Group members:

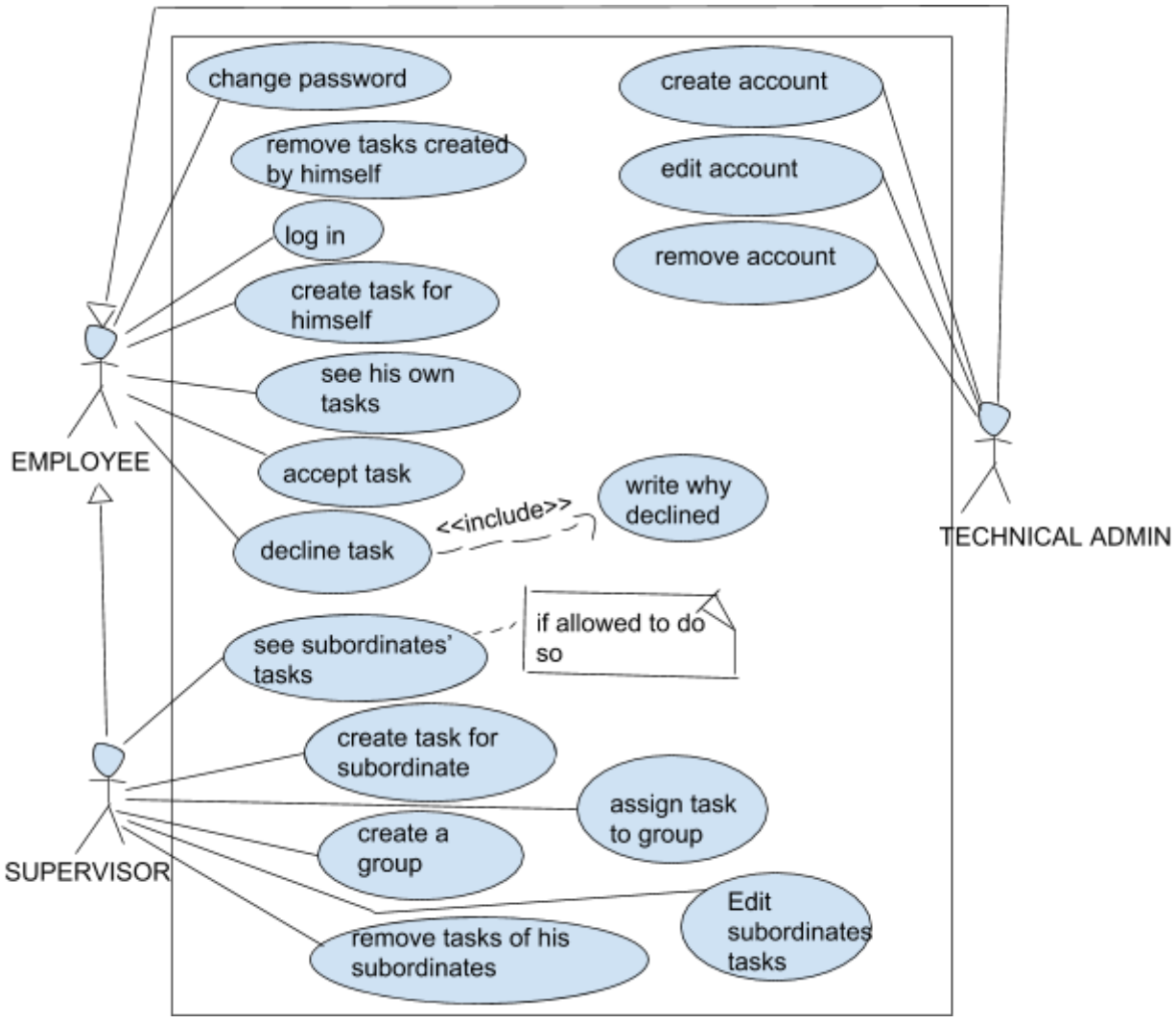
- Maciej Adamczyk s15170
- Radosław Grelus s13720
- Natalia Wysokińska s15412

LAB 1

1. **Short description:**

   This software is dedicated to employees and their supervisors. It enables them to organise their work and future tasks. It has a form of a calendar.

   Employees have to log in to the system to be able to use the calendar. An employee can create a task, remove tasks created by himself. Supervisors can assign tasks to their subordinates' calendars. If a person has no rights to a person's calendar, then this person mustn't see this person's calendar. An employee can accept or decline a task. If he/she declines it, he/she has to say why (e.g. write a comment). A user can't assign a task at hour which is already taken in his calendar. A supervisor can create groups, but also be a part of a group. He assigns tasks to groups and also removes tasks of his subordinates. A technical admin is allowed to create, edit and remove accounts of users of the calendar system. An employee can be also a supervisor or a technical admin.
2. **Use case diagram:**



3. **Descriptions of each use case:**

Natalia:

1.1 Adding employees to a group (creating a group)

1.1.1 Actors

Use-case step description

- Supervisor

1.1.2 Purpose and context

The supervisor of Accounts Receivable Team has subordinates, who work for different entities (for example German entity, French entity), but they all belong to his team. When he assigns a task, he usually wants to assign it to all employees working for a specific entity.

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1.	The AR Team Manager (supervisor) has to add specific employees to a specific group.

1.1.5 Initiating business events

The AR Team Manager has too many employees and too little time to add them one by one. Spending some time to add all employees to right groups just once is easier for him than adding them every time he needs to assign a task.

1.1.6 Basic flow of events

.	1. The manager clicks “Create a group” 2. The application shows all employees from manager’s Team by their name and surname. The manager can find the employee he is looking for in two ways: 2.1. He can select the employees he wants to add from the list on the right (when he selects the employees, their names are bolded - he knows who was selected by him already) 2.2. He can search an employee by his/hers name or by Employee ID in the search field. Employees’ names and Employee ID pop up and can be selected by the manager.

3. Selected employees are displayed in the Employee list on the right hand side
3.1. If the manager selected one of the employees by mistake, he can remove him/her from the list
4. The group should be named in the field group name
5. The manager clicks “Create a group”
6. The group is created

1.1.7 Alternative flows of events

1.1.7.1 Employee cannot be found

Use-case step description

2.2. After typing employee’s name, nothing is shown even though the employee exists and is a part of manager’s team

. 2.3. The manager can try searching for the employee by Employee ID - if it doesn’t work, he should contact with the technical admin
2.4. Technical admin needs to check employee’s account - if it was for example terminated
- if it doesn’t exist - the account needs to be created

2.5. Manager should try searching for an employee again and perform following steps
2.6. Use case ends

1.1.7.2 Group cannot be created

Use-case step description

5. Manager wants to create a group, but an error message appears saying that the group cannot be created, because the manager hasn’t rights to do that

5.1.Technical admin needs to be contacted
5.2. Technical admin needs to give rights to the manager
5.3. Manager should try logging out and logging in again
5.4. Manager should do back to the 1. point of the basic flow of events again and perform all the steps again.
5.5. Use case ends

1.1.8 Extension points

Extension point condition

Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended. It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

Use-case step description
Payment order has been placed within the opened Cashier Session

1.1 Assigning a task to a group of employees

1.1.1 Actors

Use-case step description

- Supervisor

1.1.2 Purpose and context

The supervisor of Accounts Receivable Team has subordinates, who work for different entities (for example German entity, French entity), but they all belong to

his team. When he assigns a task, he usually wants to assign it to all employees working for a specific entity.

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1.	The AR Team Manager (supervisor) has to add specific employees to a specific group.

1.1.5 Initiating business events

The AR Team Manager has too many employees and too little time to add them one by one. It's easier for him to assign a task to a group of employees he created earlier

1.1.6 Basic flow of events

.	

1. The manager clicks "Assign a task"
2. The application shows all employees from manager's Team. and all groups he created earlier. He can select all employees he wants to add, select a group and employees or select only a group. In this case the manager wants to assign a task to a group.

2.1. He can search a group by group's name in the search field

2.2. He can also select the group from the group list

3. If someone from the group isn't available in this time, the manager gets a comment saying that the employee (name) won't be able to perform a specific task.
4. The manager clicks "Ok"
5. The task is assigned to a group and can be seen in all employees' belonging to the group calendar.

1.1.7 Alternative flows of events

1.1.7.1 One of the employees from the group isn't part of this team anymore - is a part of different manager's team

Use-case step description
4. After trying to assign a task to a group, an error occurs as system cannot let the manager assign the task to all employees from the group - one of the employees isn't this manager's subordinate anymore, which means the manager doesn't have rights to assign a task to this employee
4.1. The manager should go to group settings and delete the employee from the group
4.2. The manager needs to try to assign a task to this group again
4.3.Use case ends.

1.1.7.2 Manager doesn't have right permissions to assign a task even though he should have them

Use-case step description
4 Manager wants to assign a task to a group, but an error message appears saying he can't assign a task to these employees as he doesn't have right permissions
4.1. Technical admin needs to be contacted- he should change this manager's account - give him right permissions
4.2. Manager should try to assign the task again

1.1.8 Extension points

Extension point condition
Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended. It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description
1.	Payment order has been placed within the opened Cashier Session

1.1 Removing a task from group's or employee's timetable (remove tasks of his subordinates)

1.1.1 Actors

Use-case step description
- Supervisor

1.1.2 Purpose and context

The supervisor of Accounts Receivable Team has subordinates, who work for different entities (for example German entity, French entity), but they all belong to his team. When there is no need to perform a certain task anymore, the manager should be able to remove it from one employee's or group's timetable

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1.	The AR Team Manager (supervisor) has to remove specific tasks from specific employee's or group's timetable

1.1.5 Initiating business events

When there is no need to perform a specific task anymore, there should be a functionality to simply remove the task from employee's or everyone's in a group timetable.

1.1.6 Basic flow of events

1. Manager double clicks the right task in the calendar view
2. A window with all task's details shows up
3. Manager clicks "Remove task" in the right corner
4. A comment shows up informing the manager how many employees have already accepted the task
5. Manager clicks:
5.1. "OK" - go to 6.
5.2. "Cancel" - go back to 2.
6. Task is removed from all employees' in the group timetables

1.1.7 Alternative flows of events

1.1.7.1 One of the employees from the group isn't working anymore = his account isn't present in the system

Use-case step description
5.1.1.The manager cannot remove the task, because an error occurs - the system cannot remove the task from everyone in the group as one of the employees is part of the group, but left the company, his account isn't active.
5.1.2. The manager should go to group settings and remove the employee from the group

5.1.3. The manager needs to try to remove the task again

5.1.4. Use case ends

11.1.8 Extension points

Extension point condition
Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended. It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description
1.	Payment order has been placed within the opened Cashier Session

1.1 Accept a task

1.1.1 Actors

Use-case step description
- Employee - Supervisor

1.1.2 Purpose and context

A supervisor has subordinates. He might assign tasks to individual subordinates as well as a group of subordinates. When he assigns task to people he would like to know whether an employee accepted or declined a task..

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1	An employee has to be employed and authorized to use the calendar software.

1.1.5 Initiating business events

A supervisor assigns a task to an employee. The employee has to decide whether he accepts, declines the task.

1.1.6 Basic flow of events

1. A supervisor logs in to the calendar system by providing an email address and password. (We assume that he has already an account created by a technical admin)
2. If he has successfully logged, the home screen is displayed.
3. The supervisor creates a task by clicking on the button “Create a task”.
4. A form is displayed. The supervisor fills in the form with information as follows:
    - He can type a name of the task in an input field with a label “Name of the task”
    - He can choose whether to assign a task to a group of his subordinates, a subordinate.
    - description of a task
    - a task being mandatory, voluntary

In this flow of events the supervisor chooses a subordinate and marks a task as voluntary.
5. After the supervisor fills in the form he clicks on a “Submit” button. The task is assigned to a subordinate. Now he has to wait for a response of his subordinate (unless he marked the task as mandatory)
6. A subordinate logs in to the system.
7. He gets a notification about the new task which was submitted by his supervisor.
8. The subordinate clicks on a bookmark called “Tasks”.
9. He displays information about the new task by clicking on the specific task from the list.
10. A windows is displayed with information about the task and two buttons: an “Accept” button and a “Decline” button.
11. A subordinate clicks on the “Accept” button.
12. An information about acceptance of the task is sent back to the supervisor.

1.1.7 Alternative flows of events

1.1.7.1 A user can’t log in to the system and can’t accept a task.

No.	Use-case step description
1	A supervisor/subordinate types an email and a password in the system and tries to log in.
2	He types wrong credentials. Wrong password or wrong email note is displayed. The user is returned to the beginning login screen. To the point no. 1.

1.1.7.2 The subordinate marks a task as mandatory

No.	Use-case step description
	4.1 A task is marked as mandatory. 4.2 The task is sent to one of his subordinates. 10. No accept button, decline button are displayed, because the task was marked as a mandatory task.

1.1.8 Extension points

Extension point condition
Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended. It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description

1	A task has been accepted and a notification has been sent back to a supervisor.

1.1 Decline a task

1.1.1 Actors

Use-case step description

- Employee
- Supervisor

1.1.2 Purpose and context

A supervisor has subordinates. He might assign tasks to individual subordinates as well as a group of subordinates. When he assigns task to people he would like to know whether an employee accepted or declined a task..

1.1.3 Dependencies

1.1.3.1 Included use-cases

Write why declined.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1	An employee has to be employed and authorized to use the calendar software.

1.1.5 Initiating business events

A supervisor assigns a task to an employee. The employee has to decide whether he accepts, declines the task.

1.1.6 Basic flow of events

1. A supervisor logs in to the calendar system by providing an email address and password. (We assume that he has already an account created by a technical admin)
2. If he has successfully logged, the home screen is displayed.
3. The supervisor creates a task by clicking on the button “Create a task”.
4. A form is displayed. The supervisor fills in the form with information as follows:
    - He can type a name of the task in an input field with a label “Name of the task”
    - He can choose whether to assign a task to a group of his subordinates, a subordinate.
    - description of a task
    - a task being mandatory, voluntary

In this flow of events the supervisor chooses a subordinate and marks a task as voluntary.
5. After the supervisor fills in the form he clicks on a “Submit” button. The task is assigned to a subordinate. Now he has to wait for a response of his subordinate (unless he marked the task as mandatory)
6. A subordinate logs in to the system.
7. He gets a notification about the new task which was submitted by his supervisor.
8. The subordinate clicks on a bookmark called “Tasks”.
9. He displays information about the new task by clicking on the specific task from the list.
10. A windows is displayed with information about the task and two buttons: an “Accept” button and a “Decline” button.
11. A subordinate clicks on the “Decline” button.
12. A pop up screen is displayed with empty space to write an explanation why the task has been declined. The subordinate has to write the explanation.
13. The subordinate click on a “Submit” button.
14. An information about the declination of the task is sent back to the supervisor.

1.1.7 Alternative flows of events

1.1.7.1 A user can’t log in to the system and can’t decline a task.

No.	Use-case step description
1	A supervisor/subordinate types an email and a password in the system and tries to log in.
2	He types wrong credentials. Wrong password or wrong email note is displayed. The user is returned to the beginning login screen. To the point no. 1.

1.1.7.2 The subordinate marks a task as mandatory

No.	Use-case step description

4.1 A task is marked as mandatory.
4.2 The task is sent to one of his subordinates.

10. No accept button, decline button are displayed, because the task was marked as a mandatory task. End of the flow.

1.1.8 Extension points

Extension point condition

Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended. It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description
1	A task has been declined and a notification has been sent back to a supervisor.

1.1 Log in

1.1.1 Actors

Use-case step description

- Employee/ Supervisor

1.1.2 Purpose and context

An account for a supervisor/employee has been created. The employee got his login credentials. He would like to use the calendar system and for instance create tasks for himself.

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1	The employee has to be employed and authorized to use the calendar software.

1.1.5 Initiating business events

Entering the software calendar login screen. A person has to type in his email and password.

1.1.6 Basic flow of events

1. An employees enters the calendar software.
2. A pop-up screen is displayed with “email” and “password” input fields.
3. He provides his user credentials into the input fields.
4. He clicks on the “Submit” button and waits for the system to verify his credentials.
5. After a couple of seconds he gets logged in successfully and the “Home” screen is displayed.

1.1.7 Alternative flows of events

1.1.7.1 A user can’t log in to the system.

No.	Use-case step description

5 After a couple of seconds he gets a message “Wrong password/email. Try again”. The data he entered was incorrect. He starts from the 2nd point in the flow of events.

1.1.8 Extension points

Extension point condition
Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended. It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description
1	An employee manages to successfully log in if the credentials were correct. The “Home” screen is displayed.

Radek :

1.1 Creating tasks for himself

1.1.1 Actors

No.	Use-case step description

1.	Employee

1.1.2 Purpose and context

During work an employee identified a task to be accomplished in later date. He uses the software to reserve time to complete it. Thus making sure that his boss will not add him more while he still has stuff to do.

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1.	Employee has been employed by company and has received his login data

1.1.5 Initiating business events

Employee wants to add a task to his timetable

1.1.6 Basic flow of events

No.	Use-case step description
1.	Employee logs in to the system using his email and password received on email , after entering his credentials clicks “login” button.
2	Home screen gets displayed
3	Employee clicks option “create task”.
4.	Form gets displayed prompting user to insert information about: -label or name of a task -description -start date -end date

5	Employee clicks “submit” option.
6	Message displaying inserted task with all its parameters gets displayed.

1.1.7 Alternative flows of events

1.1.7.1 Employee failed to log into his account (a)

No.	Use-case step description
1.	After entering account data and clicking “login” software displays “incorrect account id or password”
.	The flow returns to step 1 of Basic Flow.

1.1.7.2 Employee attempts to insert task in place of already occupied slot (b)

No.	Use-case step description
5.	After logging in employee tries to add task in already occupied time
.	software prompts user that he already has planned tasks for that time and returns to step 3 of basic flow

1.1.7.3 Employee enters conflicting data in form(c)

No.	Use-case step description
5.	After submitting form system detects that data is incorrect and prevents adding task
.	software prompts user that date is incorrect and returns to step 4 of basic flow.

1.1.8 Extension points

Extension point condition

Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended.
It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description
1.	Task has been added to timetable associated with correct employee

1.1 Viewing timetable(See his own tasks)

1.1.1 Actors

No.	Use-case step description
1.	Employee

1.1.2 Purpose and context

Employee has just arrived in his workplace . He doesn't really remember specification of his first task or needs to check if his supervisor added him more work for today or in the future.

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description
1.	Employee has been employed by company and has received his login data

1.1.5 Initiating business events

Employee needs to know information or time when his next task starts or if he has received any new tasks from his supervisor

1.1.6 Basic flow of events

No.	Use-case step description
1.	Employee logs into the system using his email and password acquired from admins , after entering his credentials clicks “login” button.
2.	After successful login attempt home screen gets displayed
3	Now employee can see all tasks for this week.
4	In case he needs to see a day in greater detail or entire month he can choose appropriate option .
5	All tasks are colour coded: -green for tasks he needs to do -red for taks that currently aren't planned for him -blue for tasks that require his attention

1.1.7 Alternative flows of events

1.1.7.1 Employee failed to log into his account (a)

No.	Use-case step description
2.	After entering account data and clicking “login” software displays “incorrect account id or password”
.	The flow returns to step 1 of Basic Flow.

1.1.8 Extension points

Extension point condition

Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended.
It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description
1.	Timetable associated with employee has been displayed correctly

1.1 Creating new account

1.1.1 Actors

No.	Use-case step description
1.	Technical admin

1.1.2 Purpose and context

New employee has been recruited by company. To efficiently communicate with his new supervisor he needs to have access to the system.

1.1.3 Dependencies

1.1.3.1 Included use-cases

None.

1.1.3.2 Extended use-cases

None.

1.1.4 Assumptions and pre-conditions

No.	Use-case step description

1.	Employee has been recruited.

1.1.5 Initiating business events

New employee has been recruited and needs to start completing tasks.

1.1.6 Basic flow of events

No.	Use-case step description
1.	Admin receives all important data after employee has finished recruitment process.
2	Admin logs into his account
4	After logging in he sees menu only available to admins that allows him to : -create new employee account -remove employee accounts -edit employee accounts depending on which option he chooses
5	Pressing “create new account” button will result in from popping up on screen.
6	Form consists of : - text box labeled “name’ - text box labeled “surname” - 3 text boxes labeled “date of birth(dd/mm/yyyy)” - text box labeled “pesel” - text box labeled “email” - selection field labeled “type of contract” , with two options : fulltime and parttime
7	After filling all relevant data admin submits form by selecting button labeled “create” , depending on data he received he can skip some fields . Only three fields are required : name , surname and email.
8	In case any field was filled incorrectly or required field left empty system displays error message that specifies what went wrong.

9	If there are no errors system displays “account creation complete”
10	After finalizing creation process system sends new user a randomly generated password on his email address.

1.1.7 Alternative flows of events

1.1.7.1 Incorrect type of data in date text fields (a)

No.	Use-case step description
7.	After detecting that there is a symbol not recognized as a number or that a number is to great or small to be correct in date text field .
.	System displays an error “date is incorrect”
	Flow returns to step 6 of basic flow

1.1.7.2 Wrong selection during creation dialog (b)

No.	Use-case step description
4.	Admin accidently chose a wrong option and started a wrong process like edition or removal of account.
.	wrong form gets displayed
	User presses “cancel” button
	Flow returns to step 4 of basic flow

1.1.8 Extension points

Extension point condition

Either steps or rather some conditions which unambiguously determine at which point the specific use-case may be extended.
It is recommended to use flexible condition so that the extending use-cases may be developed independently from the base use-case.

1.1.9 Post-conditions

No.	Use-case step description
1.	Account with correct credentials has been created.

LAB 2

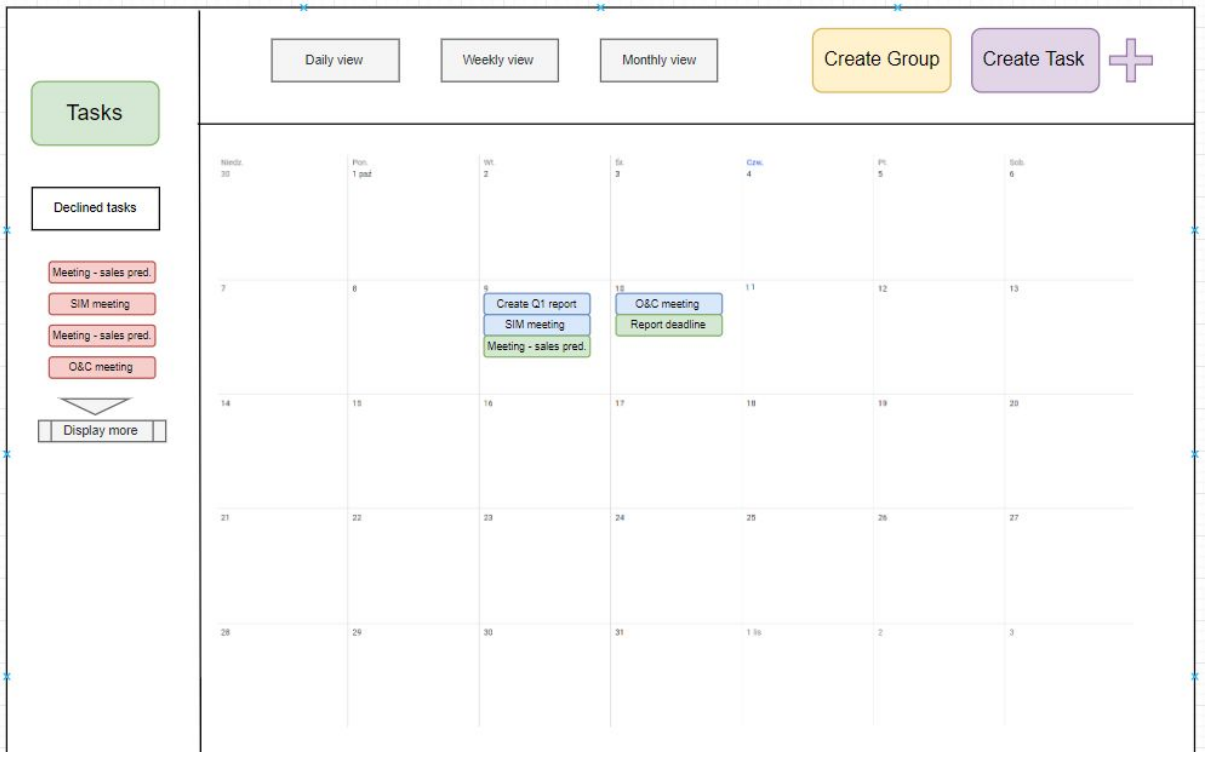
1.

Languages:
frontend: javascript, CSS, HTML, React (framework)
reason: We used javascript because we would like to have responsive page, but also we think those languages are simple, well known in the programming world - it will be easier to find a right programmer. We think React is also a good option, because it will make our application well structured and more maintainable.
backend: Java, mySQL
reason: Java is nowadays the most popular, well known language - it will be easy to find educated specialists. Furthermore, our IT project managers know Java best out of other programming languages and they can supervise programmers better. mySQL is easy to use and has a lot of functionalities that help programmers to maintain the database. Just like with Java - we know it better.
others: Vertabelo for generating database entities, Draw.io for creating mock-ups of our application

Libraries:
jQuery (we will use it for its versatility, contains a lot of features that might help us and speed up our work), Apache Tomcat (server), javax/javax.servlet (for creating servlets), java.sql (for connecting to a database), java.util (useful functionalities, data structures)
Tomcat is incredibly lightweight and highly flexible.It will make our server more stable and has features related to security, which is very important for our project as the data entered can be confidential.

2.

2.1. Home screen:

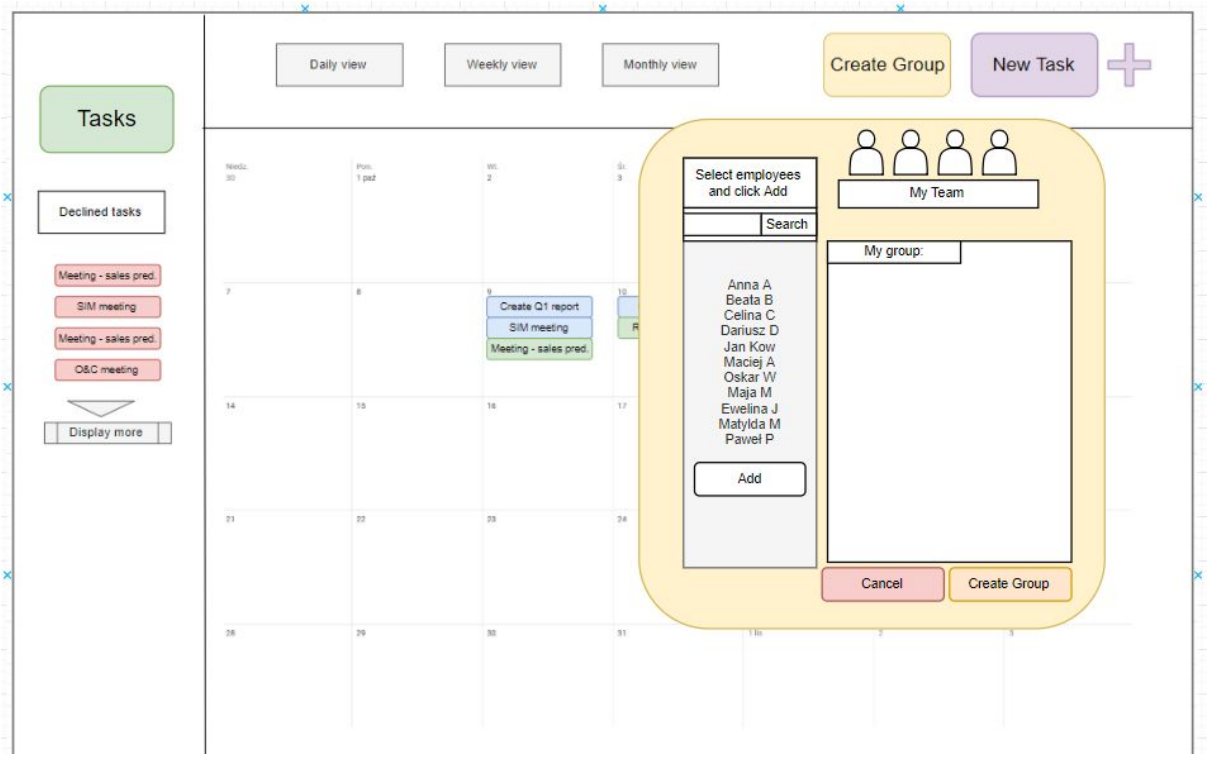


It is possible to:

- see tasks (accepted (green) and not accepted yet (blue) in the timetable (calendar) and declined on the left (red))
- create tasks
- create group
- see newly assigned tasks (in the Tasks button)
- change view (daily, weekly, monthly)

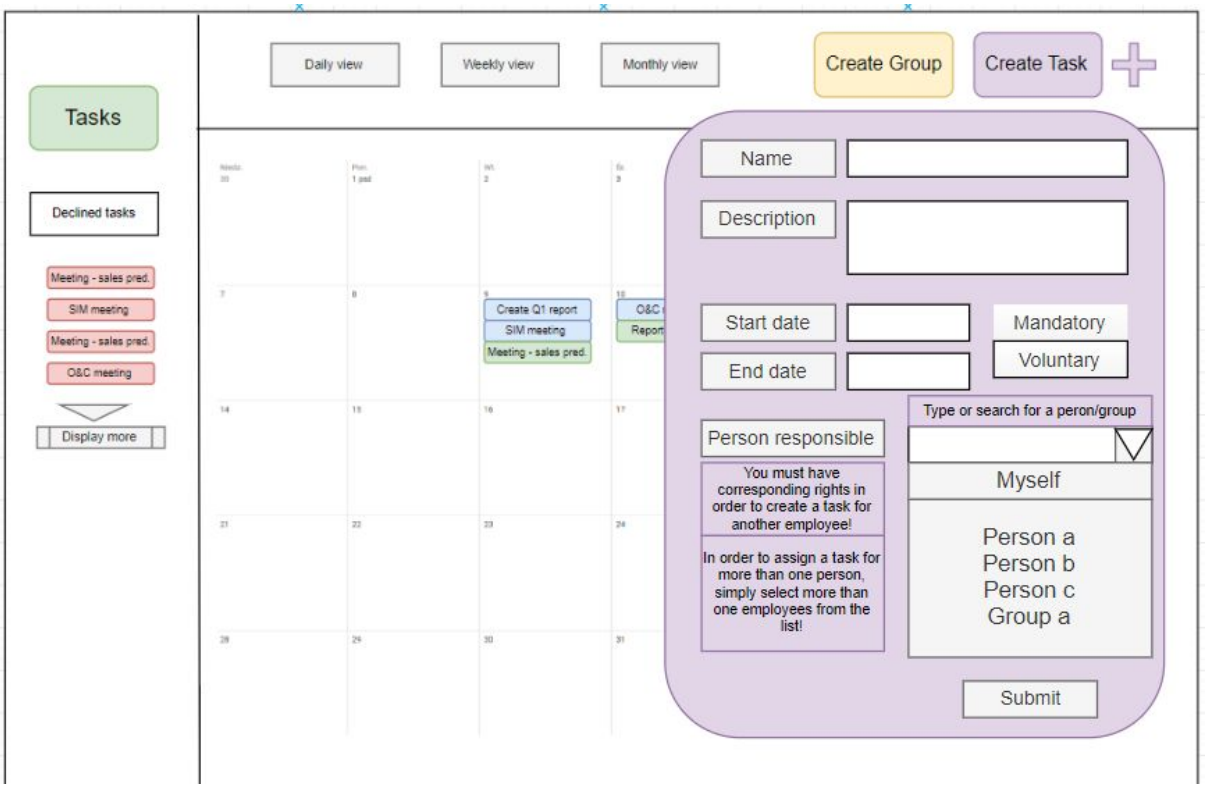
2.2. Creating a group

A user has to click the Create Group button and a window appears:



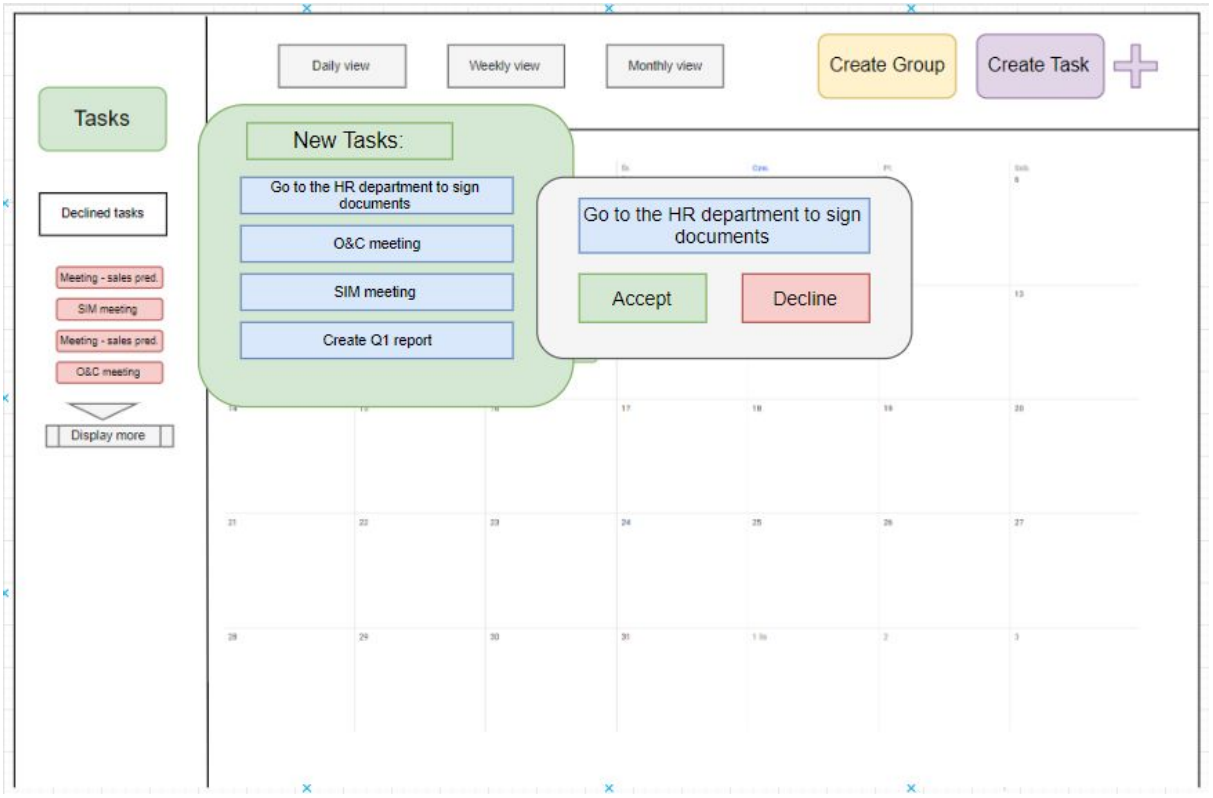
2.3. Creating a task

The user has to click the Create Task button:

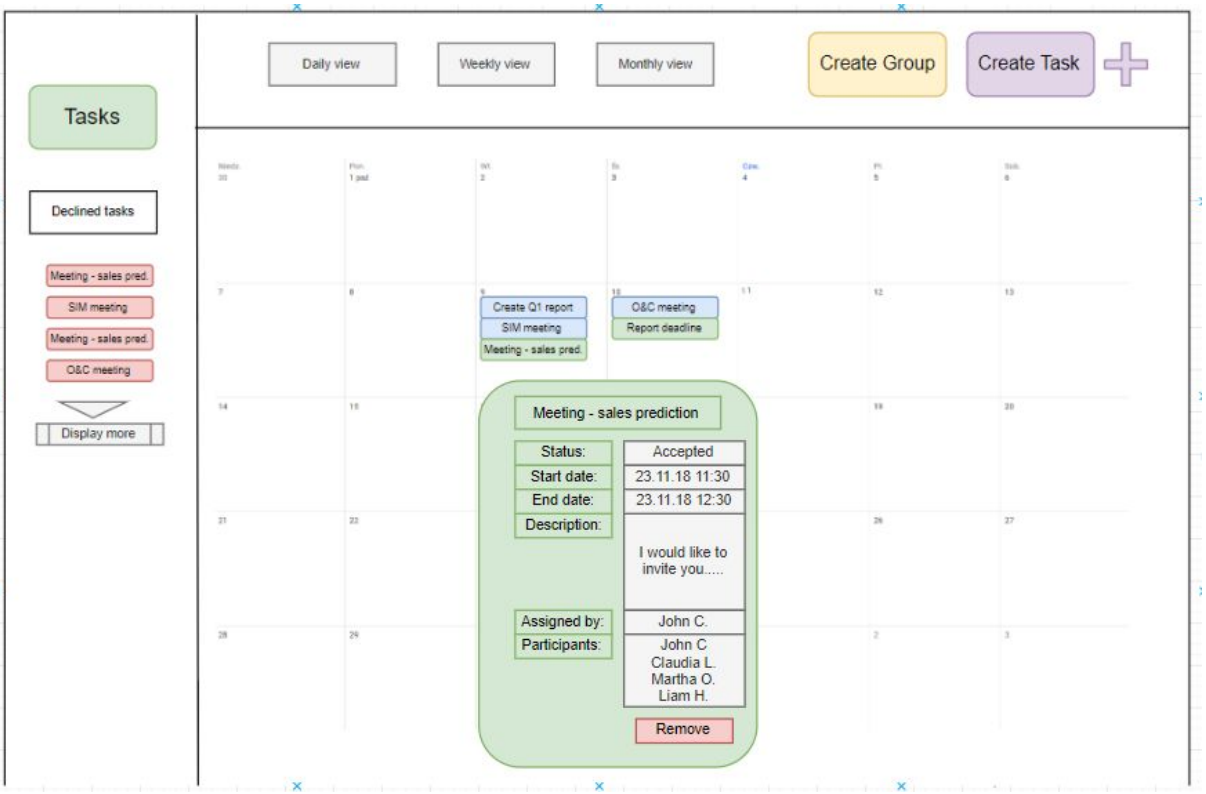


2.4. Accepting/declining a task

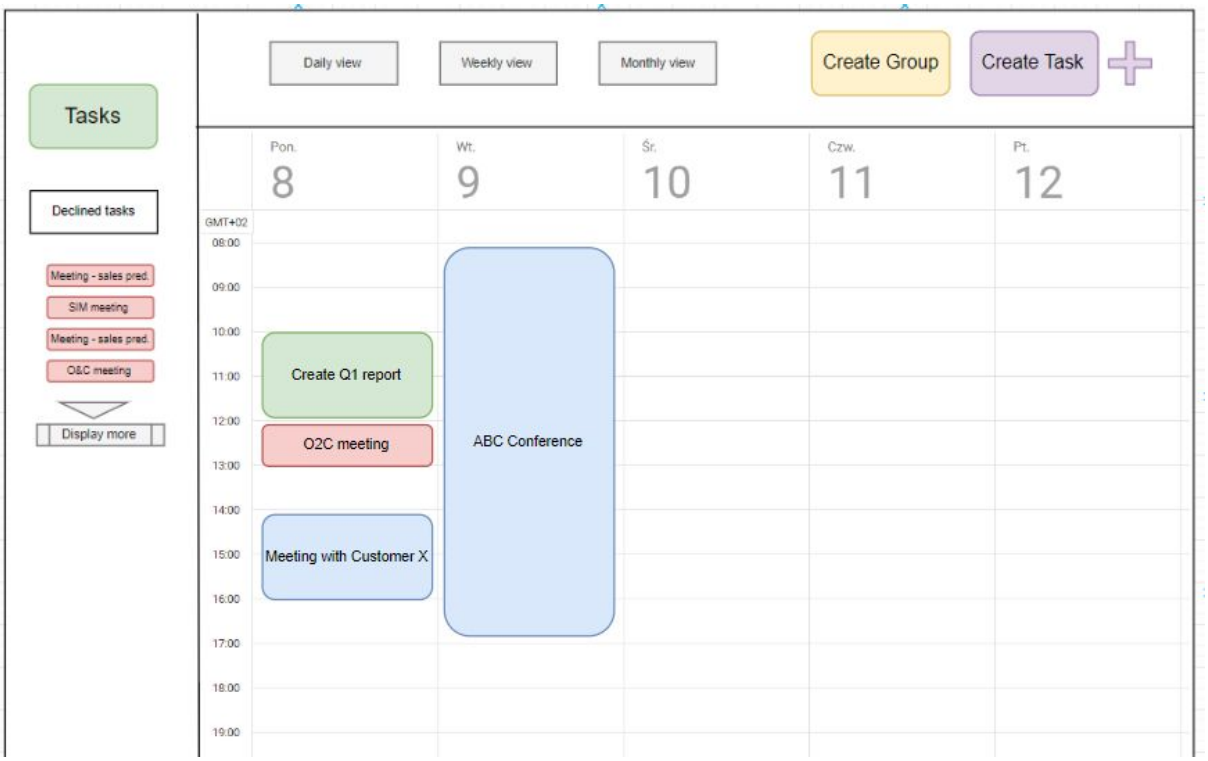
A user has to click the Tasks button:



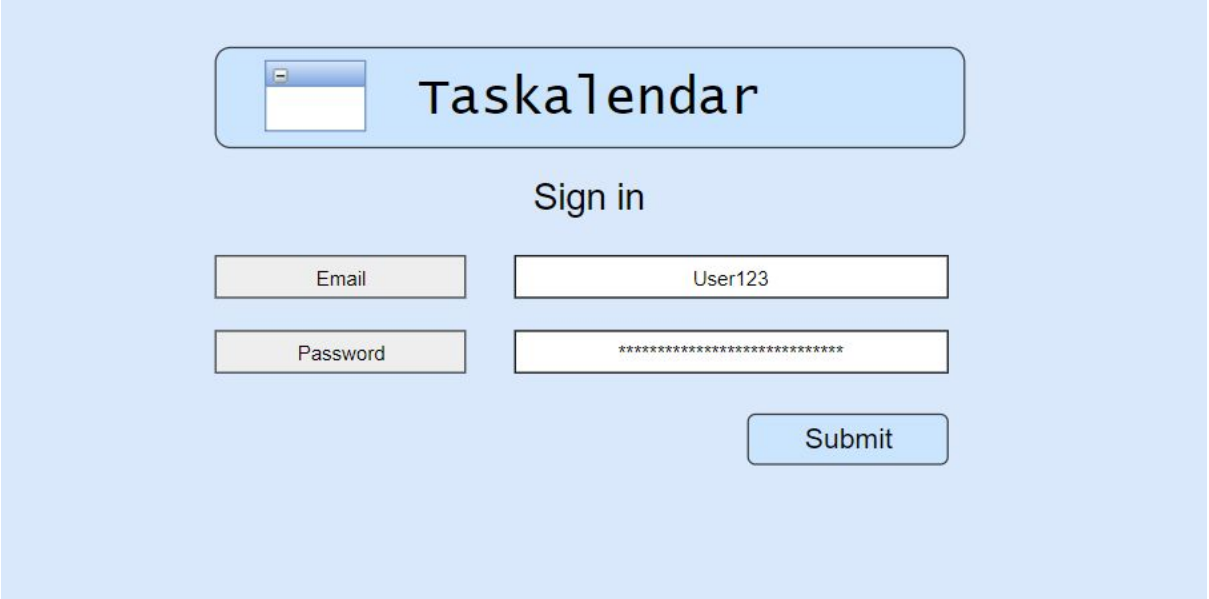
2.5. Viewing/ removing task from the calendar view (removing - only if the person viewing is the person, who assigned the task) after hovering over the task



2.6. Changing View (from monthly to weekly) after clicking Weekly view button



2.7. Logging in



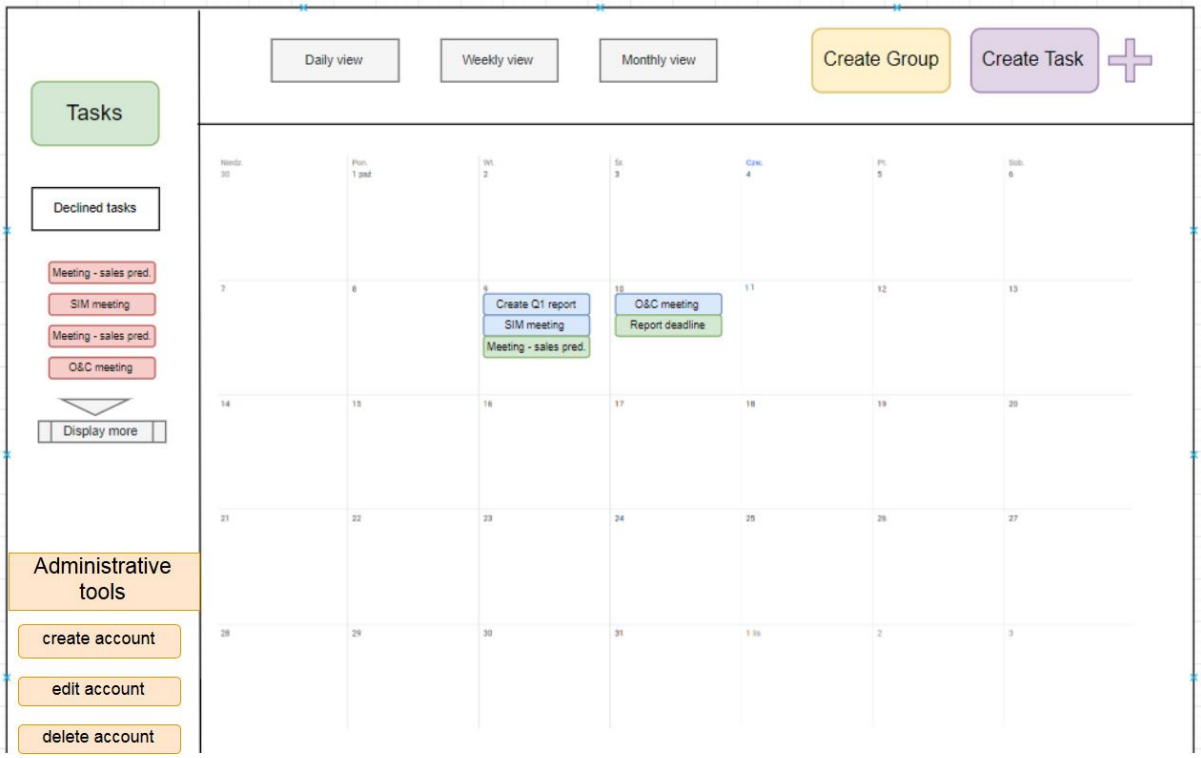
2.8 Admin’s home screen

In addition to regular employees home screen admin can:

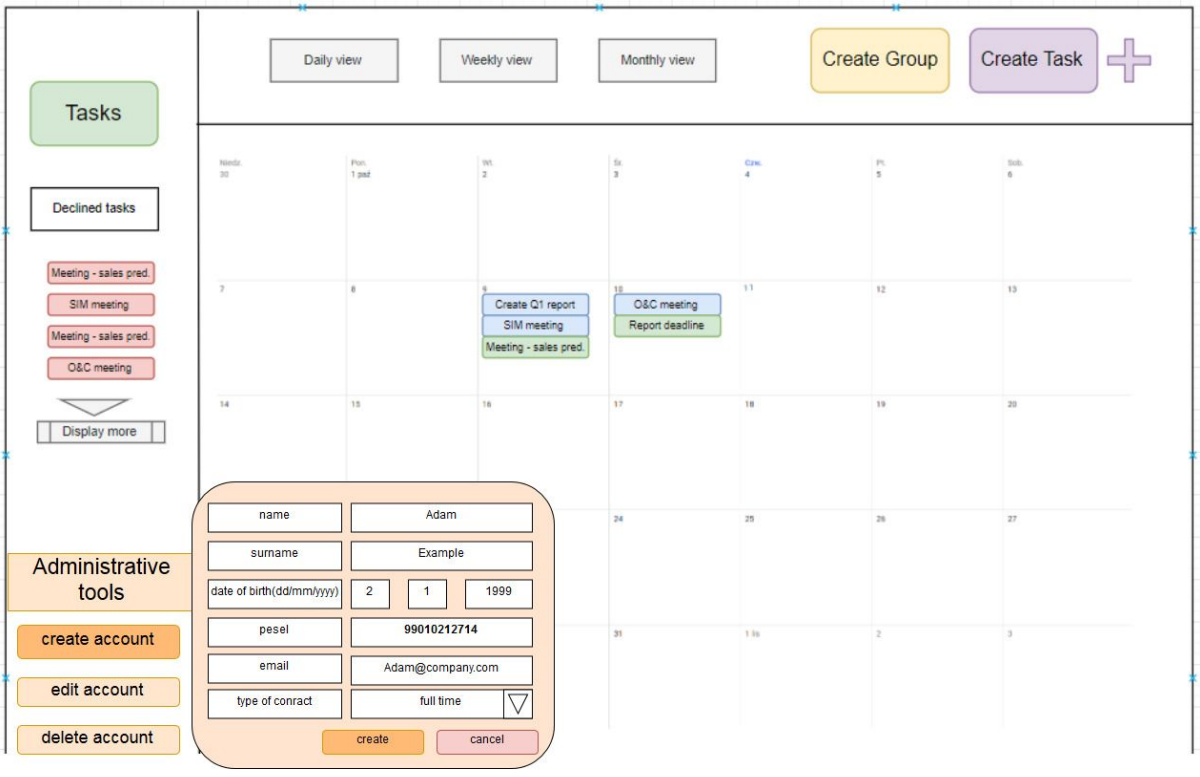
-create new accounts

-edit existing accounts

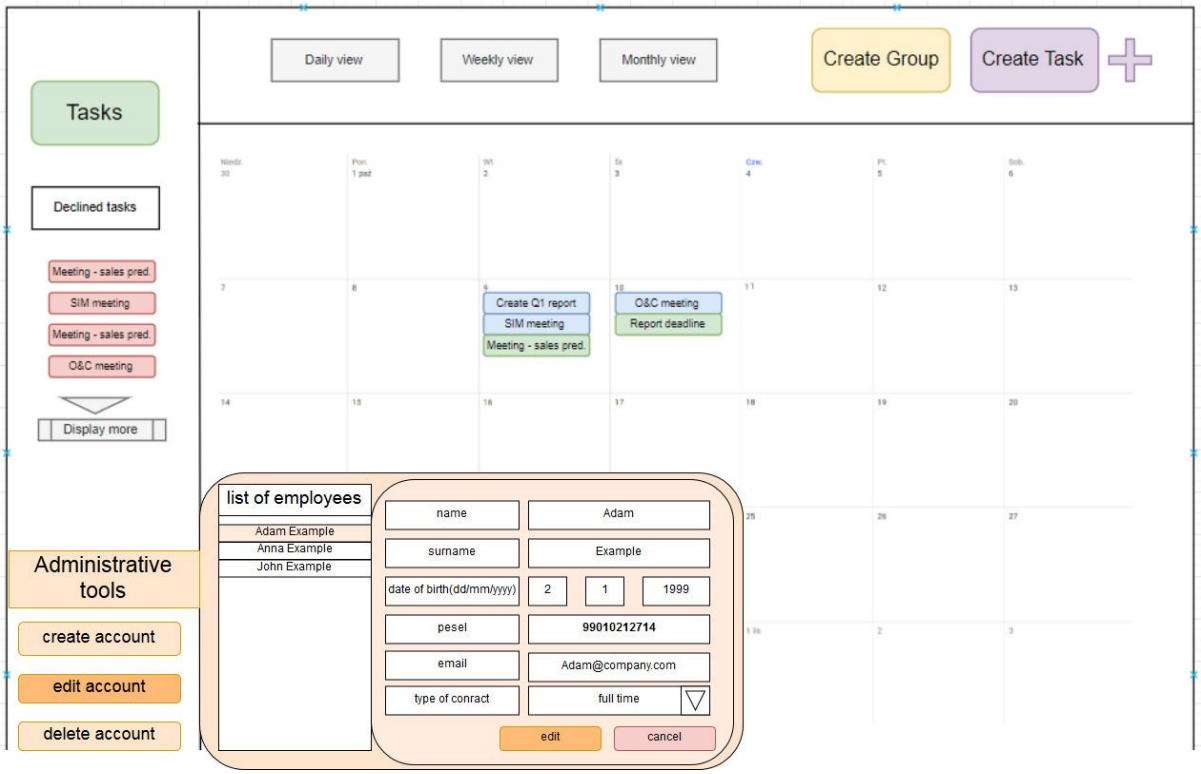
-delete existing accounts



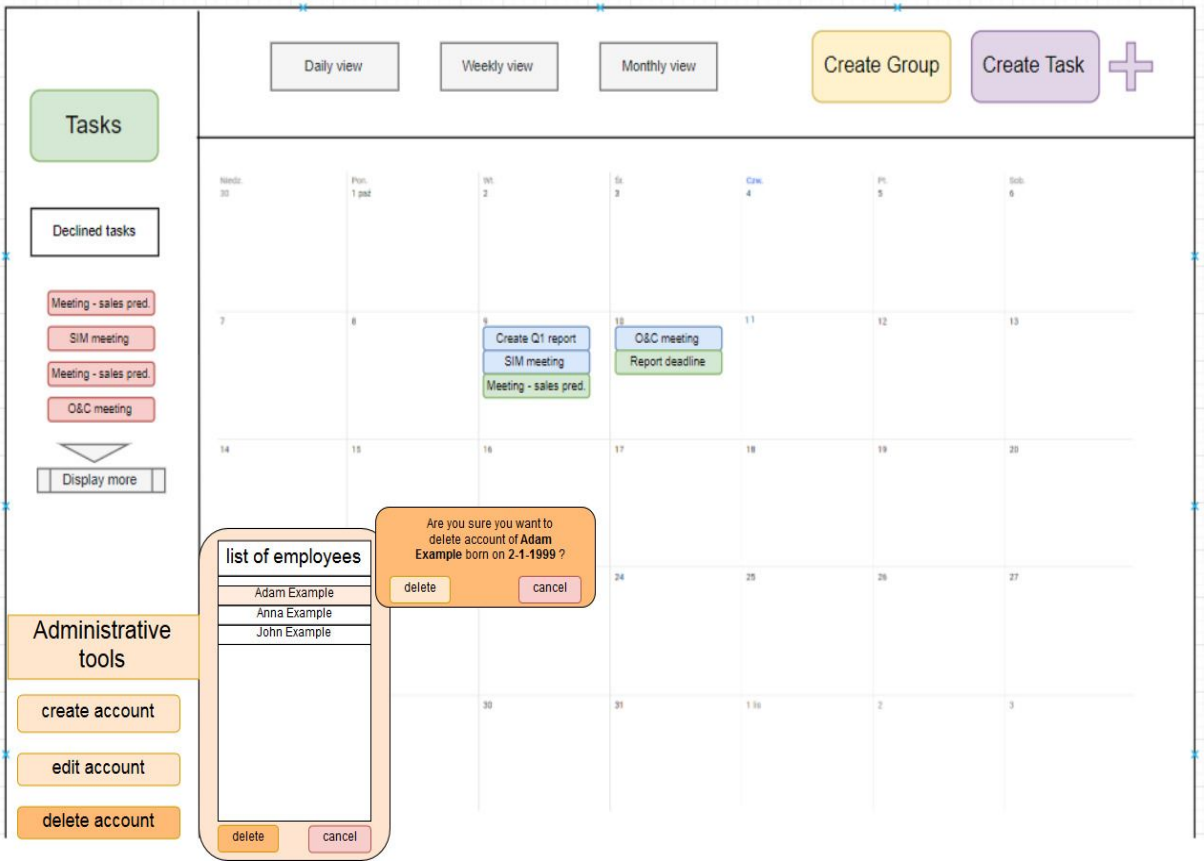
2.9 Create account



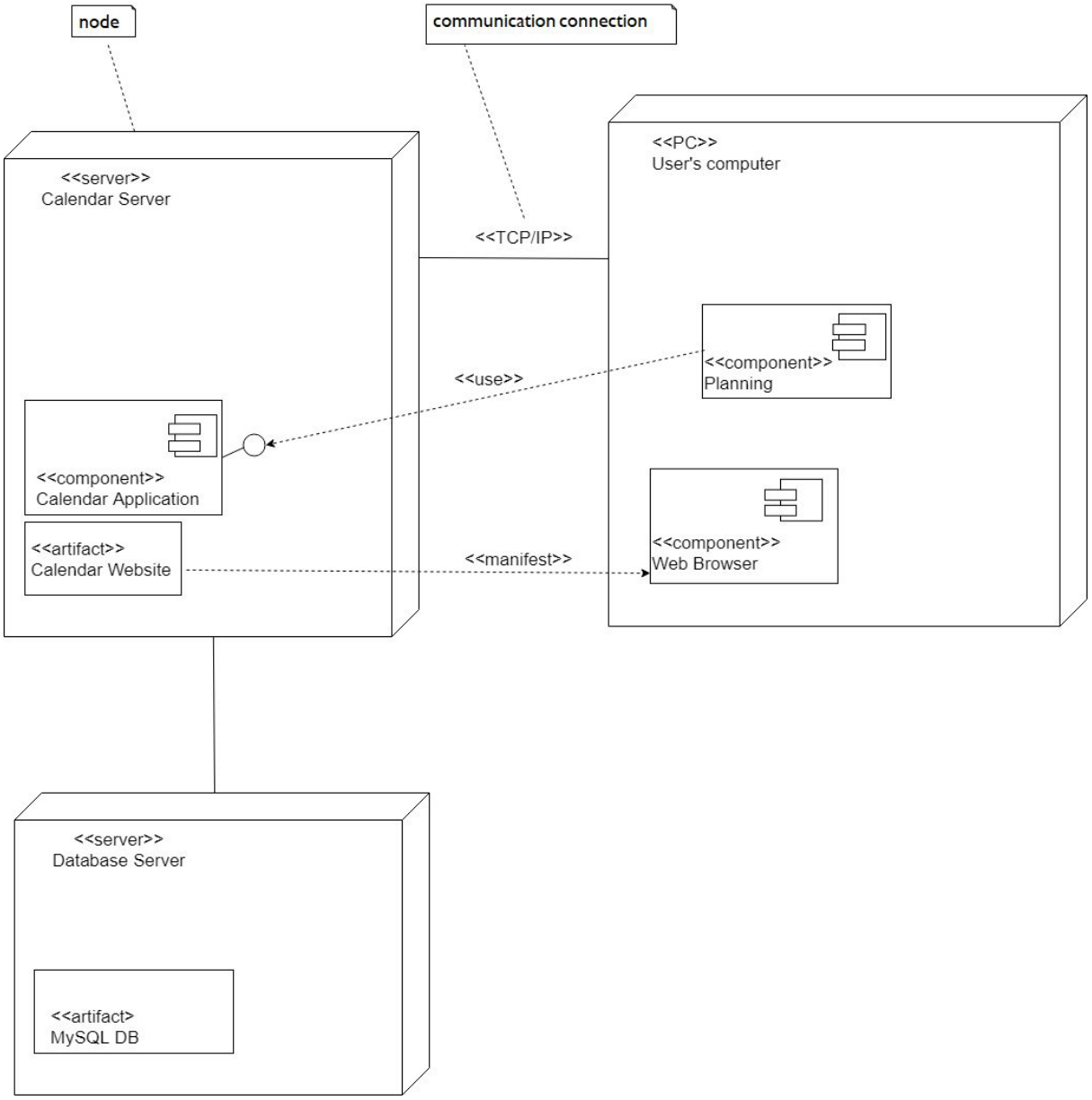
2.10 Edit account



2.11 Delete account



3. DEPLOYMENT DIAGRAM


node
communication connection
<<server>>
Calendar Server
<<PC>>
User's computer
<<TCP/IP>>
<<component>>
Planning
<<use>>
<<component>>
Calendar Application
<<artifact>>
Calendar Website
<<manifest>>
<<component>>
Web Browser
<<server>>
Database Server
<<artifact>
MySQL DB

4. Class diagram