their login information stored on the server
- The system will allow employees to check their schedules and work history from anywhere
- The system will allow employees to use sick days and vacation days, and if used, will subtract their used sick and vacation days from their total sick and vacation days allowed
- The system will allow employees to apply for paid time off and specify reasons to management
- The system will, if allowed by management, allow employees to choose their own days and hours
- The system will work for all possible jobs in the workplace
- The system will track employee's lateness and discover possible lateness trends
- The system will help to troubleshoot issues within culture of workplace
- The system will use lateness data to identify employees who may be overworked and burnt out
- The system will use multiple data parameters to help management identify why individual employees are late
- The system will not allow employees to clock in from any location other than the work site, or any computer other than the main work computer
- The system will allow employees to clock in and out, clock in and out to take their lunch, and clock in and out for their break time

Non-Functional Requirements

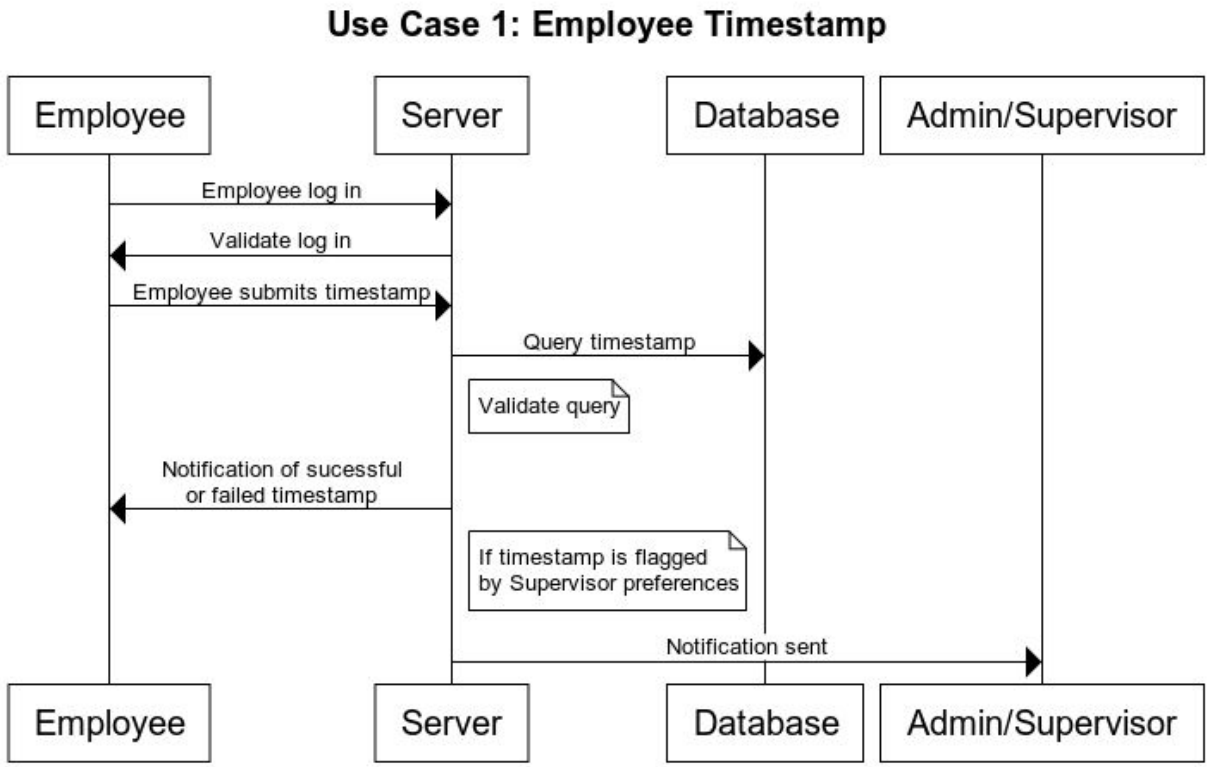
- The system will validate users' data and log users in within 20 seconds
- The system will log the user out after 5 minutes of inactivity
- The system will not allow employees to clock in after a specified period of "lateness" set by management
- The system will notify management when someone is late

Dataflow Diagram:

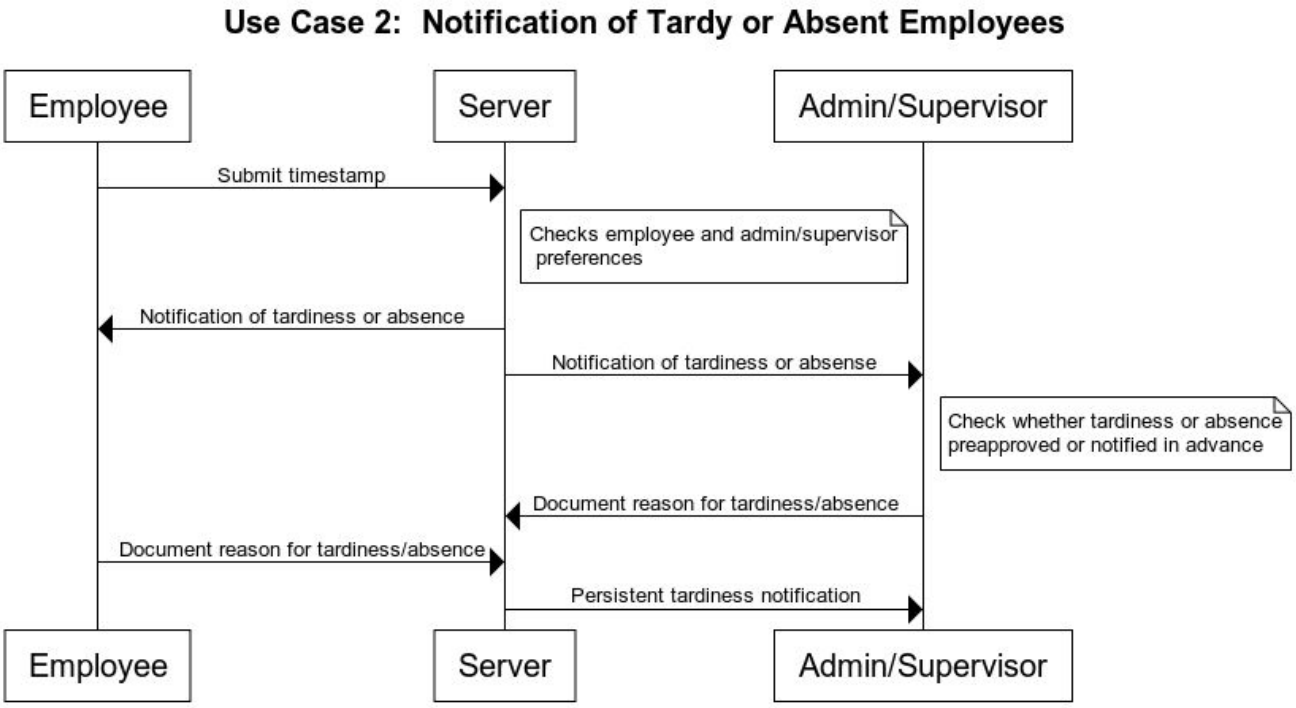


Message Sequence Charts:

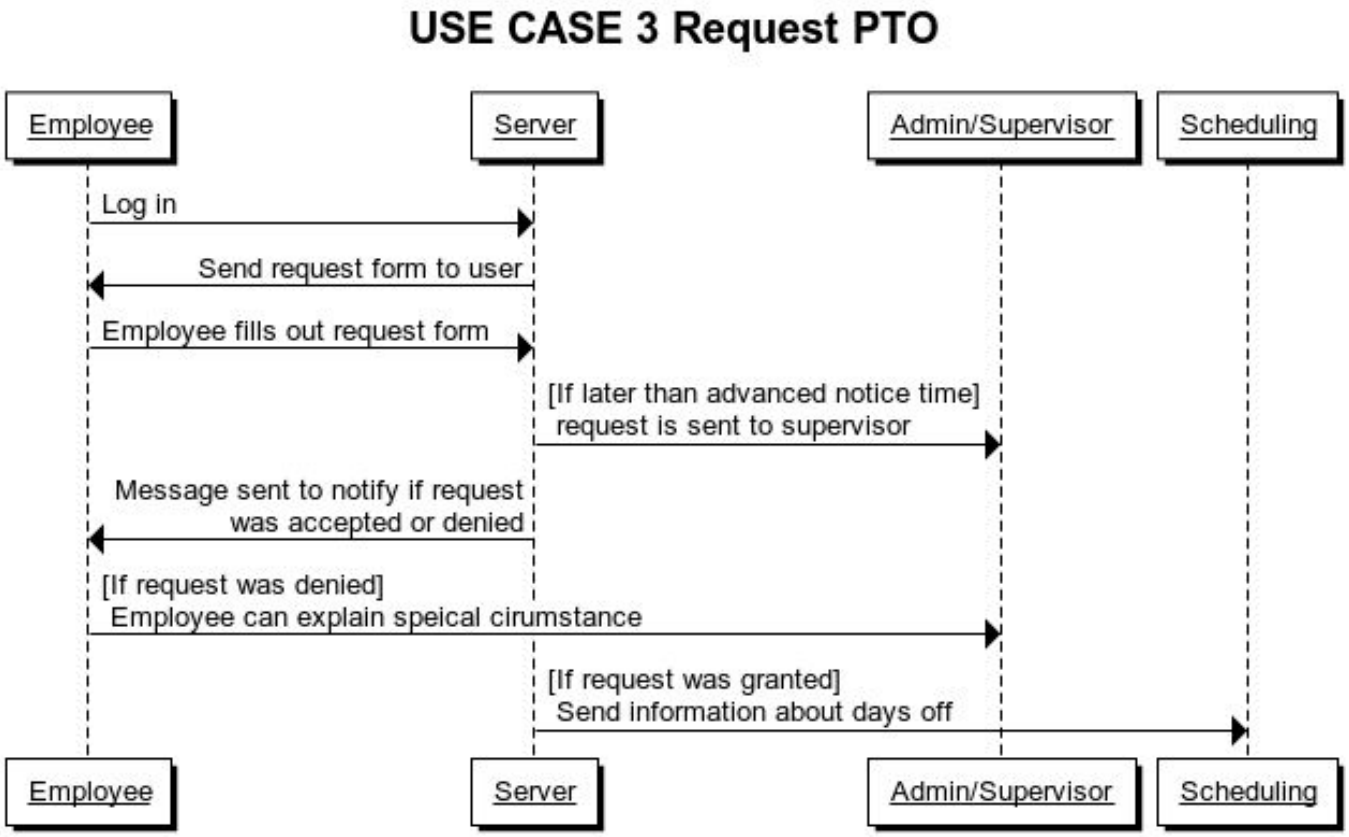
Use Case 1:



Use Case 2:


Use Case 2: Notification of Tardy or Absent Employees
Employee
Server
Admin/Supervisor
Submit timestamp
Checks employee and admin/supervisor preferences
Notification of tardiness or absence
Notification of tardiness or absense
Check whether tardiness or absence preapproved or notified in advance
Document reason for tardiness/absence
Document reason for tardiness/absence
Persistent tardiness notification
Employee
Server
Admin/Supervisor

Use Case 3:


USE CASE 3 Request PTO
Employee
Server
Admin/Supervisor
Scheduling
Log in
Send request form to user
Employee fills out request form
[If later than advanced notice time]
request is sent to supervisor
Message sent to notify if request
was accepted or denied
[If request was denied]
Employee can explain speical cirumstance
[If request was granted]
Send information about days off
Employee
Server
Admin/Supervisor
Scheduling

Customer Meeting:

Our Customer, Erica, met with our team Monday, 10/14/19, via Google Hangouts and discussed the purpose and vision of the application.

Team Contributions:

All team members met Wednesday, 10/16/19, and discussed project goals and scope. Specific to Homework 1, work was divided evenly across different tasks:

- Alex Kolstad: Requirements definition (functional/non-functional requirements) and Requirements specification (functional/non-functional requirements)
- Zachary Wetekamm: Use Case 1 structured description and message sequence chart
- Sunghoon Cho: Use Case 2 structured description and message sequence chart
- Michael Czapary: Use Case 3 structured description and message sequence chart
- Christopher Elliott: Entity Relation Diagram and Dataflow Diagram