# **Assessment Use Case Diagram**

# 

## **Initial Use Case Texts**

Creating a new agent:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the user selects to the option to create a new agent. | - |
| 2 | The user input the new agent’s details (username, password, name, address, phone) | - |
| 3 | - | System reads a list of agents and it checks the username isn’t already used and all the fields are valid. |
| 4 | - | System allocates the new agent with an Agent ID |
| 5 | - | System adds agent to the system. |
| 6 | - | System displays a confirmation. |

Modifying agent details:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the user selects to the option to modify an agent’s details. | - |
| 2 | - | The system loads details into a choice box. |
| 3 | User selects an agent to modify. | - |
| 4 | - | System loads the selected agents details into the form. |
| 5 | The user makes their desired changes to the details. | - |
| 6 | - | System checks the validity of the proposed changes. |
| 7 | - | System saves the changes. |
| 8 | - | Confirmation message is displayed. |

Make ticket sale:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the user selects the option to sell a ticket | - |
| 2 | - | The system loads a list of events into a choice box. |
| 3 | The user selects the event they would like to sell a ticket for and a quantity of tickets. | - |
| 4 | - | The system reads a list of agents and events. The system checks the valid sale date for the event to make sure it can be sold. |
| 5 | - | The system checks the quantity of the event and if there are enough tickets available. |
| 6 | Agent enters quantity of tickets they would like to sell. | The system adds the amount of sales to the agent. |
| 7 | - | The system saves the details for the event and agent. |
| 8 | - | System displays confirmation of ticket sale. |

Cancel ticket sale:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the user selects the option to cancel a ticket sale | - |
| 2 | - | The system loads the details for agents and events into 2 choice boxes. |
| 3 | The user chooses an event, a quantity of tickets to cancel and an agent to remove the sales from. | - |
| 4 | - | The system removes the tickets from the event. |
| 5 | - | The system removes the tickets from the agent. |
| 6 | - | The system saves the lists of agents and events. |
| 7 | - | System displays confirmation of ticket cancellation. |

View ticket sale activity:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the user selects the option to see clients list (ticket sale activity) | - |
| 2 | - | The system loads a list of agent and event details. |
| 3 | - | The system retrieves event details. |
| 4 | - | The system retrieves agent details. |
| 5 | - | The system displays the sale details in a table. |

## **Design Level Use Case Texts**

Creating a new agent:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the user selects to the option to create a new agent. | - |
| 2 | The user enters username, password, name, address and phone into the fields. | - |
| 3 |  | The JJT Class reads a list of agents from a file. Validity of the entered details are checked e.g. Username not taken in list, no fields blank, phone has 11 digits. |
| 4 | - | The system counts the amount of agent objects in the list and adds 1 to this amount to be used as the new agents ID. |
| 5 | - | The system creates a new Agent object using the ID created in step 4. This is agent is saved to the list of agents stored in the JJT Class. |
| 6 | - | The system then displays confirmation to the user that an agent has been created. |

Modifying agent details:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the user selects to the option to modify an agent’s details. | - |
| 2 | - | The JJT class reads the list of agents and the system uses this list to populate a choice box. |
| 3 | The user selects the agent they want to modify. | - |
| 4 | - | The system uses the agent list to get details of the agent and displays them in the fields for the user. |
| 5 | The user makes their desired changes to the attributes. | - |
| 6 |  | The system uses a validation class to check that the users amendments are all valid. E.g. Unique username, no blank fields. |
| 7 | - | The system sets the new attributes of the agent object. |
| 8 | - | The edited agent object is saved to the list from the JJT class. |
| 9 | - | The system displays confirmation to the user the agent has been modified. |

Make ticket sale:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the Agent selects the option to sell a ticket | - |
| 2 | - | The system populates the choice box with name of event. |
| 3 | The agent chooses an event from the choice box and uses a slider to set the quantity. | - |
| 4 | - | The system reads in list of events agents and events using the JJT class. |
| 5 | - | The date is retrieved for the event object chosen by the user and it is checked in a function to make sure the ticket is within a valid sale date. |
| 6 | - | The quantity for the event object is retrieved and is checked in a function to make sure there are enough tickets to make the sale. |
| 7 | - | The system gets the old sales from the agent object and adds the chosen quantity to the amount and sets the new value to them. |
| 8 | - | The System then saves the list of agents of events retrieved from the JJT class. |
| 9 | - | Confirmation of the ticket sales is then displayed. |

Cancel ticket sale:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the Agent selects the option to cancel a ticket sale | - |
| 2 | - | The system populates choice boxes with agent and event objects. |
| 3 | The user chooses an event to cancel tickets for, an amount to reverse and an agent to remove the sales from. | - |
| 4 | - | The system reads a list of agent and event objects using the JJT class. |
| 5 | - | The system gets the quantity of tickets left for the event object from the event class. |
| 6 | - | The system adds the quantity chosen by the user to the quantity of tickets left and sets the quantity using a setQuantity function. |
| 7 | - | The system then retrieves the sales from the agent object and removes the quantity before setting the new quantity. |
| 8 | - | The agent and event lists are saved. |
| 9 | - | Confirmation of the ticket cancellation is then displayed. |

View ticket sale activity:

|  |  |  |
| --- | --- | --- |
| Step | Actor Action | System Response |
| 1 | The use case begins when the Agent selects the option to see the client List (sales activity) | - |
| 2 | - | The system reads a list of agent and event objects using a function in the JJT class. |
| 3 | - | The system uses functions to get the quantity of tickets left, the location and the valid sale dates for the event object. |
| 4 | - | The system gets the sales from all the agent object from the agent list. |
| 5 | - | All the values are parsed into Strings and are assigned to labels in the form of a table. |