|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-1: Login | | |
| **Primary actor** | Employee | **Secondary actors** |  |
| **Description** | An employee uses his credentials to start work either from within the corporation or from home | | |
| **Trigger** | The employee starts his work schedule | | |
| **Preconditions** |  | | |
| **Postconditions** | POST-1: The employee is logged in | | |
| **Normal flow** | 1. **Valid Login** 2. The employee opens the application. 3. The employee enters his authentication data. 4. The application displays a confirmation message. 5. The employee sees his menu window. | | |
| **Alternative flows** |  | | |
| **Exceptions** | 1. **E1 The employee enters the wrong authentication data** 2. The application displays an error message. 3. Return to step 2 of normal flow. | | |

**Custom PC Builder**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-2: Login | | |
| **Primary actor** | Client | **Secondary actors** |  |
| **Description** | A client uses his credentials to enter the application in order to place/see an order for his PC | | |
| **Trigger** | The client opens application | | |
| **Preconditions** |  | | |
| **Postconditions** | POST-1: The client is logged in | | |
| **Normal flow** | 1. **Valid Login** 2. The client opens the application. 3. The client enters his authentication data. 4. The application displays a confirmation message. 5. The client sees the main menu window. | | |
| **Alternative flows** | **1.1 Not signed in:**  1. The client clicks on “Sign in” to create a new account  2. The application displays the Sign in window  3. The client enters his personal data to create an account  4. Account created successfully  5. Return to step 2 of normal flow | | |
| **Exceptions** | 1. **E1 The client enters the wrong authentication data** 2. The application displays an error message. 3. Return to step 2 of normal flow. 4. **E2 The client enters invalid personal data** 5. The application displays an error message. 6. Return to step 2 of alternative flow 1.1 | | |
| **ID and name** | Login | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-3:Order submission | | |
| **Primary actor** | Client | **Secondary actors** |  | |
| **Description** | A client wants to build a custom PC,enters his budget, category, description and the name of the project and places the order | | |
| **Trigger** | A client indicates that he wants to place an order | | |
| **Preconditions** | PRE-1: The client is logged in | | |
| **Postconditions** | POST-1: Build order is stored into the client`s history list with “Waiting” status  POST-2: All employees who work in the order category and are not working on an order will see the order in their list of orders | | |
| **Normal flow** | 1. **Order a PC build:** 2. The client enters his budget, category, description and name of the project 3. The application stores the order into the client`s history list with “Waiting status” 4. The application updates the list of orders for all the employees who work in the order category | | |
| **Alternative flows** |  | | |
| **Exceptions** | 1. **E1 The budget is too low:** 2. Each category has a minimum budget, and if the customer entered a budget lower than the minimum budget of the category, then the order cannot be processed 3. The application displays an error message with the minimum budget required 4. Return to step 1 of normal flow 5. E2 The name of the project already exists: 6. To add a gram of originality, each project is unique, and if the customer entered the name of an already existing project, the order cannot be processed 7. The application displays an error message with “The project name already exists” 8. Return to step 1 of normal flow | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-4: Cancel order | | |
| **Primary actor** | Client | **Secondary actors** |  |
| **Description** | Certain circumstances make the client cancel the order, he accesses the order history and selects the order he wants to cancel, then clicks on the “Cancel” button | | |
| **Trigger** | The client presses the “Cancel” button | | |
| **Preconditions** | PRE-1: Client is logged in  PRE-2: The client has placed an order | | |
| **Postconditions** | POST-1: The order is canceled and disappears from the client`s order history  POST-2: The order is deleted from the list of employee orders | | |
| **Normal flow** | 1. Cancel an order: 2. The client asks the application to show him the order history 3. The application swaps to his order history window 4. The client selects the order he wants to cancel and presses the “Cancel” button 5. The order is removed from the customer`s history | | |
| **Alternative flows** |  | | |
| **Exceptions** | 1. The order was taken by an employee: 2. If the order was taken by an employee, when the customer cancels the order, the application will close the employee's window | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5: Modify Order | | |
| **Primary actor** | Client | **Secondary actors** |  |
| **Description** | Certain circumstances make the client modify the order, he accesses the order history and selects the order he wants to modify, then clicks on the “Modify” button | | |
| **Trigger** | The client presses the “Modify” button | | |
| **Preconditions** | PRE-1: Client is logged in  PRE-2: The client has placed an order  PRE-3: The order is still in progress | | |
| **Postconditions** | POST-1: The order is modified from the client`s order history  POST-2: The order is modified from the list of employee orders | | |
| **Normal flow** | * 1. Cancel an order:   1. The client asks the application to show him the order history  2. The application swaps to his order history window  3. The client selects the order he wants to modify and presses the “Modify” button  4. The order is modified from the customer`s history | | |
| **Alternative flows** |  | | |
| **Exceptions** | 1. **E1 The employee already works on his order:** 2. If the order was taken by an employee, the employee window will refresh all his selected components if the budget is still in his working range 3. E2 The modified budget is out of employee working field: 4. If the modified budget is out of employee working field, the application will close employee`s window and update the list of orders for specific budget range | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-6: View Order History | | |
| **Primary actor** | Client | **Secondary actors** |  |
| **Description** | After the customer has logged in, he wants to see his status/previous orders | | |
| **Trigger** | The client presses “History” button | | |
| **Preconditions** | PRE-1: Client is logged in | | |
| **Postconditions** |  | | |
| **Normal flow** | 1. **View Order History:** 2. **The client asks the application to show him the order history** 3. **The application swaps to his order history window** 4. **The client can now see, cancel, modify orders** | | |
| **Alternative flows** |  | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-7: Answer order | | |
| **Primary actor** | Client | **Secondary actors** |  |
| **Description** | After an employee completes a client's order, the client can accept or decline the order | | |
| **Trigger** | Employee completes a client`s order | | |
| **Preconditions** | PRE-1: Client is logged in  PRE-2: The client has placed an order  PRE-3: An employee completes the client`s order | | |
| **Postconditions** | POST-1: Order is in delivery process or canceled | | |
| **Normal flow** | 1. **Accept order:** 2. **An employee completes the client`s order** 3. The application displays a window for the client to answer to order 4. The client accepts the order 5. The application displays a window for the client to enter the delivery data and payment method 6. The client enters the delivery data and selects the payment method 7. The application closes the window and sets the order status to “In progress of delivery” | | |
| **Alternative flows** | 1. The client does not have enough funds: 2. The client selects online payment 3. The client does not have enough funds to pay the order 4. Back to normal flow 4    1. Cancel Order:   1.The client cancel`s the order  2.The application closes the window and sets the order status to “Canceled” | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-8: View Orders list | | |
| **Primary actor** | Employee | **Secondary actors** |  |
| **Description** | After the employee has logged in, he wants to start his work and select an order to start working on it | | |
| **Trigger** | The employee has logged in | | |
| **Preconditions** | PRE-1: The employee is logged in | | |
| **Postconditions** |  | | |
| **Normal flow** | 1. **View orders list:** 2. **The application displays on the employee`s main screen the list of orders for his field** | | |
| **Alternative flows** |  | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-9: Select an order | | |
| **Primary actor** | Employee | **Secondary actors** |  |
| **Description** | After the employee has logged in, he wants to start his work and select an order to start working on it from the list of orders displayed on his main screen | | |
| **Trigger** | The employee selects an order and presses “Build” button | | |
| **Preconditions** | PRE-1: The employee is logged in | | |
| **Postconditions** | POST-1: The order status is set from “Pending” to “In progress” | | |
| **Normal flow** | * 1. Selects an order:  1. The employee selects an order from the main screen and presses “Build” button 2. The application displays the custom build window for the employee with his current order description and left budget 3. The order status is set from “Pending” to “In progess” 4. The employee is now working on an order | | |
| **Alternative flows** |  | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-10:Cancel an order | | |
| **Primary actor** | Employee | **Secondary actors** |  |
| **Description** | After the employee has started working on an order, he realizes that he no longer has the necessary components for the order and decides to cancel the client's order, giving him the reason for canceling the order | | |
| **Trigger** | The employee presses “Cancel” button | | |
| **Preconditions** | PRE-1: The employee is logged in  PRE-2: The employee selected an order | | |
| **Postconditions** | POST-1: The order status is set from “In progress” to “Canceled” | | |
| **Normal flow** | 1. Cancel order: 2. The employee presses “Cancel” button on his current working order 3. The application displays a screen asking the employee to give the reason for canceling the order 4. The employee describe the reason for canceling the order 5. The application displays a window for the client with “Order Canceled” title and the reason why his order was canceled 6. The application sets order status from “In progess” to “Canceled” | | |
| **Alternative flows** |  | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-11:Finish order | | |
| **Primary actor** | Employee | **Secondary actors** |  |
| **Description** | After the employee has finished working on the order he wants to send to the client his order | | |
| **Trigger** | The employee presses “Finish” button | | |
| **Preconditions** | PRE-1: The employee is logged in  PRE-2: The employee selected an order  PRE-3: The employee has finished the order | | |
| **Postconditions** |  | | |
| **Normal flow** | 1. Finish order: 2. The employee presses “Finish” button on his current order 3. The application closes the current working order screen for employee 4. The application displays the finalized custom build to the client | | |
| **Alternative flows** |  | | |
| **Exceptions** | 1. E1 The order costs more than the client`s budget: 2. If the order costs more then the client`s budget , the employee can`t finish the order and has to modify the order to client`s budget | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-12: Log Out | | |
| **Primary actor** | Employee | **Secondary actors** |  |
| **Description** | After the employee has finished his work schedule, he wants to disconnect from the workplace | | |
| **Trigger** | The employee presses “Log Out” button | | |
| **Preconditions** | PRE-1: The employee is logged in | | |
| **Postconditions** | POST-1: The employee is logged out | | |
| **Normal flow** | **1.0Log out**   1. **The employee presses “Log Out” button from the main screen** 2. **The application closes for the current employee** | | |
| **Alternative flows** | * 1. Still working on an order:  1. If the employee is still working on an order he has to finish, cancel, or put the order to waiting in order to Log Out 2. Back to normal flow 1 | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-13: Log Out | | |
| **Primary actor** | Client | **Secondary actors** |  |
| **Description** | After the client enjoyed the application's facilities, he decided that it was time to do something else, so he wanted to disconnect | | |
| **Trigger** | The client presses “Log Out” button | | |
| **Preconditions** | PRE-1: The client is logged in | | |
| **Postconditions** | POST-1: The client is logged out | | |
| **Normal flow** | **1.0Log out**  **1. The client presses “Log Out” button**  **2. The application closes for the current client** | | |
| **Alternative flows** |  | | |
| **Exceptions** |  | | |