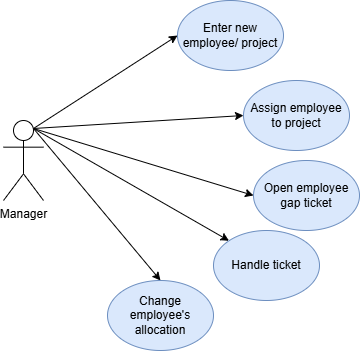
**User scenarios:**

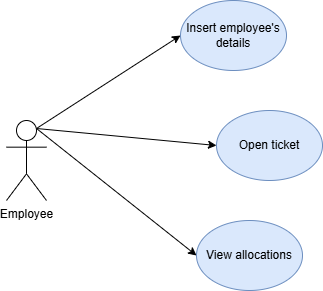
**User profile:**

The system users are project managers / human resource managers. Technological skills: good knowledge of project management tools. Works in a mid-to-large company with multiple on-going projects and employees (even projects and employees around the world).

In some of those companies' employees work on more than one project simultaneously.

Some users will have good computer skills, but some users will be far from the computer world, so the system must be very user friendly.





תמונה שמכילה קו, תרשים, עיגול, עיצוב

התיאור נוצר באופן אוטומטי

**User scenarios:**

1. **Enter a new employee/project:**

**Description:**

The manager enters a new employee/ project to the system- with all its attributes.

**Actor:**

The manager.

**Pre- conditions:**

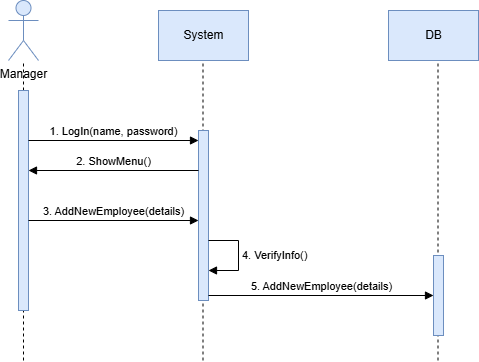
* The manager has access to the system, and is logged in.
* The employee/ project is not in the system.
* All the attributes are correct and logical (hours, age…)

**Post- conditions:**

* The new employee/ project is in the system.
* The manager can see the new employee/ project and assign it.

**Main scenario:**

1. The manager logs in to the system.
2. The manager selects the option of adding new employee/ project.
3. The manager enters all the requested attributes.
4. The system checks for correctness of the fields.
5. The system saves the new employee/ project and displays it to the manager.

****

1. **Assign employee to project:**

**Description:**

The manager assigns employee to existing project.

**Actor:**

The manager.

**Pre- conditions:**

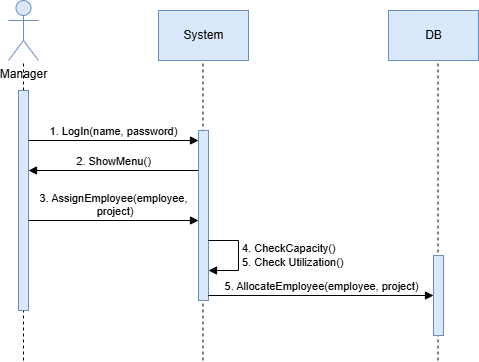
* The manager has access to the system, and is logged in.
* The employee and project are in the system.
* The employee's utilization is suitable for the project's requirements.

**Post- conditions:**

* The employee is assign to the project
* The employee's utilization updates and the project's requirements also.

**Main scenario:**

1. The manager logs in to the system.
2. The manager selects the option of assigning employee to a project.
3. The manager enters the relevant employee and project.
4. The system checks for employee's utilization and project's capacity and approve the assignment.
5. The system saves the new assignment and displays it to the manager.

****

1. **Open employee gap "ticket"**

**Description:**

The manager open a ticket of employee's absence.

**Actor:**

The manager.

**Pre- conditions:**

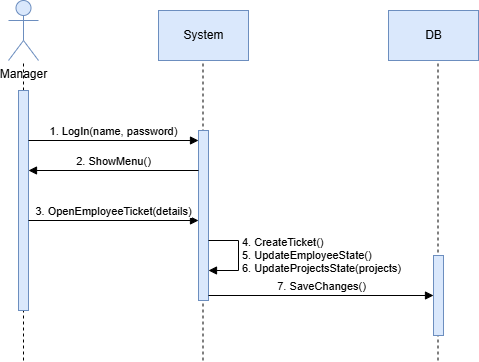
* The manager has access to the system, and is logged in.
* The employee is in the system.

**Post- conditions:**

* The ticket is in the system and visible to the manager.
* The gaps in the projects where the employee is assigned are visible to the manager in the gaps page.

**Main scenario:**

1. The manager logs in to the system.
2. The manager selects the option of opening new employee's absence ticket.
3. The manager enters the relevant employee, dates and reason of leaving.
4. The system creates gaps tickets for each project the employee is assign to, with details of the gaps.

****

1. **Handle ticket:**

**Description:**

The manager closes ticket by changing the assign of the employees between the projects.

**Actor:**

The manager.

**Pre- conditions:**

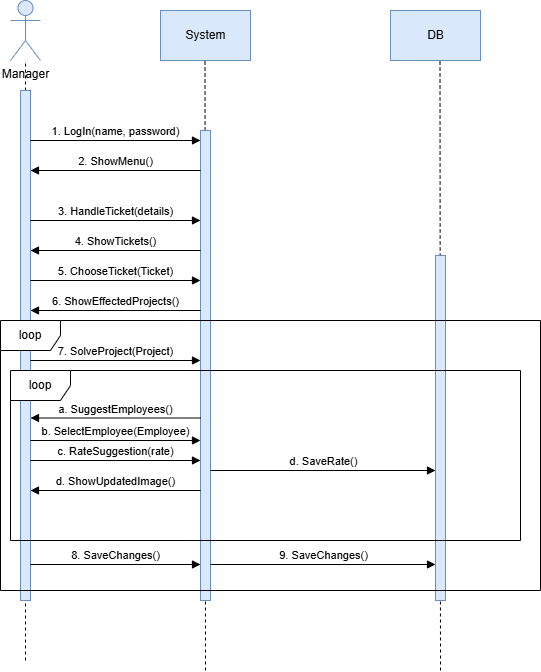
* The manager has access to the system, and is logged in.
* A gap ticket is open and visible to the manager.

**Post- conditions:**

* Optional: the ticket is closed.
* Optional: new gap ticket is created.
* Changes of employees assignments are updated in the system and visible to the manager.

**Main scenario:**

1. The manager logs in to the system.
2. The manager selects the option of view gaps tickets.
3. The system presents all the gaps tickets.
4. The manager selects option of solving a ticket for a specific ticket.
5. The system shows all the projects effected from the gap according to their priorities.
6. The manager goes into each project and enter the priorities of the project's attributes.
   1. The system suggests employees to fill the gaps based on the priorities of the places with gaps in the project.
   2. The manager chooses the right employee and assigns it to the specific part of the project.
   3. The manager rates the suggestion of the system.
   4. The systemsaves the rate in the DB.
   5. The system updates the change and shows the manager update image of the employee's assignment, projects gaps and utilization.
7. Do a-c until the manager saves the changes.



1. **Change employee's allocation:**

**Description:**

The manager changes his employee's allocation between projects manually.

**Actor:**

The manager.

**Pre- conditions:**

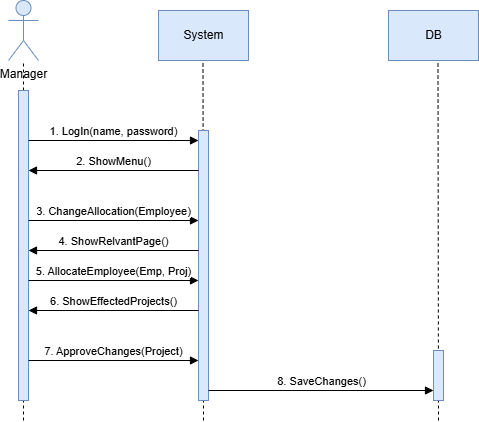
* The manager has access to the system, and is logged in.
* The relevant employee and project are existing in the system and visible to the manager.

**Post- conditions:**

* Changes of employee's assignments are updated in the system and visible to the manager.

**Main scenario:**

1. The manager logs in to the system.
2. The manager selects the option of changing employee allocation.
3. The manager chooses the relevant employee and allocates it to the new project.
4. The system shows the effect of the actionon the employee and all the projects.
5. The manager approves the changes.
6. The system updates the new allocation.

****

**Employee Use Cases:**

1. **Insert Employee details:**

**Description:**

The employee enters his details to the system- with all its attributes.

**Actor:**

The employee

**Pre- conditions:**

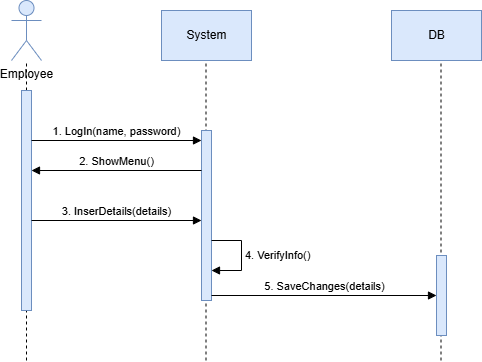
* The employee has access to the system, and is logged in.
* The employee is not in the system.
* All the attributes are correct and logical (hours, age…)

**Post- conditions:**

* The employee's details are in the system.
* The manager can see the new employee/ project and assign it.

**Main scenario:**

1. The employee logs in to the system.
2. The employee selects the option of inserting employee's details.
3. The employee enters all the requested attributes.
4. The system checks for correctness of the fields.
5. The system saves the new details and displays them to the manager.



1. **Open ticket:**

**Description:**

The employee opens a ticket for his absence.

**Actor:**

The employee.

**Pre- conditions:**

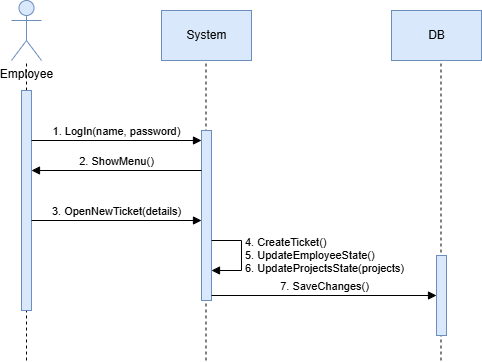
* The employee has access to the system, and is logged in.
* The employee's details are in the system.

**Post- conditions:**

* The ticket is in the system and visible to the manager.
* The gaps in the projects where the employee is assigned are visible to the manager in the gaps page.

**Main scenario:**

1. The employee logs in to the system.
2. The employee selects the option of opening a new absence ticket.
3. The employee enters the relevant details such as dates and reason for leaving.
4. The system creates gaps tickets for each project the employee is assigned to, with details of the gaps.



1. **View employee's allocation to projects:**

**Description:**

The employee views his allocation to projects- days of week, language, and percentages of work.

**Actor:**

The employee.

**Pre- conditions:**

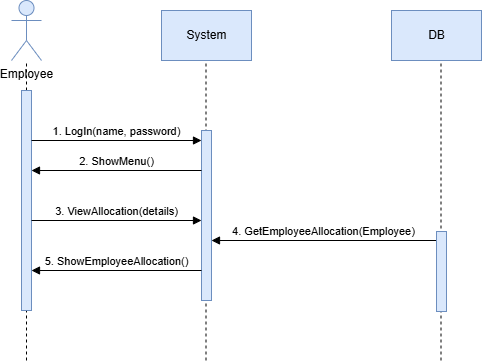
* The employee has access to the system, and is logged in.
* The employee's details are in the system.

**Post- conditions:**

* The employee gets a page with all the projects he is allocated to with all their details.

**Main scenario:**

1. The employee logs in to the system.
2. The employee selects the option of view his current allocation status.
3. The system loads a page with all the employee's allocated projects and details.



**Big Management Use Cases:**

1. **View statistics:**

**Description:**

The manager views details about all the employees, the working hours for each project and more relevant statistics.

**Actor:**

The Big Manager.

**Pre- conditions:**

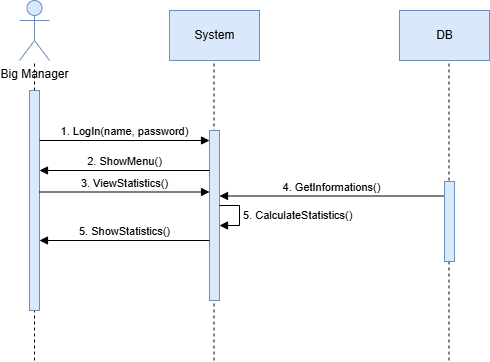
* The Big Manager has access to the system, and is logged in.

**Post- conditions:**

* The Big Manager gets a page with all the relevant statistics.

**Main scenario:**

1. The Big Manager logs in to the system.
2. The Big Manager selects the option of view statistics.
3. The system loads a page with all the relevant statistics.

****