|  |  |
| --- | --- |
| **ID:** ATS1 | **Use Case:** TicketSales |
| **Brief Description:** Ticket sales are sold through the TravelAdvisor or the OfficeManager and are sold to the customers. | |
| **Primary Actors:** TravelAdvisor,OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:**   1. System is operational. 2. Check ticket price should be performed by the actor. | |
| **Main Flow:**   1. The use case “TicketSales” starts when the actor is beginning to sell the flight tickets to the customers. 2. Actor selects the desired performance from the system. 3. The customer looks up the information related to various airlines and checks the availability of seats on flights.    1. The system provides a list of items that matches the search keywords 4. The actor checks that there are any available tickets.   4.1 The system provides more information about the selected item.   1. Actor enters ticket selling date and the ticket price for the desired performance. 2. If tickets are available the actor chooses to add item to cart    1. The system adds item to the card. 3. Actor enters the details about the customer who wants to buy the ticket.    1. The system checks for the validity of all required data entry.   **Include (TakePayment)**   1. The system connects to the database. 2. The system writes data into the database. 3. The system shows a confirmation message after successful ticket sale. | |
| **Postconditions:**   1. Every ticket sold is registered in an Air Ticket Sales report. 2. The application verifies the authenticity of the username and password and then displays information related to various flights to the customer. 3. A new sale is recorded by the actor. 4. The appropriate commission amount is recorded. | |
| **Alternative Flow:**  NoSeatsLeft  BlanksFinished  DatabaseConnectionProblem  DatabaseWriteProblem  InvalidCustomerData | |

|  |
| --- |
| **Alternative Flow:** BlanksFinished |
| **ID:** ATS1.1 |
| **Brief Description:** There are no blanks left for the TravelAdvisor who want to sell tickets. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** Actor attempted to sell a ticket but no blanks left to sell. |
| **Alternative Flow:**   1. Use case begins when the system informs the TravelAdvisor that there is no blank left to sell. 2. TravelAdvisor informs the OfficeManager that he/she needs to assign new blanks to the TravelAdvisor. 3. OfficeManager activates the functionality to assign new blanks. 4. OfficeManager assigns the blanks to the TravelAdvisor. 5. TravelAdvisor sells to ticket for the customer who wants to buy the ticket. |
| **Postconditions:** New blanks are assigned to the customer. |

|  |
| --- |
| **Alternative Flow:** NoSeatsLeft |
| **ID:** ATS1.2 |
| **Brief Description:** There is no empty seat left for the desired flight. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of selling a ticket. |
| **Alternative Flow:**   1. Use case begins when the system checks for availability of seats and informs the actor that there are no seats left. 2. Actor informs the customer. 3. Customer chooses alternative flight. 4. Actor processes to the next stage of selling the ticket. |
| **Postconditions:** Customer informed that there is no seat left for the specific flight and availability of other flights. |

|  |
| --- |
| **Alternative Flow:** DatabaseConnectionProblem |
| **ID:** ATS1.3 |
| **Brief Description:** There has been a problem while connecting to the database. |
| **Primary Actors:** TravelAdvisor, OfficeManager, SystemAdministrator |
| **Secondary Actors:** None. |
| **Preconditions:** TravelAdvisor or the OfficeManager tried to connect to the database but a problem occurred and failed. |
| **Alternative Flow:**   1. Use case begins when the TravelAdvisor or the OfficeManager fails to connect to the database. 2. TravelAdvisor or the OfficeManager informs the SystemAdministrator that the database needs to be fixed.   **Extension Point: RepairDatabase**   1. SystemAdministrator repairs the database. |
| **Postconditions:** Database is repaired. |

|  |
| --- |
| **Alternative Flow:** DatabaseWriteProblem |
| **ID:** ATS1.4 |
| **Brief Description:** There has been a problem while writing data into the database |
| **Primary Actors:** TravelAdvisor, OfficeManager, SystemAdministrator |
| **Secondary Actors:** None. |
| **Preconditions:** TravelAdvisor or the OfficeManager tried to write data into the database but a problem occurred. |
| **Alternative Flow:**   1. Use case begins when the TravelAdvisor or the OfficeManager fails to write data into the database. 2. TravelAdvisor or the OfficeManager informs the SystemAdministrator that the database needs to be fixed.   **Extension Point: RepairDatabase**   1. SystemAdministrator repairs the database. |
| **Postconditions:** Database is repaired. |

|  |
| --- |
| **Alternative Flow:** InvalidCustomerData |
| **ID:** ATS1.5 |
| **Brief Description:** TravelAdvisor or the OfficeManager enters the data about the customer but the system informs that some of the data provided is invalid or missing. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:**   1. There must be an attempt of selling a ticket. 2. Customer does not provide all the required data entry or the actor does not enter the information correctly. |
| **Alternative Flow:**   1. Use case begins when the customer wants to buy the ticket and provides information. 2. Actor enters the details about the customer. 3. System checks for the data provided. 4. System informs the actor that there has a problem with the data provided. It might be missing or incorrect. 5. Actor checks the data and if necessary, asks for additional data. 6. System checks the data again. 7. System confirms the validity. 8. Actor processes the next stage of selling the ticket. |
| **Postconditions:** The valid data about the customer is provided. |

|  |  |
| --- | --- |
| **ID:** ATS2 | **Use Case:** CancelTicket |
| **Brief Description:** This use case describes when the TravelAdvisor or the OfficeManager cancels the customers' tickets**.** | |
| **Primary Actors:** TravelAdvisor, OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:**   1. System is operational. 2. The customer who wants to cancel the ticket must have bought it through the ATS. 3. Actor who performs the action should be logged in. | |
| **Main Flow:**   1. The use case begins when the customer decides the cancel their ticket. 2. The actor provides the customer details.   2.1 The system looks up for the customer details.   1. IF the ticket was purchased within 1 year and it has not been used    1. The system accepts the cancellation.   ELSE   * 1. The system does not approve the cancellation.   **Include (RefundTicket)**   1. The system checks for all required data entry. 2. The system connects to the database. 3. The system writes data to the database. 4. The system shows confirmation message after successful cancellation. | |
| **Postconditions:**   1. A recorded is made detailing the ticket returned. 2. Details are recorded about the amount that is getting refunded to a customer. | |
| **Alternative Flow:**  OverOneYear  InvalidCustomerData | |

|  |
| --- |
| **Alternative Flow:** OverOneYear |
| **ID:** ATS2.1 |
| **Brief Description:** Customer wants to cancel the ticket but the ticket purchased over a year ago. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** Time. |
| **Preconditions:** There has been an attempt of cancelling a ticket. |
| **Alternative Flow:**   1. System checks the details of the ticket provided by the actor. 2. System will display that there is an error with the cancellation “Your cancellation has not been successful. Ticket purchased over a year ago. ” |
| **Postconditions:** Ticket will not be cancelled and the customer should be informed. |

|  |
| --- |
| **Alternative Flow:** InvalidCustomerData |
| **ID:** ATS2.2 |
| **Brief Description:** The information about the customer is invalid. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There is an attempt of cancelling the ticket. |
| **Alternative Flow:**   1. Actor provides the information about the customer who wants to cancel their ticket. 2. System checks for the data entry. 3. System informs the actor the data is invalid or missing. 4. Actor provides additional data or corrects the data provided before. 5. System checks the data and validates it. 6. Actor processes the next stage of cancelling the ticket. |
| **Postconditions:** The customer data is validated. |

|  |  |
| --- | --- |
| **ID:** ATS3 | **Use Case:** TakePayment |
| **Brief Description:** This use case describes when the TravelAdvisor or the OfficeManager takes the payment for the wanted flight tickets for the customer. | |
| **Primary Actors:** TravelAdvisor, OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:**   1. System is operational. 2. Actor is logged in. 3. Customers have decided what ticket they want to buy. | |
| **Main Flow:**   1. The use case starts when the actor processes to checkout    1. Actor activates the functionality to take the payment. 2. System calculates the total amount including taxes. 3. System asks for the customer account details.   **Extension Point: CreateCustomerAccount**   * 1. IF the customer is valued, actor gives discount   2. ELSE proceed to checkout with no discount when the customer is regular  1. System asks for the payment details. 2. IF the customer wants to buy by card,    1. Actor asks for the card details.   ELSE   * 1. Actor takes cash payment.   Extension Point: ConvertCurrency   1. The system records the payment details.    1. IF card payment: records account number, name on the card, card number, date, time, amount combined with the customer details.    2. ELSE cash payment: records amount, date and time combined with the customer details. 2. Payment must proceed and confirmation must be provided. 3. Ticket is registered to the Air Ticket Sales Report by the actor. 4. System checks for all the data entry. 5. System connects to the database. 6. The system writes data into database. 7. The system shows the confirmation message after successful payment. | |
| **Postconditions:**   1. A new sale recorded in the database by the actor. 2. System received whole amount at once. 3. Details about the payment is recorded. | |
| **Alternative Flow:**  WrongAmountReceived  InvalidCustomerData  IssueWithCardDetails | |

|  |
| --- |
| **Alternative Flow:** InvalidCustomerData |
| **ID:** ATS3.1 |
| **Brief Description:** The details about the customer is invalid. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There has been an attempt of taking the payment but the information about the customer is invalid or missing. |
| **Alternative Flow:**   1. Use case begins when the customer provides information. 2. Actor enters the details about the customer. 3. System checks for the data provided. 4. System informs the actor that there has a problem with the data provided. It might be missing or incorrect. 5. Actor checks the data and if necessary, asks for additional data. 6. System checks the data again. 7. System confirms the validity. 8. Actor processes the next stage of payment. |
| **Postconditions:** System approved the data provided. |

|  |
| --- |
| **Alternative Flow:** WrongAmountReceived |
| **ID:** ATS3.2 |
| **Brief Description:** The amount received from the customer at the checkout is less than expected. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of taking the payment. |
| **Alternative Flow:**   1. This use case begins when the system asks for the payment details. 2. Customer provides the amount of money which is required by the actor. 3. Actor enters the amount to the system. 4. System checks and informs the actor that the amount provided is incorrect. 5. Actor informs the customer. 6. Customer provides the correct amount of money. 7. System checks and validates. |
| **Postconditions:** Correct amount of money is received. |

|  |
| --- |
| **Alternative Flow:** IssueWithCardDetails |
| **ID:** ATS3.3 |
| **Brief Description:** The details about the card which is being used for the payment cannot be accepted. The card might be expired etc. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of taking the payment. |
| **Alternative Flow:**   1. Actor provides the card details to the system. 2. System checks the details and informs the user that the card cannot be accepted. 3. Actor informs back to the customer. |
| **Postconditions:** No sales are made with the card which is not accepted by the system. |

|  |  |
| --- | --- |
| **ID:** ATS4 | **Use Case:** AccessSalesReports |
| **Brief Description:** This is the use case where the sales reports are accessed by the OfficeManager. | |
| **Primary Actors:** OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:**   1. System is operational. 2. OfficeManager should be logged in. 3. Tickets sold by the advisor must be registered. 4. Sales reports must be created before it can be accessed. | |
| **Main Flow:**   1. The use case starts when the OfficeManager activates the functionality. 2. The system will ask whether they want to access interline or domestic sales report. 3. The actor selects the report he/she wants to access. 4. System displays the desired report.   **Include (View)** | |
| **Postconditions:** Reports are accessed by the airline. | |
| **Alternative Flow:**  SalesReportNotCreated | |

|  |
| --- |
| **Alternative Flow:** SalesReportNotCreated |
| **ID:** ATS4.1 |
| **Brief Description:** No sales report is created. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** After sales made, there has been no attempt of creating a sales report by the OfficeManager. |
| **Alternative Flow:**   1. Either TravelAdvisor or the OfficeManager tries to access the sales report. 2. System informs the actor that there is no sales report created. 3. TravelAdvisor informs the OfficeManager. 4. OfficeManager creates the sales report.   **Include (CreateSalesReport)** |
| **Postconditions:** Sales report is created. |

|  |  |
| --- | --- |
| **ID:** ATS5 | **Use Case:** CreateSalesReport |
| **Brief Description:** The OfficeManager creates sales report which contains all the sales transactions corresponding sub-totals and grand totals. | |
| **Primary Actors:** OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:**   1. OfficeManager has logged in to the system. 2. System is operational. | |
| **Main Flow:**   1. Use case begins when the OfficeManager activates the functionality to generate sales report. 2. System calculates sub-totals and grand totals and displays Individual Sales Report. 3. System calculates sub-totals and grand totals and displays Global Sales Report. 4. System combines Individual and Global Reports in a file. 5. OfficeManager selects save report. 6. System checks for all the data entry. 7. System connects to the database. 8. The system writes data into database. 9. The system shows the confirmation message after successful creation of a report. | |
| **Postconditions:** The sales report is generated. | |
| **Alternative Flow:**  DatabaseConnectionProblem  DatabaseWriteProblem | |

|  |
| --- |
| **Alternative Flow:** DatabaseConnectionProblem |
| **ID:** ATS5.1 |
| **Brief Description:** There has been a problem while connecting to the database. |
| **Primary Actors:** OfficeManager, SystemAdministrator |
| **Secondary Actors:** None. |
| **Preconditions:** OfficeManager tried to connect to the database but a problem occurred and failed. |
| **Alternative Flow:**   1. Use case begins when the OfficeManager fails to connect to the database. 2. OfficeManager informs the SystemAdministrator that the database needs to be fixed.   **Extension Point: RepairDatabase**   1. SystemAdministrator repairs the database. |
| **Postconditions:** Database is repaired. |

|  |
| --- |
| **Alternative Flow:** DatabaseWriteProblem |
| **ID:** ATS5.2 |
| **Brief Description:** There has been a problem while writing data into the database |
| **Primary Actors:** OfficeManager, SystemAdministrator |
| **Secondary Actors:** None. |
| **Preconditions:** OfficeManager tried to write data into the database but a problem occurred. |
| **Alternative Flow:**   1. Use case begins when the OfficeManager fails to write data into the database. 2. OfficeManager informs the SystemAdministrator that the database needs to be fixed.   **Extension Point: RepairDatabase**   1. SystemAdministrator repairs the database. |
| **Postconditions:** Database is repaired. |

|  |  |
| --- | --- |
| **ID:** ATS6 | **Use Case:** AssignBlanks |
| **Brief Description:** Blanks are received by the airline and added to the stock of blanks. OfficeManager allocates the blanks to each TravelAdvisor before they can sell them to a customer. Unless a blank is assigned it cannot be sold. | |
| **Primary Actors:** OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:** The system is operational and blanks are received from the airline and added to the stock of blanks with a unique blank ID. | |
| **Main Flow:**   1. Actor activated the functionality for assigning a blank.   **Extension Point: ReassignBlanks**   1. OfficeManager selects a specific TravelAdvisor to assigns the blanks. 2. The system validates that each blank is assigned to a unique TravelAdvisor. 3. OfficeManager selects save. 4. System checks for all the data entry. 5. System connects to the database. 6. The system writes data into database. 7. The system shows the confirmation message after successful assign process. | |
| **Postconditions:** A blank is assigned to a TravelAdvisor. | |
| **Alternative Flow:**  StockIsEmpty | |

|  |  |
| --- | --- |
| **Alternative Flow:** StockIsEmpty | |
| **ID:** ATS6.1 | |
| **Brief Description:** The stock of blanks is empty therefore there is no blank to assign. | |
| **Primary Actors:** OfficeManager / SystemAdministrator | |
| **Secondary Actors:** None. | |
| **Preconditions:** OfficeManager must be logged in. | |
| **Alternative Flow:**   1. The system prompts the actor that the stock is empty. 2. Either OfficeManager or the SystemAdministrator adds new blanks to the stock with a unique ID. 3. System saves the changes. | |
| **Postconditions:** Blanks are added to the stock. | |
| **ID:** ATS7 | **Use Case:** RepairDatabase |
| **Brief Description:** The database fails to perform some actions such as storing new data, deleting data or accessing the database. It needs to be repaired by the SystemAdministrator. | |
| **Primary Actors:** SystemAdministrator | |
| **Secondary Actors:** None. | |
| **Preconditions:** The system is operational. | |
| **Main Flow:**   1. Users of the ATS informs the SystemAdministrator that there is an issue with the database and it needs to be repaired. 2. Actor backs up the database. 3. Actor gets exclusive access to the database. 4. Actor analysis the database and finds the problem. 5. Actor informs other ATS users how long they must avoid using the database. 6. Actor repairs the database. 7. Actor saves the changes. | |
| **Postconditions:** Database is repaired. | |
| **Alternative Flow:** None. | |

|  |  |
| --- | --- |
| **ID:** ATS8 | **Use Case:** ChangeCommissionRate |
| **Brief Description:** OfficeManager can change the commission rate after he/she agreed with the AirVia. | |
| **Primary Actors:** OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:**   1. System is operational. 2. OfficeManager has logged in. 3. OfficeManager and AirVia agrees with the commission rate. | |
| **Main Flow:**   1. OfficeManager selects the functionality to change the commission rate. 2. OfficeManager changes the commission rate. 3. OfficeManager saves the changes. 4. System checks for all the data entry. 5. System connects to the database. 6. The system writes data into database. 7. The system shows the confirmation message after successful assign process. | |
| **Postconditions:** Commission rate has been changed. | |
| **Alternative Flow:**  InvalidDataEntry | |

|  |
| --- |
| **Alternative Flow:** InvalidDataEntry |
| **ID:** ATS8.1 |
| **Brief Description:** The data provided as the commission rate is invalid. |
| **Primary Actors:** OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of changing the commission rate. |
| **Alternative Flow:**   1. OfficeManager provides specifies the rate and selects ‘Save Commission Rate’ 2. System checks the validity of the data provided i.e. if the rate is sensible.   IF the rate is sensible, system saves the changes.  ELSE system shows an error ‘Invalid Data’ |
| **Postconditions:** Commission rate in validated. |

|  |  |
| --- | --- |
| **ID:** ATS9 | **Use Case:** CreateUserAccount |
| **Brief Description:** ATS provides functionality to crate accounts for its users. Accounts are created by the OfficeManager. | |
| **Primary Actors:** OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:** No user exist in ATS for the new user. | |
| **Main Flow:**   1. Actor activates the functionality for creating an account. 2. The system places an empty user account in the session. 3. The system opens a form for entering the user details such as username, email address and password. 4. The actor enters details.    1. System confirms their validity 5. System requires to choose the role of the newly created user.   5.1 OfficeManager specifies the role.   1. The system creates a new user record in ATS. | |
| **Postconditions:** User details are stored in the database. | |
| **Alternative Flow:**  InvalidUserInformation  UnspecifiedRole | |

|  |
| --- |
| **Alternative Flow:** InvalidUserInformation |
| **ID:** ATS9.1 |
| **Brief Description:** The information provided for the new user is invalid or missing. |
| **Primary Actors:** OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of creating a new user account. |
| **Alternative Flow:**   1. Use case starts when the actor provides the information to the system. 2. System checks for the validity of the information provided.   IF required data is entered, system creates the user.  ELSE system asks for additional information or correction |
| **Postconditions:** User information is validated. |

|  |
| --- |
| **Alternative Flow:** UnspecifiedRole |
| **ID:** ATS9.2 |
| **Brief Description:** OfficeManager have not provided the information for the role of the new user. Unless the role is specified, new user account cannot be created. |
| **Primary Actors:** OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of creating a new ATS user. |
| **Alternative Flow:**   1. System asks for the role of the new user. 2. OfficeManager specifies the role. 3. System gives privileges to the user according to the type of the role. 4. System saves the changes. |
| **Postconditions:** Required information is provided. |

|  |  |
| --- | --- |
| **ID:** ATS10 | **Use Case:** RefundTicket |
| **Brief Description:** This use case describes how the customer get their money back when they cancel their ticket. | |
| **Primary Actors:** TravelAdvisor, OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:** There must be an attempt of cancelling the ticket. | |
| **Main Flow:**   1. Use case starts when the system accepts the cancellation. 2. System display the details of the ticket purchased including amount.   IF it was a card payment, system requires for the card used to pay for the ticket.  ELSE actor gives the cash to the user.   1. After the transection,   IF it was a card payment, system displays ‘Successful’.  ELSE actor selects ‘Done’, and system displays ‘Successful’.   1. Actor keeps record of the ticket which has been returned. | |
| **Postconditions:**   1. Customer gets their money. 2. Log file of the actor is updated. | |
| **Alternative Flow:** None. | |

|  |  |
| --- | --- |
| **ID:** ATS11 | **Use Case:** CreateCustomerAccount |
| **Brief Description:** ATS provides functionality create accounts for its customer to keep information about them. | |
| **Primary Actors:** TravelAdvisor, OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:**   1. System is operational. 2. Actor is logged in to the system. | |
| **Main Flow:**   1. Use case starts when the actor activates the functionality to create a user account. 2. System asks for the customer information: name, surname, email, phone number etc. 3. Actor enters the information provided by the customer. 4. System checks for the validity of the information. 5. System connects to the database. 6. The system writes data into database. 7. The system shows the confirmation message after successful account creation. | |
| **Postconditions:** A user account created with a unique email. | |
| **Alternative Flow:** InvalidCustomerData | |

|  |
| --- |
| **Alternative Flow:** InvalidCustomerData |
| **ID:** ATS11.1 |
| **Brief Description:** Use case describes what happens when the data provided by the actor is incorrect or missing. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of creating a user account. |
| **Alternative Flow:**   1. Use case begins when the actor enters data to the system. 2. Actor enters the details about the customer. 3. System checks for the data provided. 4. System informs the actor that there has a problem with the data provided. It might be missing or incorrect. 5. Actor checks the data and if necessary, asks for additional data. 6. System checks the data again. 7. System confirms the validity. 8. Actor processes the next stage of creating a user account. |
| **Postconditions:** The new user details are provided as required by the system. |

|  |  |
| --- | --- |
| **ID:** ATS12 | **Use Case:** ConvertCurrency |
| **Brief Description:** Use case describes when the customer wants to pay in US Dollars instead of local currency. | |
| **Primary Actors:** TravelAdvisor, OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:** There must be an attempt of taking money by the TravelAdvisor or the OfficeManager. | |
| **Main Flow:**   1. Actor asks for the money from the customer who wants to buy tickets. 2. Customer wants to pay in US Dollars instead of local currency.   IF paying by card; system automatically converts the money when the card details are provided.  ELSE  Actor selects ‘Payment in USD’ and system converts the local amount into USD.   1. Actor tells the amount in USD to the customer. 2. Customer pays the whole amount. 3. Actor puts the money in the till and gives change in USD if necessary. | |
| **Postconditions:** Money is converted. | |
| **Alternative Flow:**  OtherTypeOfCurrency | |

|  |
| --- |
| **Alternative Flow:** OtherTypeOfCurrency |
| **ID:** ATS12.1 |
| **Brief Description:** Customer may want to pay in other types of currency. ATS does not accept any other currency apart from USD. |
| **Primary Actors:** TravelAdvisor, OfficeManager |
| **Secondary Actors:** None. |
| **Preconditions:** There must be an attempt of making payment. |
| **Alternative Flow:**   1. Customer wants to pay in other type of currency. 2. Actor informs the customer that they do not accept other currencies. 3. Customer chooses either local currency or US Dollars to pay in. 4. Actor processes the next stage of converting the money or taking payment. |
| **Postconditions:** There has not been any payment in currency other than the US Dollars. |

|  |  |
| --- | --- |
| **ID:** ATS13 | **Use Case:** View |
| **Brief Description:** ATS provides functionality to view and print the reports. | |
| **Primary Actors:** OfficeManager | |
| **Secondary Actors:** PDFReader | |
| **Preconditions:** OfficeManager has logged in to the system. | |
| **Main Flow:**   1. Use case starts when the OfficeManager accesses the reports. 2. OfficeManager selects a specific report to be viewed. 3. System sends files to the PDFReader. 4. PDFReader opens the selected file.   IF actor wants to print the files, he/she can select print option provided by the PDFReader. | |
| **Postconditions:** Files are viewed. | |
| **Alternative Flow:** None. | |

|  |  |
| --- | --- |
| **ID:** ATS14 | **Use Case:** ReassignBlanks |
| **Brief Description:** A blanks initially assigned to an TravelAdvisor can be reassign to a different TravelAdvisor by the OfficeManager. | |
| **Primary Actors:** OfficeManager | |
| **Secondary Actors:** None. | |
| **Preconditions:** OfficeManager. | |
| **Main Flow:**   1. Actor activates the functionality to assign a blank. 2. OfficeManager selects a specific TravelAdvisor to assigns their blanks to another TravelAdvisor. 3. OfficeManager reassigns the new blanks to the other TravelAdvisor. 4. The system validates that each blank is assigned to a unique TravelAdvisor. 5. OfficeManager selects save. 6. System checks for all the data entry. 7. System connects to the database. 8. The system writes data into database. 9. The system shows the confirmation message after successful assign process | |
| **Postconditions:** Blanks are reassigned to a different TravelAdvisor. | |
| **Alternative Flow:** None. | |