**Use case template**

Descriptions of template fields:

* **ID and name:** Title should be descriptive and should usually begin with a verb, e.g. order, calculate, input, etc. ID can have any format but must be unique among all use cases.
* **Primary actor:** Person that wishes to accomplish a goal through the use of the system. Only a single primary actor per use case.
* **Secondary actors:** Actors that have an interest in the completion of the goal but that do not directly interact with the system.
* **Description:** Concise description of the purpose of the use case.
* **Trigger:** Condition internal or external to the system that prompts the use case to start.
* **Preconditions:** Conditions that must be true before the use case starts. Each should be labeled with an ID unique to the use case.
* **Postconditions:** Conditions that must be true after the use case ends normally. Each should be labeled with an ID unique to the use case.
* **Normal flow:** Detailed step-by-step description of the logical flow of the use case. It should describe an explicit two way interaction, with the system prompting for input and the actor responding accordingly. Each step should be numbered.
* **Alternative flows:** Flows that achieve the same goal as the normal flow but are expected to be less common or lower priority.
* **Exceptions:** Conditions that result in the normal flow ending prematurely due to an unrecoverable condition in the system. The condition that causes the flow should be clearly stated, as should be any other decisions that the actor must make in this situation.

|  |  |
| --- | --- |
| **ID and name** | UC-1: Login |
| **Primary actor** | Employee |
| **Description** | An employee needs to login to use the system using a username and a password. |
| **Trigger** | A company employee indicates that he wants to login into the system. |
| **Preconditions** | PRE-1: Company employee has the system app installed on computer.  PRE-2: Username and password of the company employee are stored in a database. |
| **Postconditions** | POST-1: The employee company is logged in and his window’s open. |
| **Normal flow** | 1. **Login to system** 2. Company Employee opens the system’s app. 3. He introduced his username and password. 4. The employee presses the login button. 5. His logged in and can use the system on its own window. |
| **Alternative flows** |  |
| **Exceptions** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-2: Assign task | | |
| **Primary actor** | Boss | **Secondary actors** | Worker |
| **Description** | The boss of the company wants to assign a task to one of the workers that is present at the job at that time. | | |
| **Trigger** | The boss selects a worker from the list of workers. | | |
| **Preconditions** | PRE-1: The boss is logged in to the system.  PRE-2: The worker that he wants to assign work is present. | | |
| **Postconditions** | POST-1: The worker will see in his window the new task.  POST-2: The worker can start the new task. | | |
| **Normal flow** | 1. **Assign a task to a worker** 2. The boss selects the worker. 3. He inserts the task description in the text-area. 4. He presses the send button. 5. Task will appear in worker list. | | |
| **Alternative flows** |  | | |
| **Exceptions** | **1.0.E1 Forgets to select a worker**  1. System will display an error message that “No worker is selected”.  **1.0.E2 Forgets to write a task**  1. System will display an error message that “No task is written”.  **1.0.E3 Tries to send a task to a worker that is not present**  1. System will display an error message that “This worker is not present”. | | |
| **ID and name** | UC-3: Mark a task as solved | | |
| **Primary actor** | Worker | **Secondary actors** | Boss |
| **Description** | A worker finished its task and wants to mark it as done in his tasks list. | | |
| **Trigger** | The worker press to change the status of a task. | | |
| **Preconditions** | PRE-1: The worker is logged in to the system.  PRE-2: The worker has that task assigned.  PRE-3: The worker has finished the task. | | |
| **Postconditions** | POST-1: The task will appear as done in worker and boss list.  POST-2: The boss can remove that task from the worker. | | |
| **Normal flow** | 1. **Mark a task as done** 2. The worker searches for the task he wants to mark as done. 3. He changes the status of the task in done. | | |
| **Alternative flows** |  | | |
| **Exceptions** |  | | |

|  |  |
| --- | --- |
| **ID and name** | UC-4: See tasks owned |
| **Primary actor** | Worker |
| **Description** | A worker logs in in the system and he is on his own window. |
| **Trigger** | The worker logs in. |
| **Preconditions** | PRE-1: The worker is logged in to the system.  PRE-2: The worker window opened. |
| **Postconditions** | POST-1: The task will appear in worker’s list. |
| **Normal flow** | 1. **View own tasks** 2. Worker is on his own window and see his tasks in a table and its status. |
| **Alternative flows** | **2.0 There are no tasks available**  1. The worker will see an empty table that will populate when his boss assigns work to him. |
| **Exceptions** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5: Remove task | | |
| **Primary actor** | Boss | **Secondary actors** | Worker |
| **Description** | The boss wants to remove a task from a worker that is marked as done. | | |
| **Trigger** | The boss sees the task marked as done. | | |
| **Preconditions** | PRE-1: The worker has marked that task as done. | | |
| **Postconditions** | POST-1: The task will disappear from worker’s list and from the list of done tasks for that worker. | | |
| **Normal flow** | **1.0 Remove a task**   1. The boss opens the done-tasks list for a worker by double-clicking on it. 2. He selects the task he wants to remove and press the remove button. | | |
| **Alternative flows** | **2.0 There are no done-tasks available**  1. The Boss will see an empty table that will populate when the worker will mark a task as done. | | |
| **Exceptions** |  | | |

|  |  |
| --- | --- |
| **ID and name** | UC-6: Add worker |
| **Primary actor** | Boss |
| **Description** | The boss wants to add a new worker to the system. |
| **Trigger** | The boss intents to add a new worker. |
| **Preconditions** | PRE-1: The boss is logged in to the system. |
| **Postconditions** | POST-1: The internal system will generate username and password for the new worker and his own window. |
| **Normal flow** | **1.0 Add a new worker**   1. The boss inserts a new worker’s name in the “Employee name” text field. 2. He presses “Add worker” button. |
| **Alternative flows** | **2.0 There is already a worker with that name**  1. The email and password for the new worker will be different from the ones of the worker already existing. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **ID and name** | UC-7: Delete worker |
| **Primary actor** | Boss |
| **Description** | The boss wants to delete a worker from the system. |
| **Trigger** | The boss intents to delete a worker. |
| **Preconditions** | PRE-1: The boss is logged in to the system. |
| **Postconditions** | POST-1: The deleted worker will no longer have access to the system.  POST-2: The deleted worker will not appear anymore in boss window. |
| **Normal flow** | **1.0 Delete a worker**   1. The boss selects a worker from his window. 2. He pressed the delete button. |
| **Alternative flows** |  |
| **Exceptions** | **1.0E1 Forgets to select a worker to delete before pressing the delete button**  1. System will display an error message to the boss to select a worker before pressing delete. |

|  |  |
| --- | --- |
| **ID and name** | UC-8: Update worker |
| **Primary actor** | Boss |
| **Description** | The boss wants to update a worker from the system. |
| **Trigger** | The boss intents to update a worker. |
| **Preconditions** | PRE-1: The boss is logged in to the system. |
| **Postconditions** | POST-1: The updated worker will receive new username and passwords generated by the system. |
| **Normal flow** | **1.0 Update a worker**   1. The boss selects a worker from his window. 2. He inserts a new name in the text field. 3. He pressed the update button. |
| **Alternative flows** |  |
| **Exceptions** | **1.0E1 Forgets to select a worker to update before pressing the update button**  1. System will display an error message to the boss to select a worker before pressing update. |

|  |  |
| --- | --- |
| **ID and name** | UC-9: See all workers |
| **Primary actor** | Boss |
| **Description** | The boss wants to see all his workers. |
| **Trigger** | The boss logs in. |
| **Preconditions** | PRE-1: The boss is logged to the system.  PRE-2: The boss’s window is opened. |
| **Postconditions** | POST-1: The boss will see the list of workers. |
| **Normal flow** | **1.0 See the list of workers**  1. The boss is on his window and sees the list of workers. |
| **Alternative flows** | **2.0 There is no worker on list**  1.0 The boss will see an empty list. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **ID and name** | UC-10: See done tasks of a worker |
| **Primary actor** | Boss |
| **Description** | The boss wants to see the tasks of a worker that are done. |
| **Trigger** | The boss intents to see the task that a worker finished. |
| **Preconditions** | PRE-1: The boss is on his page of the system. |
| **Postconditions** | POST-1: The boss will see the list of done tasks of a worker. |
| **Normal flow** | **1.0 See the list of done tasks**   1. The boss double-clicks on a worker from his list. 2. The list of finished tasks is displayed. |
| **Alternative flows** | **2.0 There is no finished tasks for the selected worker**  1.0 The boss will see an empty list. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **ID and name** | UC-11: Log out |
| **Primary actor** | Employee |
| **Description** | An employee wants to log out from the system. |
| **Trigger** | The employee intents to log out. |
| **Preconditions** | PRE-1: The employee is on his window. |
| **Postconditions** | POST-1: The employee is logged out and his window is closed.  POST-2: His status will appear as absent in Boss window. |
| **Normal flow** | **1.0 Log out from the system**   1. The employee pressed the log out button. 2. His status is changed in the boss’s window. 3. His window is closed. |
| **Alternative flows** |  |
| **Exceptions** |  |