|  |  |
| --- | --- |
| **Test Case ID:** 1 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/high):** high | **Test Designed date:** 21/12/2020 |
| **Module Name:** login\_as\_agent | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  login as agent | **Test Execution date** 21/12/2020 |
| **Description:** The function return true if username and password  is exist on agents.txt. | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** username and password | |
| **Dependencies:** on agents.txt | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | login\_as\_agent | Username: harel  Password:123 | Expect to return true | get true | Pass |  |
| 2 | login\_as\_agent | Username: harel1  Password:123 | Expect to return false | Get false | pass |  |

|  |  |
| --- | --- |
| **Test Case Template** | |
| **Test Case ID:** 2 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** high | **Test Designed date:** 21/12/2020 |
| **Module Name:** package\_management | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:** See all the options available in the function | **Test Execution date** :21/12/2020 |
| **Description:** -------------------------------------------- | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** Agent has valid username and password | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2 | package\_management | Username: harel  Password:123 | Be able to see:  [1] add new package  [2] edit package  [3] remove package | Agent see :  [1] add new package  [2] edit package  [3] remove package | Pass |  |

|  |  |
| --- | --- |
|  | |
| **Test Case ID:** 3 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** high | **Test Designed date:** 21/12/2020 |
| **Module Name:** add\_package\_to\_db | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:** add new package to db | **Test Execution date** 21/12/2020 |
| **Description:** agent be able to set a new package to db | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** Agent has valid username and password | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.1 | add\_package\_to\_db | start date:24/12/2020  end date:28/12/2020  destination:isreal  price:50  number\_of\_passengers:2 | Adding a package to the database | The package was added to the database | Pass |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Test Case ID:** 4 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi | | **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 | | **Module Name:** remove\_package\_to\_db | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi | | **Test Title:** remove exist package from db | **Test Execution date** 21/12/2020 | | **Description:** agent be able to set a new package to db | **Requirement ID** input and output ///// ????? | |  |  | |  |  | | **Pre-conditions:** Agent has valid username and password | | | **Dependencies:** | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** | | 2.1 | remove\_package\_to\_db | Serial number=74523 | removing a package to the database | The package was removed to the database | Pass |  |  |  | | --- | |  | | |
| **Test Case ID:** 5 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** get\_start\_date\_of\_package | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:** return date | **Test Execution date** 21/12/2020 |
| **Description:** The function asks a day, month and year -> and returns a date struct | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** Agent has valid username and password | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  Date:24/12/2020 | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.1.1 | get\_start\_date\_of\_package |  | Return struct with the inserted date | The function Absorbed a day, month, year and returned an object with the inserted date | Pass |  |

|  |  |
| --- | --- |
|  | |
| **Test Case ID:** 6 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** get\_end\_date\_of\_package | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:** return date | **Test Execution date** 21/12/2020 |
| **Description:** The function asks a day, month and year -> and returns a date struct | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** Agent has valid username and password | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.1.2 | get\_end\_date\_of\_  package | Date:30/12/2020 | Return struct with the inserted date | The function Absorbed a day, month, year and returned an object with the inserted date | Pass |  |

|  |  |
| --- | --- |
|  | |
| **Test Case ID:** 7 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** get\_destination | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:** return destination string | **Test Execution date** 21/12/2020 |
| **Description:** The function asks destination and  return destination | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** Agent has valid username and password | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  Condition: Up to 100 characters in the destination | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.1.3 | get\_destination | string –destination:poland | The function asks destination and  return destination | The function  return destination | pass |  |

|  |  |
| --- | --- |
|  | |
| **Test Case ID:** 8 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** get\_price | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:** return price | **Test Execution date** 21/12/2020 |
| **Description:** The function asks price and  return | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  Condition:price should be highter then 0 | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.1.4 | get\_price | Price=0 | Receive an output message that the number cannot be equal to 0 and request a new price.  Send an output message that the number cannot be equal to 0 and ask for a new price | An output message was sent that the number could not be equal to 0 and a correct number was entered and the same number was returned | pass |  |

|  |  |
| --- | --- |
| **Test Case ID:** 9 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** get\_number\_of\_passengers | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:** return number of passengers | **Test Execution date** 21/12/2020 |
| **Description:** The function asks number of passengers | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  Condition:number of passengers should be highter then 0  Number of passengers=0 | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.1.5 | get\_number\_of\_passengers |  | Receive an output message that the number cannot be equal to 0 and request a new number passengers  Send an output message that the number cannot be equal to 0 and ask for a new number passengers | An output message was sent that the number could not be equal to 0 and a correct number was entered and the same number was returned | pass |  |

|  |  |
| --- | --- |
|  | |
| **Test Case ID:** 10 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** high | **Test Designed date:** 21/12/2020 |
| **Module Name:** Reply\_to\_messages\_by\_an\_agent | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  The agent responds to the message | **Test Execution date** 21/12/2020 |
| **Description:** The function asks text up to 300 characters | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  Condition: text up to 300 characters    Massage="its test" | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 3 | Reply\_to\_messages\_by\_an\_agent |  | The function returns the answer to the customer, and prints a message if the sender is a success / failure | An output message was sent to the customer.  And the "Successfully sent" message was printed | pass |  |

|  |  |
| --- | --- |
|  | |
| **Test Case ID:** 11 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** high | **Test Designed date:** 21/12/2020 |
| **Module Name:** edit\_package | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  update the package price | **Test Execution date** 21/12/2020 |
| **Description:** The function asks for a new price and package number | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  asks for a new price and package number | **Expected Result**  The function will print to the agent the following5 things: | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.2 | edit\_package |  | 1. edit by price 2. edit by arrive date 3. edit by leave date 4. edit amount for package 5. return to agent menu | What is happening is exactly as we expected | pass |  |

|  |  |
| --- | --- |
|  | |
| **Test Case ID:** 12 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** edit\_by\_price | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  Verify edit price of package | **Test Execution date** 21/12/2020 |
| **Description:** The function updates the price of the package | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  Serial number=78589  Price=200$ | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 2.2.1 | edit\_by\_price |  | We expect to receive a serial number of the package and update the price according to the price brought by the agent | The price of the package updated | pass |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | | | **Test Case ID:** 13 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi | | **Test Priority (Low/Medium/High):** high | **Test Designed date:** 21/12/2020 | | **Module Name:** get\_serial\_number | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi | | **Test Title:**  return serial package | **Test Execution date** 21/12/2020 | | **Description:** The function return serial package selected | **Requirement ID** input and output | |  |  | |  |  | | **Pre-conditions:** serial number | | | **Dependencies:** if seial number exist on db packages return true | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** | | 1 | get\_serial\_number | Serial number=78589 | Expect to return true number of package: | get true | Pass |  | | 2 | get\_serial\_number | Serial number=1234(isn’t exist on db packages) | Expect to return false number of package: | get false | Pass |  | | |
| **Test Case ID:** 14 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** edit\_by\_arrive\_date | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  Verify edit by arrive date of package | **Test Execution date** 21/12/2020 |
| **Description:** The function asks for arrive date | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data**  Date =21/12/2020 | **Expected Result**  Expect to get a date: day / month / year.  And change the date of the package to the date we received | **Actual Result**  Arrival date changed | **Status (Pass/Fail)** | **Notes** |
| 2.2.2 | edit\_by\_arrive\_date |  |  |  | pass |  |

|  |  |
| --- | --- |
| **Test Case ID:** 15 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** edit\_by\_leave\_date | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  Verify edit by leave date of package | **Test Execution date** 21/12/2020 |
| **Description:** The function asks for arrive date | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | edit\_by\_leave\_date | Date =21/12/2020 | Expect to get a date: day / month / year.  And change the date of the package to the date we received | Arrival date changed | Pass |  |

|  |  |
| --- | --- |
| **Test Case ID:** 16 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/High):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** Order\_management | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  print Order\_management | **Test Execution date** 21/12/2020 |
| **Description:** The function prints menu. | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Order\_management |  | Expect to print:  [1] Orders awaiting confirmation"  [2] Canceled orders  [3] back to agent | As Expected | Pass |  |

|  |  |
| --- | --- |
| **Test Case ID:** 17 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/high):** high | **Test Designed date:** 21/12/2020 |
| **Module Name:** PendingApprovalOrder | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  approve order | **Test Execution date** 21/12/2020 |
| **Description:** The function update status of order. | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | PendingApprovalOrder | Serial number=8858 | Expect to approve order: | Approve successfully | Pass |  |

|  |  |
| --- | --- |
| **Test Case ID:** 18 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/high):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** get\_mail | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  get valid email | **Test Execution date** 21/12/2020 |
| **Description:** The function return mail. | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** Mail should be up 20 characters ,@ . must include | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | get\_mail | test@gmail.com | Expect to approve order: | Approve successfully | Pass |  |

|  |  |
| --- | --- |
| **Test Case ID:** 19 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/high):** low | **Test Designed date:** 21/12/2020 |
| **Module Name:** get\_mail | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  if email is valid | **Test Execution date** 21/12/2020 |
| **Description:** The function return if email is valid  Email should be up to 20 characters and include @ . | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** none | |
| **Dependencies:** none | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | get\_mail | Test1233#gmail#com | Expect to return false | Get false | Pass |  |
|  |  |  |  |  |  |  |
| 2 | get\_mail | Test@gmail.com | Expect to return true | Get true | pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Test Case ID:** 20 | **Test Designed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Priority (Low/Medium/high):** high | **Test Designed date:** 21/12/2020 |
| **Module Name:** agent\_menu | **Test Executed by** Alex Serdukov, Aviva Malako, Dvir Dishi |
| **Test Title:**  agent menu | **Test Execution date** 21/12/2020 |
| **Description:** The function print agent menu | **Requirement ID** input and output |
|  |  |
|  |  |
| **Pre-conditions:** username and password | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | agent\_menu | Username: harel  Password:123 | Expect to print agent: | print successfully | Pass |  |

|  |  |
| --- | --- |
| **Project Name:** | |
| **Test Case Template** | |
| **Test Case ID:** 21 | **Test Designed by:** elon |
| **Test Priority (Low/Medium/High): med** | **Test Designed date:** 22/12/2020 |
| **Module Name:** Sort by date | **Test Executed by:** elon |
| **Test Title:** Sort packages by date | **Test Execution date:** 22/12/2020 |
| **Description:** The function receives a number of packets and returns them sorted from the near to distant date. | **Requirement ID** : The orders will be represented by categories and chronological order as order date/date of package etc. |
|  |  |
|  |  |
| **Pre-conditions:** The function should have an array with packages and its size. | |
| **Dependencies:** The function depends on the function that copies vacations | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Choose option find vacation | 1 | Find vacation should be open menu for user. | Open Menu. | Pass |  |
| 2 | Choose option 'to view existing package' | 1 |  |  | Pass |  |
| 3 | Choose option 'sort by date | 1 | Open sort by date function | Open sort by date function | Pass |  |
| 4 | Create new array in size that we get | Package\* sort\_by\_date = new Package[size]; | Array of Package in size that we get | Array of Package in size that we get | Pass |  |
| 5 | Copy all vacation to new array from the array that we get | Call to copy vacation and copy all package by for loop | New array with all package | New array with all package | Pass |  |
| 6 | Create temp package | Package temp |  |  | Pass |  |
| 7 | For loop | Copy to temp current package | In temp there is the current package | In temp there is the current package | Pass |  |
| 8 | While loop | Check close date & not deviation from size | true | true | Pass |  |
| 9 | Move package to next location | Call copy vacation  And copy current index to next |  |  | Pass |  |
| 10 | While loop | J<0 && the date in current index small from temp | false | false | Pass |  |
| 11 | Copy to the new array in next index current package - temp | Call copy vacation and copy temp to next index |  |  | Pass |  |
| 12 | For loop | I >=size | false | false | Pass |  |
| 13 | Finish sorting | Return new array | Return new array sort | Return new array sort | Pass |  |

**Post-conditions:** When the rest of the customer wants to sort packages by date, this function is activated and arranges the packages from the near to far date.

**Project Name:**

|  |  |
| --- | --- |
|  | |
| **Test Case Template** | |
| **Test Case ID:**22 | **Test Designed by:** elon |
| **Test Priority (Low/Medium/High):** med | **Test Designed date:** 22/12/2020 |
| **Module Name: sort by price** | **Test Executed by:** elon |
| **Test Title:** Sort packages by price | **Test Execution date:** 22/12/2020 |
| **Description:** The function receives a number of packets and returns them sorted from the small price to big. | **Requirement ID :** **Customers will be able to sort packages by price** |
|  |  |
|  |  |
| **Pre-conditions:** The function should have an array with packages and its size. | |
| **Dependencies:** The function depends on the function that copies vacations. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Choose option find vacation | 1 | Find vacation should be open menu for user. | Open Menu. | Pass |  |
| 2 | Choose option 'to view existing package' | 1 |  |  | Pass |  |
| 3 | Choose option 'sort by price | 2 | Open sort by price function | Open sort by price function | Pass |  |
| 4 | Create new array in size that we get | Package\* sort\_by\_price = new Package[size]; | Array of Package in size that we get | Array of Package in size that we get | Pass |  |
| 5 | Copy all vacation to new array from the array that we get | Call to copy vacation and copy all package by for loop | New array with all package | New array with all package | Pass |  |
| 6 | Create temp package | Package temp |  |  | Pass |  |
| 7 | For loop | Copy to temp current package | In temp there is the current package | In temp there is the current package | Pass |  |
| 8 | While loop | Check close date & not deviation from size | true | true | Pass |  |
| 9 | Move package to next location | Call copy vacation  And copy current index to next |  |  | Pass |  |
| 10 | While loop | J<0 && the price in current index small from temp | false | false | Pass |  |
| 11 | Copy to the new array in next index current package - temp | Call copy vacation and copy temp to next index |  |  | Pass |  |
| 12 | For loop | I >=size | false | false | Pass |  |
| 13 | Finish sorting | Return new array | Return new array sort | Return new array sort | Pass |  |

**Post-conditions:** When the other customer wants to sort packages by price, this function is activated and arranges the packages from low to high.

|  |  |
| --- | --- |
| **Project Name:** | |
| **Test Case Template** | |
| **Test Case ID:**23 | **Test Designed by:** elon |
| **Test Priority (Low/Medium/High):med** | **Test Designed date:** 22/12/2020 |
| **Module Name:** Get\_size\_vacations\_by\_location | **Test Executed by:** elon |
| **Test Title:** Quantity of packages by location | **Test Execution date:** 22/12/2020 |
| **Description:** The function receives packages and returns the number of packages with the desired destination. | **Requirement ID** : filter by location |
|  |  |
|  |  |
| **Pre-conditions:** The function should have an array of packages and a desired location | |
| **Dependencies:** The function performs internal calculations and is independent of any factor | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Check equals | If location = location | Counter++ | Add 1 to counter | pass |  |
| 2 | Return counter | Return counter | Return counter | Return counter | pass |  |

**Post-conditions:** The function received an array of packages kept in a static variable and a location requested by the customer and checked how many packages there are from the same location. The function went through all the vacations in the repository and counted each vacation whose destination was equal to the destination it received as a parameter

|  |  |
| --- | --- |
| **Project Name:** | |
| **Test Case Template** | |
| **Test Case ID:**24 | **Test Designed by:** elon |
| **Test Priority (Low/Medium/High):med** | **Test Designed date:** 22/12/2020 |
| **Module Name:** Get\_vactions\_by\_location | **Test Executed by:** elon |
| **Test Title:** Packages by destination | **Test Execution date:** 22/12/2020 |
| **Description:** A function that receives an array of packages and a location and returns a new array of packages with the resulting location | **Requirement ID** : The customer will be able to filter packages by destination. |
|  |  |
|  |  |
| **Pre-conditions:** The function should get an array of packages and a location that the customer wanted | |
| **Dependencies:** The function depends on the function that returns the amount of vacations by location. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Choose option find vacation | Choice = 1 | Entrance to find vaction | Entered to find vacation | pass |  |
| 2 | Choose option to filter | Choice = 2 | Entrance to package filter | Entered to package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Entered elat to location | pass |  |
| 4 | enter passenger amount | 0 | No filter by passenger amount | passenger amount=0 | pass |  |
| 5 | Enter date | 0 | No filter by date | Date=0 | pass |  |
| 6 | Call to function Get\_vactions\_by\_location |  | Entrance to function | Entered to function | pass |  |
| 7 | Call to function Get\_size vactions\_by\_location |  | Return count packege | Return 2 | pass |  |
| 8 | Create new array |  |  |  | pass |  |
| 9 | Introducing filtered package into the array | Goes through all the DB and puts all the vacations with the destination Eilat | Array with all package with location elat | Array with 2 package with location elat | pass |  |

**Post-conditions:** **The function created a new set and copied all the vacations with the location it received from the customer.**

**Project Name:**

**Test Case Template**

**Test Case ID:25**

**Test Priority (Low/Medium/High):**med **Test Designed by:** michael & elon

**Test Designed date:**22/12/2020

**Module Name:** Get\_size\_vacations\_by\_amount **Test Executed by:** michael & elon

**Test Execution date:**22/12/2020

**Test Title:** Quantity of packages by amount  **Requirement ID:** filter by passengers amount

**Description:** Check that the function returns the exact amount of packages with the desired number of passengers.

**Pre-conditions:** The function should have an array of packages and a desired number of passengers.

**Dependencies:** The function performs internal calculations and is independent of any factor.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Check equals | If amount = amount | Counter++ | Add 1 to counter | pass |  |
| 2 | Return counter | Return counter | Return counter | Return counter | pass |  |

**Post-conditions:** We sent an array of functions with packages and the desired number of passengers to the function and checked that the function returns the amount of packages with the same number of passengers

The function went through all the vacations in the database and counted each vacation whose number of passengers was equal to the number of passengers it received as a parameter

**Project Name:**

**Test Case Template**

**Test Case ID:**26

**Test Priority (Low/Medium/High):**med **Test Designed by:** michael & elon

**Test Designed date:** 22/12/2020

**Module Name:** Get\_vactions\_by\_passenger\_amount **Test Executed by:** michael & elon

**Test Execution date:** 22/12/2020

**Test Title:** Packages by passenger amount  **Requirement ID:** The customer will be able to filter packages by by passenger amount.

**Description:** Check that the function returns an array only with vacations with the desired amount of passengers

**Pre-conditions:** The function should have a set of vacations and a desired number of passengers

**Dependencies:** The function depends on the function that returns the amount of packages with the same amount of passengers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Choose option find vacation | Choice =1 | Entrance to find vacation | Entered to find vacation | pass |  |
| 2 | Choose option to filter | Choice =2 | Entrance to package filter | Entered to package filter |  |  |
|  |  |  |  | Entered to find vacation | pass |  |
| 3 | Enter location | 0 | No filter by date | Location =0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Enter date | 0 | No filter by date | No filter by date | pass |  |
| 6 | Call to function Get\_vactions\_by\_passenger amount |  | Entrance to function | Entered to function | pass |  |
| 7 | Call to function Get\_size\_vactions\_by\_passenger amount |  | Return count of package with 5 passenger amount | Return 2 | pass |  |
| 8 | Create new array | Goes through all the DB and puts in all the vacations with the amount of passengers 5 | Array with 2 packages | Array with 2 packages that passenger amount = 5 | pass |  |
|  |  |  |  |  |  |  |

**Post-conditions**: In the test we started a process from the customer menu to find a package and chose to filter by number of passengers 5 and finally the function we tested created only an array of packages with number of passengers 5.

.

**Project Name:**

**Test Case Template**

**Test Case ID:**27

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:**22/12/2020

**Module Name:** Get\_size\_vacations\_by\_date **Test Executed by:** michael & elon

**Test Execution date:**22/12/2020

**Test Title:** Quantity of packages by date  **Requirement ID:** filter by date

**Description:** Check that the function returns the exact number of packages with the requested date

**Pre-conditions:** The function should have an array of vacations. Check in and check out

**Dependencies:** The function performs internal calculations and is independent of any factor.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Check equals | If date = date | Counter++ | Add 1 to counter | pass |  |
| 2 | Return counter | Return counter | Return counter | Return counter | pass |  |

**Post-conditions**: We sent to the function an array with packages and a requested date and it returned the number of packages with this date.

The function went through all the vacations in the database and counted each vacation whose date was equal to the date it received as a parameter

**Project Name:**

**Test Case Template**

**Test Case ID:**28

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:** 22/12/2020

**Module Name:** Get\_vactions\_by\_date **Test Executed by:** michael & elon

**Test Execution date:** 22/12/2020

**Test Title:** Packages by date  **Requirement ID:** The customer will be able to filter packages by by date

**Description:** Check that the function returns only the vacations with the date it was received

**Pre-conditions:** The function should have an array of vacations. And date of check in and check out.

**Dependencies:** The function depends on the function that returns the number of packets with the desired date.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Choose option find vacation | Choice =1 | Entrance to find vaction | Entered to find vacation | pass |  |
| 2 | Choose option to filter | Choice =2 | Entrance to package filter | Entered to package filter |  |  |
|  |  |  |  | Entered to find vacation | pass |  |
| 3 | Enter location | 0 | No filter by date | Location =0 | pass |  |
| 4 | Enter passenger amount | 0 | No filter by date | passenger amount = 0 | pass |  |
| 5 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 6 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 |  |  |
| 7 | Call to function Get\_vactions\_by\_date |  | Entrance to function | Entered to function | pass |  |
| 8 | Call to function Get\_size\_vactions\_by\_date |  | Return count of package with this date | Return 2 | pass |  |
| 9 | Create new array | Goes through all the DB and puts in all the vacations with this date | Array with 2 packages | Array with 2 packages that check in = 2/2/2021  check out = 5/2/2021 | pass |  |

**Post-conditions:** In the test we started a process from the customer menu to find a package and chose to filter by check-in and check-out date and finally the function we tested created an array only of the packages with the same check-in and check-out we selected.

**Project Name:**

**Test Case Template**

**Test Case ID:**29

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:** 22/12/2020

**Module Name:** Package\_filtering **Test Executed by:** michael & elon

**Test Execution date:** 22/12/2020

**Test Title:** Procurement process by filtering

**Requirement ID:**

The orders will be represented by categories and chronological order as order date/date of package etc.

The agency customers will be able to search package by many categories such as package dates/departure date/destination etc.

**Description:** Checking the purchase of a package with filtering.

**Pre-conditions:** The function should get user name.

**Dependencies:** All the functions of filtering and sorting. Order. Payment. Status update, Find customer, Update Client Wish list, search Package By Serial Number, Print all vacations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **Test 1 - direct purchase without filters and sorts** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Destination = 0  Passenger = 0  Date = 0 |  |  | pass |  |
| 3 | Don't want to sort | 0 | Print package | Print package | pass |  |
| 4 | purchase | Serial number = 2579 | Payment completed successfully | Payment completed successfully | pass |  |
| 5 | Update status | call to function UpdatePackegeStatus | Paid up | Paid up | pass |  |
| **Test 2 - direct purchase without sorts just filter by location** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | elat  Passenger = 0  Date = 0 | Location = elat | Location = elat | pass |  |
| 3 | Don't want to sort | 0 | Print package | Print package | pass |  |
| 4 | purchase | Serial number = 1754 | Payment completed successfully | Payment completed successfully | pass |  |
| 5 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 3 - direct purchase without sorts just filter by passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Location = 0  Passenger = 5  Date = 0 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 3 | Don't want to sort | 0 | Print package | Print package | pass |  |
| 4 | purchase | Serial number = 2298 | Payment completed successfully | Payment completed successfully | pass |  |
| 5 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 4 - direct purchase without sorts just filter by date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Location = 0  Passenger = 0  Date = 1 |  |  | pass |  |
| 3 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 4 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 5 | Don't want to sort | 0 | Print package | Print package | pass |  |
| 6 | purchase | Serial number = 2298 | Payment completed successfully | Payment completed successfully | pass |  |
| 7 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 5 - direct purchase without sorts just filter by location& passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount =5 | passenger amount =5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort? | 0 | No sorting | No sorting | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 1845 | serial number =1845 | serial number =1845 |  |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 6 - direct purchase without sorts just filter by location& date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount =0 | passenger amount =0 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort? | 0 | No sorting | No sorting | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 9475 | serial number =9475 | serial number =9475 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 7 - direct purchase without sorts just filter by location& passenger amount &date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort? | 0 | No sorting | No sorting | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 3829 | serial number =3829 | serial number =3829 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 8 - direct purchase with sort by date and filter by location** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount = 0 | passenger amount = 0 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by date | 1 |  |  | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 1754 | serial number =1754 | serial number =1754 | pass |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 9 - direct purchase with sort by date and filter by passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by date | 1 |  |  | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 2298 | serial number =2298 | serial number =2298 | pass |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 10 - direct purchase with sort by date and filter by date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Location = 0  Passenger = 0  Date = 1 |  |  | pass |  |
| 3 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 4 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 5 | Sort by date | 1 | Print package | Print package | pass |  |
| 6 | purchase | Serial number = 2298 | Payment completed successfully | Payment completed successfully | pass |  |
| 7 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 11 - direct purchase with sort by date and filter by location & passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount =5 | passenger amount =5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by date | 1 |  |  | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 1845 | serial number =1845 | serial number =1845 |  |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 12 - direct purchase without sorts just filter by passenger amount &date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort? | 0 | No sorting | No sorting | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 3829 | serial number =3829 | serial number =3829 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 13 - direct purchase with sort by date and filter by location & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount =0 | passenger amount =0 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by date | 1 | Sorting by date | Sorting by date | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 9475 | serial number =9475 | serial number =9475 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 14 - direct purchase with sort by date and filter by passenger amount & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by date | 1 | Sorting by date | Sorting by date | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 3829 | serial number =3829 | serial number =3829 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 15 - direct purchase with sort by price and filter by location** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount = 0 | passenger amount = 0 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by price | 2 | Sorting by price | Sorting by price | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 1754 | serial number =1754 | serial number =1754 | pass |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 16 - direct purchase with sort by price and filter by passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by price | 2 | Sorting by price | Sorting by price | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 2298 | serial number =2298 | serial number =2298 | pass |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus | Paid up | Paid up | pass |  |
| **Test 17 - direct purchase with sort by price and filter by date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Location = 0  Passenger = 0  Date = 1 |  |  | pass |  |
| 3 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 4 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 5 | Sort by price | 2 | Print package | Print package | pass |  |
| 6 | purchase | Serial number = 2298 | Payment completed successfully | Payment completed successfully | pass |  |
| 7 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 18 - direct purchase with sort by price and filter by location & passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount =5 | passenger amount =5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by price | 2 | Sorting by price | Sorting by price | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 1845 | serial number =1845 | serial number =1845 |  |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 19 - direct purchase with sort by price and filter by location & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount =0 | passenger amount =0 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by price | 2 | Sorting by price | Sorting by price | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 9475 | serial number =9475 | serial number =9475 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 20 - direct purchase with sort by price and filter by passenger amount** **& date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by price | 2 | Sorting by price | Sorting by price | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 3829 | serial number =3829 | serial number =3829 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 21 - direct purchase with sort by price and filter by location& passenger amount** **& date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by price | 2 | Sorting by price | Sorting by price | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 3829 | serial number =3829 | serial number =3829 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 22 - direct purchase with sort by price & date and filter by location** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount = 0 | passenger amount = 0 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by price & date | 3 | Sorting by price & date | Sorting by price & date | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 1754 | serial number =1754 | serial number =1754 | pass |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 23 - direct purchase with sort by price & date and filter by passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by price & date | 3 | Sorting by price & date | Sorting by price & date | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 2298 | serial number =2298 | serial number =2298 | pass |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus | Paid up | Paid up | pass |  |
| **Test 24 - direct purchase with sort by price & date and filter by date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Location = 0  Passenger = 0  Date = 1 |  |  | pass |  |
| 3 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 4 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 5 | Sort by price & date | 3 | Sorting by price & date | Print package | pass |  |
| 6 | purchase | Serial number = 2298 | Payment completed successfully | Payment completed successfully | pass |  |
| 7 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 25 - direct purchase with sort by price & date and filter by location& passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount =5 | passenger amount =5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Sort by price & date | 3 | Sorting by price & date | Sorting by price & date | pass |  |
| 7 |  |  | Print package | Print package | pass |  |
| 8 | Enter serial number | 1845 | serial number =1845 | serial number =1845 |  |  |
| 9 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 10 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 26 - direct purchase with sort by price & date and filter by location& date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount =0 | passenger amount =0 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by price & date | 3 | Sorting by price & date | Sorting by price & date | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 9475 | serial number =9475 | serial number =9475 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 27 - direct purchase with sort by price & date and filter by passenger amount & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by price & date | 3 | Sorting by price & date | Sorting by price & date | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 3829 | serial number =3829 | serial number =3829 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 28 - direct purchase with sort by price & date and filter by location& passenger amount & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Sort by price & date | 3 | Sorting by price & date | Sorting by price & date | pass |  |
| 9 |  |  | Print package | Print package | pass |  |
| 10 | Enter serial number | 3829 | serial number =3829 | serial number =3829 | pass |  |
| 11 | purchase |  | Payment completed successfully | Payment completed successfully | pass |  |
| 12 | Update status | Call to UpdatePackegeStatus function | Paid up | Paid up | pass |  |
| **Test 29 - add to wish list and filter by location** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount = 0 | passenger amount = 0 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 7 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 8 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 9 | The package added |  | The package added | The package added | pass |  |
| 10 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 30 - add to wish list and filter by passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 7 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 8 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 9 | The package added |  | The package added | The package added | pass |  |
| 10 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test 31 - add to wish list and filter by date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Location = 0  Passenger = 0  Date = 1 |  |  | pass |  |
| 3 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 4 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 5 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 6 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 7 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 8 | The package added |  | The package added | The package added | pass |  |
| 9 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 32 - add to wish list and filter by location& passenger amount** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount =5 | passenger amount =5 | pass |  |
| 5 | Filter by date? | 0 | No filter by date | No filter by date | pass |  |
| 6 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 7 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 8 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 9 | The package added |  | The package added | The package added | pass |  |
| 10 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 33 - add to wish list and filter by location& date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 0 | passenger amount =0 | passenger amount =0 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 9 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 10 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 11 | The package added |  | The package added | The package added | pass |  |
| 12 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 34 - add to wish list and filter by passenger amount & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | 0 | Location = 0 | Location = 0 | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 9 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 10 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 11 | The package added |  | The package added | The package added | pass |  |
| 12 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 35 - add to wish list and filter by location &passenger amount & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Choice =2 | Open package filter | Open package filter | pass |  |
| 3 | Enter location | elat | Location = elat | Location = elat | pass |  |
| 4 | Enter passenger amount | 5 | passenger amount = 5 | passenger amount = 5 | pass |  |
| 5 | Filter by date? | 1 |  |  | pass |  |
| 6 | Enter date – check in | 2/2/2021 | check in = 2/2/2021 | check in = 2/2/2021 | pass |  |
| 7 | Enter date – check out | 5/2/2021 | check out = 5/2/2021 | check out = 5/2/2021 | pass |  |
| 8 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 9 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 10 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 11 | The package added |  | The package added | The package added | pass |  |
| 12 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 36 - add to wish list without filter** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Choose option to filter | Destination = 0  Passenger = 0  Date = 0 |  |  | pass |  |
| 3 | Add to wish list | 4 | Check = 4 | Check = 4 | pass |  |
| 4 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 5 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 6 | The package added |  | The package added | The package added | pass |  |
| 7 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |

**Post-conditions:** In the tests we performed in the function we tried all the options that the customer can perform in the purchase process by filters and filters**. It should be noted that before the tests I noticed that there was no call to the function that filters by date.** I also performed the tests on the same packages in DB all the time with the same values ​​of destination and number of passengers and date because the tests of the functionality of the functions of the filters and filters were performed separately and were successful.

**Project Name:**

**Test Case Template**

**Test Case ID:**30

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:**23/12/2020

**Module Name:** Order status. **Test Executed by:** michael & elon

**Test Execution date:**23/12/2020

**Test Title:** Tracking the status of orders

**Requirement ID:** A customer will be able to cancel an order

**Description:** Check that the customer sees the status of the orders and can cancel a package.

**Pre-conditions:** The function should have a username.

**Dependencies:** The function depends on a function that returns a customer by username and a function that updates status.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **Test 1 View package status and cancel a package** | | | | | | |
| 1 | Choose option order status | Choice =2 | Open order status function | Get in to function | pass | 1 |
| 2 | Find client | Call to searchClientByUserName | Index for this client | Index for this client | pass |  |
| 3 | Find client | Call with index to getClientByIndex | Client we sent his username | Client we sent his username | pass |  |
| 4 |  |  | Print package | Print package | pass |  |
| 5 | To cancel press 1 | 1 | Call to UpdatePackegeStatus | Call to UpdatePackegeStatus | pass |  |
| 6 |  |  | The update sucsses | The update sucsses | pass |  |
| 7 | Back to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 2 just View package status without cancel a package** | | | | | | |
| 1 | Choose option order status | Choice =2 | Open order status function | Get in to function | pass |  |
| 2 | Find client | Call to searchClientByUserName | Index for this client | Index for this client | pass |  |
| 3 | Find client | Call with index to getClientByIndex | Client we sent his username | Client we sent his username | pass |  |
| 4 |  |  | Print package | Print package | pass |  |
| 5 | To cancel press 1 | 0 |  |  | pass |  |
| 6 |  |  | Print all packege and status | Print all packege and status | pass |  |
| 7 | Back to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 3 no packages for this client** | | | | | | |
| 1 | Choose option order status | Choice =2 | Open order status function | Get in to function | pass |  |
| 2 | Find client | Call to searchClientByUserName | Index for this client | Index for this client | pass |  |
| 3 | Find client | Call with index to getClientByIndex | Client we sent his username | Client we sent his username | pass |  |
| 4 |  |  | **No active orders** | **No active orders** |  |  |
| 5 | Back to client menu | Call to client menu | Open client menu | Open client menu | pass |  |

**Post-conditions:** In the review we entered from the menu to Order Status we saw all the purchases and their status and canceled a purchase and received a message that the update was successful.

**Project Name:**

**Test Case Template**

**Test Case ID:**31

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:**26/12/2020

**Module Name:** order vacation **Test Executed by:** michael & elon

**Test Execution date:** 26/12/2020

**Test Title:** Purchasing process

**Requirement ID:**

the order/contact form will have all the needed details, such as name, phone number, mail etc. that way the travel agent will not experience difficulties trying to make contact.

**Description:** As part of the purchase process, checking the receipt of details from the customer and updating them in the system and updating the status of the package

**Pre-conditions:** The function should get a serial number of the package and a username

**Dependencies:** The function depends on the following functions: searchClientByUserName , searchPackegeBySerialNumber, UpdatePackegeStatus, printPackege, getPackegeByIndex, searchPackegeBySerialNumber, UpdateClientsOrderDetails.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **Test 1 Ordering process without checking input errors** | | | | | | |
| 1 | Check serial number and user name | Calling for search functions | Value that return != -1 | False (The test is normal) | pass |  |
| 2 | Update package status | Call to UpdatePackegeStatus | Change status to in process | Change status to in process | pass |  |
| 3 | To order press 1 | 1 | Choice = 1 | Choice = 1 | pass |  |
| 4 | Enter name | elon | Name = elon | Name = elon | pass |  |
| 4 | Last name | ifrah | Last name = ifrah | Last name = ifrah | pass |  |
| 5 | Phone number | 0544445176 | Phone number = 0544445176 | Phone number = 0544445176 | pass |  |
| 6 | Update details | Call to UpdateClientsOrderDetails | true | true | pass |  |
| 7 | Return value | Return true | Back to however read | Back to however read | pass |  |
| **Test 2 Ordering process with checking input errors** | | | | | | |
| 1 | Check serial number and user name | Calling for search functions | Value that return != -1 | False (The test is normal) | pass |  |
| 2 | Update package status | Call to UpdatePackegeStatus | Change status to in process | Change status to in process | pass |  |
| 3 | To order press 1 | 1 | Choice = 1 | Choice = 1 | pass |  |
| 4 | Enter name | elon | Name = elon | Name = elon | pass |  |
| 4 | Last name | ifrah | Last name = ifrah | Last name = ifrah | pass |  |
| 5 | Phone number | 054444517 | Phone number = 054444517 | Phone number = 054444517 | fail |  |
| 6 | Incorrect phone number try again | 0544445176 | Phone number = 0544445176 | Phone number = 0544445176 | pass |  |
| 7 | Update details we sent wrong user name | Call to UpdateClientsOrderDetails | false | false | fail |  |
| 8 | Return value | Return false | Back to however read | Back to however read | pass |  |
| Test 3 Stop the process and return to the menu | | | | | | |
| 1 | Check serial number and user name | Calling for search functions | Value that return != -1 | False (The test is normal) | pass |  |
| 2 | Update package status | Call to UpdatePackegeStatus | Change status to in process | Change status to in process | pass |  |
| 3 | To order press 1 | 0 | Choice = 0 | Choice = 0 | pass |  |
| 4 | Update package status | Call to UpdatePackegeStatus | Change status to none | Change status to none | pass |  |
| 5 | Return value | Return false | Back to however read | Back to however read | pass |  |

**Post-conditions:** At the beginning of the test the function was given a username and package number. It prints the sabotage and allows you to continue ordering or return to the main menu. If the customer chooses to continue she takes personal details as shown in the requirements document and updates the order details. The function returns true if the order continues otherwise returns false**.**

**Project Name:**

**Test Case Template**

**Test Case ID:**32

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:** 27/12/2020

**Module Name:** find vacation **Test Executed by:** michael & elon

**Test Execution date:** 27/12/2020

**Test Title:** The process of finding a vacation and purchasing

**Requirement ID:** The agency customers will be able to make a reservation, and the quantity will be updated accordingly

**Description:** Check that the customer can search for a package and make a purchase

**Pre-conditions:** the function should have a username.

**Dependencies:** The function depends: Print all vacations, get All Packages, Packages file Size, search Package By Serial Number, sort by price, sort by date, Find customer, Update Client Wish list, Order vacation, Payment, Update Package Status.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| Test 1 view existing packages without sort | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
|  |  |  |  |  | pass |  |
| 3 | without sort | 0 | Print all DB | Print all DB |  |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Order vacation | Call order vacation sent serial number and username | true | true | pass |  |
| 7 | payment | Call payment sent serial number and username | true | true | pass |  |
| 8 |  |  | Payment completed successfully | Payment completed successfully | pass |  |
| 9 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 2 view existing packages and sort by date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Sort by date | 1 | Print all DB with sorting | Print all DB with sorting |  |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Order vacation | Call order vacation sent serial number and username | true | true | pass |  |
| 7 | payment | Call payment sent serial number and username | true | true | pass |  |
| 8 |  |  | Payment completed successfully | Payment completed successfully | pass |  |
| 9 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 3 view existing packages and sort by price** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Sort by price | 2 | Print all DB with sorting | Print all DB with sorting |  |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Order vacation | Call order vacation sent serial number and username | true | true | pass |  |
| 7 | payment | Call payment sent serial number and username | true | true | pass |  |
| 8 |  |  | Payment completed successfully | Payment completed successfully | pass |  |
| 9 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 4 view existing packages and sort by price & date** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Sort by price & date | 3 | Print all DB with sorting | Print all DB with sorting |  |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Order vacation | Call order vacation sent serial number and username | true | true | pass |  |
| 7 | payment | Call payment sent serial number and username | true | true | pass |  |
| 8 |  |  | Payment completed successfully | Payment completed successfully | pass |  |
| 9 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 5 view existing packages and add to wish list** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Add to wish list | 4 | Print all DB with sorting | Print all DB with sorting | pass |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 7 | Update client wish list | Call to Update client wish list function | true | true | pass |  |
| 8 | The package added |  | The package added | The package added | pass |  |
| 9 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 6 filter packages** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Filter packages | 2 | Open package filter function | package filter open | pass | The process continue there |
| **Test 7 back to menu** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | Back to menu | 3 | Return to client menu | The client menu open | pass |  |
| **Test 8 check serial number input** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Sort by price | 2 | Print all DB with sorting | Print all DB with sorting |  |  |
| 4 | Enter serial number | 1 | serial number = 1 | serial number = 1 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | false | false | pass |  |
| 6 | Enter again serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 7 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 8 | Order vacation | Call order vacation sent serial number and username | true | true | pass |  |
| 9 | payment | Call payment sent serial number and username | true | true | pass |  |
| 10 |  |  | Payment completed successfully | Payment completed successfully | pass |  |
| 11 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 9 check payment fails** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Sort by price | 2 | Print all DB with sorting | Print all DB with sorting |  |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Order vacation | Call order vacation sent serial number and username | true | true | pass |  |
| 7 | payment | Call payment sent serial number and username | false | false | pass |  |
| 8 | Update status package to none | Call to UpdatePackegeStatus | Payment not successfully | Payment not successfully | pass |  |
| 9 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 10 check order vacation fails** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Sort by price | 2 | Print all DB with sorting | Print all DB with sorting |  |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Order vacation | Call order vacation sent serial number and username | false | false | pass |  |
| 7 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |
| **Test 11 check update client wish list fails** | | | | | | |
| 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  |
| 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  |
| 3 | Add to wish list | 4 | Print all DB with sorting | Print all DB with sorting | pass |  |
| 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  |
| 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  |
| 6 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  |
| 7 | Update client wish list | Call to Update client wish list function | false | false | pass |  |
| 8 |  |  | The package not added | The package not added | pass |  |
| 9 | Back to menu | Call to client menu | Open client menu | Open client menu | pass |  |

**Post-conditions:** In the test we performed, we entered the menu as a customer and chose to find a vacation. The function manages the process of finding a vacation without filters or with filters. With sorts and without sorts

**Project Name:**

**Test Case Template**

**Test Case ID:**33

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:**27/12/2020

**Module Name:** find customer **Test Executed by:** michael & elon

**Test Execution date:**27/12/2020

**Test Title:** Find customer from DB by user name  **Requirement ID:**

**Description:** The function gets a username and returns the appropriate customer

**Pre-conditions:** The function should have a username

**Dependencies:** the function depends on the functions: get all clients, client file size, client menu.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Create new array | Call to get all clients | Array of all DB client | Array of all DB client | pass |  |
| 2 | Create size | Call to client file size | Size = amount of clients | Size = amount of clients | pass |  |
| 3 | For loop | I<size | true | true | pass |  |
| 4 | Find customer | Usrename = username of client in current index | Return client in current index | Return client in current index | pass |  |
| 5 | For loop | I >=size | Customer not found by username | Customer not found by username | pass |  |
| 6 | Enter username | elon | Username = elon | Username = elon | pass |  |
| 7 | Back to client menu | Call to client menu | Open client menu | Client menu open | pass |  |

**Post-conditions**: The function got a username and ran on all the customers' DB and as soon as it found the customer it returned it and if the customer is not found it picks up a new username and returns to the customer menu

**Project Name:**

**Test Case Template**

**Test Case ID:**34

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:**27/12/2020

**Module Name:** find customer2 **Test Executed by:** michael & elon

**Test Execution date:**27/12/2020

**Test Title:** Find customer from DB by user name  **Requirement ID:**

**Description:** The function gets a username and returns pointer to the appropriate customer

**Pre-conditions:** The function should have a username

**Dependencies:** the function depends on the functions: get all clients, client file size, client menu.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Create new array | Call to get all clients | Array of all DB client | Array of all DB client | pass |  |
| 2 | Create size | Call to client file size | Size = amount of clients | Size = amount of clients | pass |  |
| 3 | For loop | I<size | true | true | pass |  |
| 4 | Find customer | Usrename = username of client in current index | Return client in current index | Return client in current index | pass |  |
| 5 | For loop | I >=size | Customer not found by username | Customer not found by username | pass |  |
| 6 |  |  | Customer not found | Customer not found | pass |  |
| 7 | Return null | Return null | Return null | Return null | pass |  |

**Post-conditions**: The function got a username and ran on all the customers' DB and as soon as it found the customer it returned pointer to it and if the customer is not found it return null.

**Project Name:**

**Test Case Template**

**Test Case ID:**35

**Test Priority (Low/Medium/High):** med **Test Designed by:** michael & elon

**Test Designed date:** 27/12/2020

**Module Name:** print wish list **Test Executed by:** michael & elon

**Test Execution date:** 27/12/2020

**Test Title:** print wish list

**Requirement ID:** The customer will be able to add packages to the wish list and make purchases through there

**Description:** The function prints the packages that the customer has in the wish list

**Pre-conditions:** the function get a client

**Dependencies:** the function depends on the functions: get all clients, client file size, print vacation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Create new array | Call to get all clients | Array of all DB client | Array of all DB client | pass |  |
| 2 | Create size | Call to client file size | Size = amount of clients | Size = amount of clients | pass |  |
| 3 | For loop for array customer | I< 20 | true | true | pass |  |
|  | Check current index != 0 | customer.wish\_list[i] != 0 | true | true | pass |  |
|  | For loop for array DB | J < size | true | true | pass |  |
|  | Check serial number of current index of customer with serial number of current index of DB | vacations[j].serial\_number == customer.wish\_list[i] | TRUE | TRUE | pass |  |
|  | Print package from current index in array DB | Call to print vacation | Print package | Print package | pass |  |

**Post-conditions:** The function goes through the customer's array and prints all the packages he puts on the wish list.

**Project Name:** Harel travels

**Test Case Template**

**Test Case ID:**36

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** michael

**Test Designed date:**22/12/2020

**Module Name:** Copy date **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Performs copying from source to target  **Requirement ID:**

**Description:** Test if The function gets a date and copies the source to target.

**Pre-conditions:** Function need to get a date .

**Dependencies:** The function performs an ancillary process related to the system and not to the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Get a Date(day , month ,year) |  |  |  | pass |  |
| 2 | Copy the day from source to target |  |  |  | pass |  |
| 3 | Copy the month from source to target |  |  |  | pass |  |
| 4 | Copy the year from source to target |  |  |  | pass |  |
|  |  |  |  |  |  |  |

**Post-conditions: The function was given a date and copied it according to its fields as required.**

**Project Name:**Harel travels

**Test Case Template**

**Test Case ID:**37

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** michael

**Test Designed date:**22/12/2020

**Module Name:** Copy vacation **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Performs copying from source to target  **Requirement ID:**

**Description:** Test if function gets a vacation and copies the source to target.

**Pre-conditions:** Function need to get a vacation.

**Dependencies:** The function depends on perfection of the 'copy\_date' function.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Get a vacation |  |  |  | pass |  |
| 2 | Copy from source to target all fields of vacation, |  |  |  | pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:** **The function is used by us in the filtering and sorting functions and its integrity check is performed in tests of the sorting and filtering functions.**

**Project Name:** Harel travels

**Test Case Template**

**Test Case ID:**38

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** michael

**Test Designed date:**22/12/2020

**Module Name:** Payment(wish list-make order) **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Receives a user and checks the correctness of the means of payment  **Requirement ID:**

**Description:** Test if the function Checking payment method correctly.

**Pre-conditions:** User need to be connected to system.

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Logged in | User :michael  Password:123 | Should logged in to system and open menu | Logged in and open menu | pass |  |
| 2 | Select 'wish list' | Press :3 | Should print all package in wish list and ask for make an order | Print all package in wishlist and ask for make an order | pass |  |
| 3 | Place an order | Press: 1 | Should ask for a serial number | Asked for a serial number | pass |  |
| 4 | Choose vacation | Input serial number:2680828 | Should go to payment | Go to payment | pass |  |
|  | \*\*\*Payment\*\*\* |  |  |  |  |  |
| 5 | Filling details | First name:michael  Last name : harush  Id:208829085  Creditcard:45800901087326201 (over 16) | Should print error message(I write card number >16) | print error message:"the credit card nust be with 16 digits ,try again" | pass |  |
| 6 | Enter valid card number | CreditCard:4580090108732620 | Should get the card number and print the next step (CVC) | Get the card number and ask for CVC | pass |  |
| 7 | Incorrect filling of cvc | Cvc:1234 | Should print error message | Print error message : "CVC must be 3 digits exactly ,try again" | pass |  |
| 8 | Filling correct CVC | CVC:123 | Should take me back to wish list and print "payment completed successfully " | Print " payment completed successfully" | pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:** **When attempting to purchase from the wish list I was transferred to the payment function within which a proper process of checking data and making a payment was carried out.**

**Project Name:**Harel travels

**Test Case Template**

**Test Case ID:**39

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** michael &elon

**Test Designed date:**23/12/2020

**Module Name:** message **Test Executed by:** michael & elon

**Test Execution date:** 23/12/2020

**Test Title:** contact  **Requirement ID:**

**Description:** Test contact and view existing messages

**Pre-conditions:** The function need to get a user name (customer).

**Dependencies:** The function Depends on three other functions: Updating a message at a customer, printing a message, create new message.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Choose in menu 'message' |  | Should open the message | Open the message | pass |  |
| 2 | Select '0' if want to read your message or '1' write a new message | Select:'0' | Should show all messages | Show all messages | pass |  |
| 3 | Select to write a new message | Title: order  Contact: I want to know what about my order. | Should print all messages include the new message | Print all message include the new message | pass |  |

**Post-conditions:** The function received a user and gave two options: read existing messages or add a message and print all the messages

**Project Name:** Harel travels

**Test Case Template**

**Test Case ID:**40

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** Michael

**Test Designed date:**22/12/2020

**Module Name:** Payment(package filtering-destination ,amount) **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Receives a user and checks the correctness of the means of payment  **Requirement ID:**

**Description:** Test if the function Checking payment method correctly.

**Pre-conditions:** User need to be connected to system, The database should contain vacations.

**Dependencies:** all function of filters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Logged in | User: michael  Password:123 |  |  | pass |  |
| 2 | Select 'find vacation' | Press :1 | Should open find vacation function | Open find vacation function | pass |  |
| 3 | Select 'package filtering' | Press: 2 | should open package filtering function | Open package filtering | pass |  |
| 4 | Enter destination | Destination: London | Should get destination and ask for passenger amount | Get and ask for passenger amount | pass |  |
| 5 | Enter passenger amount | Passenger amount:2 | Should get passenger amount and ask for check in and out | Get passenger amount and ask for check in and out | pass |  |
| 6 | continue | Press:0 | Should ask if I want to sort | Ask if I want to sort | pass |  |
| 7 | Print package result without sorting | Press:0 | Should print all package with the destination and passenger amount I ask for | Prints all packages suitable for filtering | pass |  |
| 8 | Select serial number | Serial number:2088290 | Should get serial number and move on to payment | Get serial number and move on to payment | pass |  |
| 9 | Filling details | First name: michael  Last name: harush  Id:208829085  Creditcard:4580090108732620 | Should get the cardNumber and ask for expiring date | Get the card Number and ask for expiring date | pass |  |
| 10 | Filling expiring date | Expiring date  Month:06  Year:26 | Should get expiring date and ask for cvc | Get expiring date and ask for CVC | pass |  |
| 11 | Filling CVC | CVC:123 | Should print "payment complete successfully" | Print "payment complete successfully" | pass |  |

**Post-conditions: I got to the payment option through Find vacation and Package Filter.**

**Project Name:**Harel travels

**Test Case Template**

**Test Case ID:**41

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** Payment(find vacation) **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Receives a user and checks the correctness of the means of payment  **Requirement ID:**

**Description:** Test if the function Checking payment method correctly.

**Pre-conditions:** User need to be connected to system.

**Dependencies:** This possibility can be reached from several function : Find\_vacation ,package filtering and wish list.

Each case test will be given in the following tests

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Logged in | User: michael  Password:123 |  |  | pass |  |
| 2 | Select 'find vacation' | Press :1 | Should open find vacation function(menu) | Open find vacation function | pass |  |
| 3 | Select 'to view existing package' | Press: 1 | should asked if I want to sort | Asked if I want to sort | pass |  |
| 4 | Select 'If you don’t want to sort' | Press 0 | Should print all packages and asked for serial number | Print all packages and asked for serial number | pass |  |
| 5 | Enter serial number | Serial number:280909 | Should get serial number and move on to payment | Get serial number and move on to payment | pass |  |
|  | \*\*\*Payment\*\*\* |  |  |  |  |  |
| 6 | Filling details | First name: michael  Last name: harush  Id:208829085  Creditcard:4580090108732620 | Should get the cardNumber and ask for expiring date | Get the card Number and ask for expiring date | pass |  |
| 7 | Filling expiring date | Expiring date  Month:06  Year:26 | Should get expiring date and ask for cvc | Get expiring date and ask for CVC | pass |  |
| 8 | Filling CVC | CVC:123 | Should print "payment complete successfully" | Print "payment complete successfully" | pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions: I got to the payment option through Find vacation**

**Project Name:** Harel travels

**Test Case Template**

**Test Case ID:**42

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** wish list(return menu) and make order from wish list **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** get a user and print is wish list package, and order vacation **Requirement ID:**

**Description:** test if the function print wish list and let me back to menu and back to wish list to make order

**Pre-conditions:** the function have to get user name and password and should be package in DB

**Dependencies:** payment ,client menu, print package

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Logged in | User :michael  Password:123 | Should logged in to system and open menu | Logged in and open menu | pass |  |
| 2 | Select 'wish list' | Press :3 | Should print all package in wish list and ask for make an order or back to menu | Print all package in wishlist and ask for make an order or bace to menu | pass |  |
| 3 | Back to menu | Press: 0 | Should take me back to menu | Take me back to menu | pass |  |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Add package to wish list and make order** | | | | | | | | 1 | Choose option find vacation | Choice =1 | Open menu for find package | The menu open | pass |  | | 2 | view existing packages | 1 | Choice =1 | Choice =1 | pass |  | | 3 | Add to wish list | 4 | Print all DB with sorting | Print all DB with sorting | pass |  | | 4 | Enter serial number | 3829 | serial number = 3829 | serial number = 3829 | pass |  | | 5 | Check serial number | Call to function searchPackegeBySerialNumber | true | true | pass |  | | 6 | Find customer | Call to find customer and sent user name | To get customer whose username it this | Requested customer was received | pass |  | | 7 | Update client wish list | Call to Update client wish list function | true | true | pass |  | | 8 | The package added |  | The package added | The package added | pass |  | | 9 | Return to client menu | Call to client menu | Open client menu | Open client menu | pass |  | | 10 | Select 'wish list' | Press :3 | Should print all package in wish list and ask for make an order | Print all package in wishlist and ask for make an order | pass |  | | 11 | Place an order | Press: 1 | Should ask for a serial number | Asked for a serial number | pass |  | | 12 | Choose vacation | Input serial number:3829 | Should go to payment | Go to payment | pass |  | | 13 | \*\*\*Payment\*\*\* |  |  |  |  |  | | 14 | Filling details | First name:michael  Last name : harush  Id:208829085  Creditcard:45800901087326201 (over 16) | Should print error message(I write card number >16) | print error message:"the credit card nust be with 16 digits ,try again" | pass |  | | 15 | Enter valid card number | CreditCard:4580090108732620 | Should get the card number and print the next step (CVC) | Get the card number and ask for CVC | pass |  | | 16 | Incorrect filling of cvc | Cvc:1234 | Should print error message | Print error message : "CVC must be 3 digits exactly ,try again" | pass |  | | 17 | Filling correct CVC | CVC:123 | Should take me back to wish list and print "payment completed successfully " | Print " payment completed successfully" | pass |  | | | | | | | |

**Post-conditions: I got to the payment option through Wishlist.**

**Project Name:**Harel travels

**Test Case Template**

**Test Case ID:**43

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** michael&elon

**Test Designed date:**22/12/2020

**Module Name:** Check Date **Test Executed by:** Michael&elon

**Test Execution date:** 22/12/2020

**Test Title:** The function get a date and check if there is package with this date. **Requirement ID:**

**Description:** Test if function actually checked

**Pre-conditions:** the function should get 2 date.

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| **Test 1 Check equals date** | | | | | | |
| 1 | Sent date | 1/1/2020 , 1/1/2020 | Date1 =1/1/2020, date2=1/1/2020 | Date1 =1/1/2020, date2=1/1/2020 |  |  |
| 2 | Day comparison | date1.day == date2.day | true | true |  |  |
| 3 | Month comparison | date1.month == date2.month | true | true |  |  |
| 4 | year comparison | date1. year == date2. year | true | true |  |  |
| 5 | Value return | Return true | true | true |  |  |
| **Test 2 Check not equals date** because day | | | | | | |
| 1 | Sent date | 2/1/2020 , 1/1/2020 | Date1 =2/1/2020, date2=1/1/2020 | Date1 =2/1/2020, date2=1/1/2020 |  |  |
| 2 | Day comparison | date1.day == date2.day | false | false |  |  |
| 3 | Month comparison | date1.month == date2.month | true | true |  |  |
| 4 | year comparison | date1. year == date2. year | true | true |  |  |
| 5 | Value return | Return true | false | false |  |  |
| **Test 3 Check not equals date** because month | | | | | | |
| 1 | Sent date | 1/2/2020 , 1/1/2020 | Date1 =1/2/2020, date2=1/1/2020 | Date1 =1/2/2020, date2=1/1/2020 |  |  |
| 2 | Day comparison | date1.day == date2.day | true | true |  |  |
| 3 | Month comparison | date1.month == date2.month | false | false |  |  |
| 4 | year comparison | date1. year == date2. year | true | true |  |  |
| 5 | Value return | Return true | false | false |  |  |
| **Test 4 Check not equals date** because year | | | | | | |
| 1 | Sent date | 1/1/2020 , 1/1/2021 | Date1 =1/1/2020, date2=1/1/2021 | Date1 =1/1/2020, date2=1/1/2021 |  |  |
| 2 | Day comparison | date1.day == date2.day | true | true |  |  |
| 3 | Month comparison | date1.month == date2.month | true | true |  |  |
| 4 | year comparison | date1. year == date2. year | false | false |  |  |
| 5 | Value return | Return true | false | false |  |  |

**Post-condition:** In our review, we covered all the options: an equal date. And dates are not equal due to day / month / year

**Test Case Template**

**Test Case ID:**44

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** michael

**Test Designed date:**22/12/2020

**Module Name:** print all vacations **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** print all vacation in DB  **Requirement ID:**

**Description:** Test if function print all vacations from DB.

**Pre-conditions:** DB should contain vacations.

**Dependencies:** depends on the function 'print package' .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Logged in | User: michael  Password:123 |  |  | pass |  |
| 2 | Select 'find vacation' | Press :1 | Should open find vacation function(menu) | Open find vacation function | pass |  |
| 3 | Select 'to view existing package' | Press: 1 | should asked if I want to sort | Asked if I want to sort | pass |  |
| 4 | Select 'If you don’t want to sort' | Press 0 | Should print all packages and asked for serial number | Print all packages and asked for serial number | pass |  |

**Post-conditions:** **To show that the function prints all the vacations I came to it through the 'find vacation' option and it did the job and printed all the packages in the database.**

**Test Case Template**

**Test Case ID:**45

**Test Priority (Low/Medium/High):** Medium **Test Designed by:** michael

**Test Designed date:**22/12/2020

**Module Name:** home page **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Home page of the system call the main menu  **Requirement ID:**

**Description:** Test if home page call main menu

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Open system |  | Should print the home page and open the main menu | Print the home page and open the main menu | pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:** **In checking the home page I checked that the home page is actually printed and there is a move to the main menu.**

**Test Case Template**

**Test Case ID:**46

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** main menu & log in as client(log in

After choose login as client &register user) **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** option log in as agent or log in/register as client  **Requirement ID:**

**Description:** Test the options of the function main menu that including log in as client .

**Pre-conditions:** valid user name and password.

**Dependencies:** perfection login and registration functions.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Login as client | Press:1 | Should open menu of client(log in/register) | open menu of client(log in/register) | pass |  |
| 2 | Log in | Press:1 | Should Ask for user name and password | Ask for user name and password | pass | The function log in after choose log in as client. |
| 3 | Enter user name and wrong password | User: michael  Password:12(wrong) | Should print error message and let me try again | Print error message and try again | pass |  |
| 4 | Enter good password | Password:123 | Should take me to client menu | Take me to client menu | pass |  |
| 5 | Log out | Press:5 | Should take me to main menu again | Take me to main menu | pass |  |
| 6 | Login as client | Press:1 | Should open menu of client(log in/register) | open menu of client(log in/register) | pass |  |
| 7 | register | Press:2 | Should open register and ask for details | Open register and ask for details | pass | Function register user |
| 8 | fill in details | User name: michael  Password:1234  Mail: michael@gmail.com | Should print 'create has success  And open client menu  ' | Print "create has success' and open client menu | pass |  |

**Post-conditions: From The main menu function I choose log in as client after I succeed I back to main menu and choose register Both processes worked**

**Test Case Template**

**Test Case ID:**47

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name: main menu(login as agent)** **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** log in as agent from the main menu  **Requirement ID:**

**Description:** Test the option login as agent from the main menu

**Pre-conditions:** valid user name and password of agent

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Log in as agent | Press:2 | Should asked me for user name and password | Asked for user name and password | pass |  |
| 2 | Fill details | User name: micha  Password:123 | Should logged in and show me agent menu | open agent menu | pass |  |

**Post-conditions:** **Through the main menu I logged in as an agent and the process worked.**

**Test Case Template**

**Test Case ID:**48

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** client menu **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Client menu options  **Requirement ID:**

**Description:** Test all options the client have

**Pre-conditions:** valid user name and password

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Login as client | Press:1 | Should open menu of client(log in/register) | open menu of client(log in/register) | pass |  |
| 2 | Log in | Press:1 | Should Ask for user name and password | Ask for user name and password | pass |  |
| 3 | Enter user name and password | User: michael  Password:123 | Should take me to client menu | Take me to client menu | pass |  |
| 4 | Find a vacation | Press 1 | Should take me to find vacation | Take me to find vacation | pass |  |
| 5 | Back to client menu | Press:3 | Should take me back to client menu | Take me back to client menu | pass |  |
| 6 | Order status | Press :2 | Should take me to order status | Take me to order status | pass |  |
| 7 | Back to client menu |  |  |  | pass |  |
| 8 | Wish list | Press:3 | Should take me to wish list | Take me to wish list | pass |  |
| 9 | Back to menu | Press:0 | Should take me back to client menu | Take me back to client menu | pass |  |
| 10 | messages | Press:4 | Should take me to messages | Take me to messages | pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:** **In checking the client menu I went into each option and saw that it brings me to the right place**

**Test Case Template**

**Test Case ID:**49

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** check if user name exist **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Function get user name and check if user name exist in DB  **Requirement ID:**

**Description:** Test if the function actually check if there is or there is not specific name in DB

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Login as client | Press:1 | Should open menu of client(log in/register) | open menu of client(log in/register) | pass |  |
| 2 | register | Press:2 | Should open register and ask for details | Open register and ask for details | pass |  |
| 3 | user name | User name:michael | Should send to the function 'check if user name exist' and should return false | Send to the function and return false("name already exist ,try again) | pass |  |
| 4 | Enter new user name | User:michch | Should send to the function 'check if user name exist' and should return true | Send to the function and return true(the next level enter password) | pass |  |

**Post-conditions:** **Through registration as a client I checked the function, once I typed an existing name and the second time I typed a new name the function performed what it had to perform.**

**Test Case Template**

**Test Case ID:**50

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** get input **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** check string  **Requirement ID:**

**Description:** A function that receives strings

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Log in - customer | 1 | Log in / register | Log in / register | pass |  |
| 2 | register | 2 |  |  | pass |  |
| 3 | Enter username | moshe | Username = Moshe | Username = Moshe | pass |  |
| 4 | Return string |  | Return Moshe | Return Moshe | pass |  |

**Post-conditions:**

**Test Case Template**

**Test Case ID:**51

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** get password **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Asked for password and check it in DB  **Requirement ID:**

**Description:** Test if function Asked for password and checked password if password available

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Login as client | Press:1 | Should open menu of client(log in/register) | open menu of client(log in/register) | pass |  |
| 2 | Log in | Press:1 | Should Ask for user name and password | Ask for user name and password | pass |  |
| 3 | Enter user name and password | User: michael  Password:123 | Should send password to function and open client menu | Send password to function and open client menu | pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:** **Through as a client I checked the correctness of the function, she received a password and performed her work.**

**Test Case Template**

**Test Case ID:**52

**Test Priority (Low/Medium/High):** Medium **Test Designed by:**michael

**Test Designed date:**22/12/2020

**Module Name:** get user name **Test Executed by:** michael

**Test Execution date:** 22/12/2020

**Test Title:** Function get a user name and check availability  **Requirement ID:**

**Description:** Test if function get a user name

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Login as client | Press:1 | Should open menu of client(log in/register) | open menu of client(log in/register) | pass |  |
| 2 | Log in | Press:1 | Should open "get user name" and asked for a user name | Ask for user name | pass |  |
|  |  |  |  |  |  |  |

**Post-conditions: Through as a client I checked the correctness of the function, she asked and get a user name.**

**Test Case ID:**53

**Test Priority (Low/Medium/High): High** **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Add package to DB  **Test Executed by:** Omer Alali

**Test Execution date:** **:** 21/ 12/20

**Test Title:** Create a random serial number  **Requirement ID:**

**Description:** Create a random serial number for package

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Run the function while the program is running |  | Random number | 19243 | Pass |  |
| 2 | Run the function while the program is running immediately again |  | Random number | 19295 | Pass |  |
| 3 | Run the function after rebooting the program |  | Random number | 19540 | Pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Project Name:**

**Test Case Template**

**Test Priority (Low/Medium/High): High** **Test Designed by:** Omer Alali

**Test Case ID:** 54 **Test Designed date:** 21/ 12/20

**Module Name:** Add package to DB  **Test Executed by:** Omer Alali

**Test Execution date:** **:** 21/ 12/20

**Test Title:** Create a new package  **Requirement ID:**

**Description** Create a new package by destination, price start date, end date, and number of passengers. Return Package struct

**Pre-conditions:** Enter all the parameters for the function; check the correct input before sending to the function

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Create a new package | "London", 450, 12/05/20, 17/05/20, 2 | New package updated with the details entered | New package updated with the details entered | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**55

**Test Priority (Low/Medium/High): Medium** **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Add package to DB  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Delay function  **Requirement ID:**

**Description**: Performs a slight pause, assists the function of creating a serial number to prevent instances of creating the same serial number twice

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function |  | Pause for about a second on the output screen | Pause for about a second on the output screen | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**56

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** searchPackegeBySerialNumber function **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** search index package by serial number  **Requirement ID:**

**Description**: Receives a serial number of a package and returns its index in the database

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Enter a serial number for the function | 12345 | -1  (not exist) | -1 | Pass |  |
| 2 | Enter a serial number for the function | 19243 | 3  (index) | 3 | Pass |  |
| 3 | Enter a serial number for the function without  Packeges.dat file | 19243 | -2  (file opening fail) | -2 | Pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**57

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** PackegesfileSize function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Get the amount of packages in the database **Requirement ID:**

**Description**:

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function |  | 4 | 4 | Pass |  |
| 2 | Activating the function with 0 packages in file |  | 0 | 0 | Pass |  |
| 3 | Activating the function without  Packeges.dat file |  | -1 | -1 | Pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**58

**Test Priority (Low/Medium/High): Low**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** printPackegeDestination function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Receives a package and prints its destination **Requirement ID:**

**Description**:

**Pre-conditions:** Valid package

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function | Valid package | London | London | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**59

**Test Priority (Low/Medium/High): Low**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** printPackegeSerialNumber function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Receives a package and prints its serial number **Requirement ID:**

**Description**:

**Pre-conditions:** Valid package

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function | Valid package | 19243 | 19243 | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**60

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** getPackegeByIndex function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Receives a package by index **Requirement ID:**

**Description**: Returns a package according to its index, if the index is incorrect returns an empty package

**Pre-conditions:** Valid index

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Get package by valid index | 2 | Serial number: 24400  Destination: holand  Price: 299$  Arrival: 20/12/2020  leaving: 25/12/2020  Number of passengers: 3 | Serial number: 24400  Destination: holand  Price: 299$  Arrival: 20/12/2020  leaving: 25/12/2020  Number of passengers: 3 | Pass |  |
| 2 | Get package by not valid index | 10 | Serial number: 0  Destination: None  Price: 0$  Arrival: 0/0/0  leaving: 0/0/0  Number of passengers: 0 | Serial number: 0  Destination: None  Price: 0$  Arrival: 0/0/0  leaving: 0/0/0  Number of passengers: 0 | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**61

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** getAllPackeges function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Returns all packages in the database **Requirement ID:**

**Description**: Returns a pointer to a dynamic array with all the packages in the database.

**Pre-conditions**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function |  | Get all packages in file | Get all packages in file | Pass |  |
| 2 | Activating the function with 0 packeges in DB |  | Null | A pointer initializes to a junk entry | Fail | It is required to add conditions before entering copy |
| 3 | Activating the function without Packeges.dat file |  | Null | The program crashes | Fail | It is required to add conditions before entering copy |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**62

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** UpdatePackegeStatus function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title**: Package status update **Requirement ID:**

**Description**: Update Packege Status by serial number and status: 0 = none, 1 = In procces, 2 = Paid up 3 = Canceled 4 = "Approved" .if the update fail return -1

**Pre-conditions:** Valid status and serial number

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Status update to none | 0, 19243 | Status = None | Status has not changed | Fail | Case should be added to case 0 |
| 2 | Status update to in process | 1, 19243 | Status = in process | Status = in process | Pass |  |
| 3 | Status update to paid up | 2, 19243 | Status = paid up | Status = paid up | Pass |  |
| 4 | Status update to Canceled | 3, 19243 | Status = Canceled | Status = Canceled | Pass |  |
| 5 | Status update to Approved | 4, 19243 | Status = Approved | Status = Approved | Pass |  |
| 6 | Update the status of a package with a valid status and an invalid serial number | 1, 20200 | Return – 1 | Return – 1 | Pass |  |
| 7 | Update the status of a package with invalid status and a valid serial number | 8 , 19243 | Return – 1 | Return 1; | Fail | Need to add a return value of -1 to the default case |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**63

**Test Priority (Low/Medium/High): Low**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** printDate function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** print Date struct **Requirement ID:**

**Description**:

**Pre-conditions:** Valid Date struct

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function | Valid Date | 12/04/93 | 12/04/93 | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**64

**Test Priority (Low/Medium/High): Medium** **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** printPackege function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** print Package struct **Requirement ID:**

**Description**:

**Pre-conditions:** Valid Package struct

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function | Valid Date | Serial number: 24400  Destination: holand  Price: 299$  Arrival: 20/12/2020  leaving: 25/12/2020  Number of passengers: 3 | Serial number: 24400  Destination: holand  Price: 299$  Arrival: 20/12/2020  leaving: 25/12/2020  Number of passengers: 3 | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**65

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Update package in DB **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Update package Price  **Requirement ID:**

**Description**: Update packege price by serial number and new price. If the update success return true else return false

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Price update with invalid package number | 450, 2020 | 0  (not exist) | 0 | Pass |  |
| 2 | Price update | 1111, 24400 | Price: 1111 | Price: 1111 | Pass |  |

**Post-conditions:**

**Test Case ID:**66

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Update package in DB **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Update package Arrival Date  **Requirement ID:**

**Description**: Update packege Arrival Date by serial number and new price. If the update success return true else return false

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Arrival date update with invalid package number | Date (12/05/20), 2020 | 0  (not exist) | 0 | Pass |  |
| 2 | Arrival date update | Date (12/05/20), 24400 | Arrival: 12/05/20 | Arrival: 12/05/20 | Pass |  |

**Post-conditions:**

**Test Case ID:**67

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Update package in DB **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Update package leaving date  **Requirement ID:**

**Description**: Update packege leaving Date by serial number and new price. If the update success return true else return false

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Leaving date update with invalid package number | Date (12/05/20), 2020 | 0  (not exist) | 0 | Pass |  |
| 2 | Leaving date update | Date (12/05/20), 24400 | leaving: 12/05/20 | Arrival: 12/05/20 | Pass |  |

**Post-conditions:**

**Test Case ID:**68

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** add package amount **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Package duplication  **Requirement ID:**

**Description**: Receives a package and creates a new package with its details. (Does not copy: status, serial number, and name of buyer - get a default value)

**Pre-conditions:** valid package

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Package duplication | Package(  Serial number: 24315  Destination: usa  Price: 599$  Arrival: 16/12/2020  leaving: 18/12/2020  Number of passengers: 2  Status: Paid up  Name of buyer: Moshe) | Package(  Serial number: random  Destination: usa  Price: 599$  Arrival: 16/12/2020  leaving: 18/12/2020  Number of passengers: 2  Status: None  Name of buyer: None) | Package(  Serial number: 33580  Destination: usa  Price: 599$  Arrival: 16/12/2020  leaving: 18/12/2020  Number of passengers: 2  Status: None  Name of buyer: None) | Pass |  |

**Post-conditions:**

**Test Case ID:**69

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Order management  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Print all packages waiting for approval  **Requirement ID:**

**Description**: Screen printing of all packages waiting for agent approval, if no packages the function returns -1

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Print all packages waiting for approval when in Database there are no packages waiting for approval |  | Return – 1 | Return – 1 | Pass |  |
| 2 | Print all packages waiting for approval Database has packages with all the different statuses |  | Print 1 package that has status Paid up | Print 1 package that has status Paid up | Pass |  |

**Post-conditions:**

**Test Case ID:**70

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Update package in DB **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Update package buyer name.  **Requirement ID:**

**Description**: Update packege buyer name. by serial number and new name. If the update success return true else return false

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Buyer name update with invalid package number | "omer", 2020 | 0  (not exist) | 0 | Pass |  |
| 2 | Buyer name update | "omer", 24400 | Buyer name: omer | Buyer name: omer | Pass |  |

**Post-conditions:**

**Test Case ID:**71

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Update package in DB **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Delete package from DB  **Requirement ID:**

**Description**: Deleting a packet from the database by a serial number, if the packet does not exist should return 0

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Delete a package from the database | 24400 | Deleting the package from the database | The package has been deleted | Pass |  |
| 2 | Delete a package that does not exist from the database | 24400 | Return 0 | Return 0 | Pass |  |
|  |  |  |  |  |  |  |
| 3 | Delete a package from the database when Packeges.dat file not exist | 19475 | Return 0 | The program crashes | Fail | Adding a condition that if the file is not opened returns 0 |

**Post-conditions:**

**Test Case ID:**72

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Order management  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Print all packages that Canceled  **Requirement ID:**

**Description**: Screen printing of all packages that Canceled, if no packages the function returns -, if the opening failed return -2

**Pre-conditions:**

**Dependencies:** Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Print all packages that Canceled when Packeges.dat file not exist |  | Return – 2 | Return – 2 |  |  |
| 2 | Print all packages that Canceled when in Database there are no packages that Canceled |  | Return – 1 | Return – 1 | Pass |  |
| 3 | Print all packages that Canceled . Database has packages with all the different statuses |  | Print 1 package that has status Canceled | Print 1 package that has status Canceled | Pass |  |

**Post-conditions:**

**Test Case ID:**73

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** User register **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Create a new client struct  **Requirement ID:**

**Description**: Create new client struct by user name, password, mail address.

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Create a new client struct | "omer", "1234", "omer@gmail.com" | Client (omer, 1234, omer@gmail.com) | Client (omer, 1234, omer@gmail.com) | Pass |  |

**Post-conditions:**

**Test Case ID:**74

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** User registers **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Add a client to the database  **Requirement ID:**

**Description**: Adding a new client to the database by client structure, if no client file exists the software will create a new file.

**Pre-conditions:**

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Add a client when a client file does not exist | Client( "omer", "1234", [omer@gmail.com](mailto:omer@gmail.com)) | A customer file is created and the customer is added to the file | A customer file is created and the customer is added to the file | Pass |  |
| 2 | Add a new client to an existing file | Client( "omer", "1234", [omer@gmail.com](mailto:omer@gmail.com)) | The client is added to the end of the existing file | The client is added to the end of the existing file | Pass |  |

**Post-conditions:**

**Test Case ID:**75

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** searchClientByUserName function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** search index client by user name  **Requirement ID:**

**Description**: Receives a user name of a client and returns its index in the database

**Pre-conditions:**

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Entering a non-existent customer name | "Yossi" | -1  (not exist) | -1 | Pass |  |
| 2 | Enter a serial number for the function | "Omer" | 2  (index) | 2 | Pass |  |
| 3 | Enter a user name for the function without  Clients.dat file | "Omer" | -2  (file opening fail) | -2 | Pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**76

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** ClientsfileSize function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Get the amount of clients in the database  **Requirement ID:**

**Description**:

**Pre-conditions:**

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function |  | 3 | 3 | Pass |  |
| 2 | Activating the function with 0 clients in file |  | 0 | 0 | Pass |  |
| 3 | Activating the function without  Clients.dat file |  | -1 | -1 | Pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**77

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** getAllPackeges function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Returns all s clients in the database **Requirement ID:**

**Description**: Returns a pointer to a dynamic array with all the clients in the database.

**Pre-conditions**

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function |  | Get all Clients file in file | Get all Clients in file | Pass |  |
| 2 | Activating the function with 0 clients in DB |  | Null | A pointer initializes to a junk entry | Fail | It is required to add conditions before entering copy |
| 3 | Activating the function without Clients.dat file |  | Null | Null | Pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**78

**Test Priority (Low/Medium/High): High** **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** getCllientByIndex function  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Receives a client by index **Requirement ID:**

**Description**: Returns a client according to its index, if the index is incorrect returns an empty client

**Pre-conditions:** Valid index

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Get client by valid index | 0 | Client( "omer", "1234", [omer@gmail.com](mailto:omer@gmail.com)) | Client( "omer", "1234", [omer@gmail.com](mailto:omer@gmail.com)) | Pass |  |
| 2 | Get package by not valid index | 10 | Client( None, None, None) | Client( None, None, None) | Pass |  |
|  |  |  |  |  |  |  |

**Post-conditions:**

**Test Case ID:**79

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Order package **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Update Clients Order Details  **Requirement ID:**

**Description**: get user name, full name (name, last name) and phone and update the details in this client

**Pre-conditions:**

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Order Details update with invalid user name | "omer", "omer", "alali", 0549382115 | Return 0  (not exist) | Return 0 | Pass |  |
| 2 | Order Details update | "omer", "omer", "alali", 0549382115 | Name: omer  Last name: alali  Phone: 0549382115 | Name: omer  Last name: alali  Phone: 0549382115 | Pass |  |

**Post-conditions:**

**Test Case ID:**80

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Order package **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Update Client Payment Details  **Requirement ID:**

**Description**: get user name, and credit card number and id update the details in this client

**Pre-conditions:**

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Order Details update with invalid user name | "Yossi", 1234123412341234, 203345968 | Return 0  (not exist) | Return 0 | Pass |  |
| 2 | Order Details update | "omer", 1234123412341234, 203345968 | ID: 203345968  Credit card: 1234123412341234 | ID: 203345968  Credit card: 1234123412341234 | Pass |  |

**Post-conditions:**

**Test Case ID:**81

**Test Priority (Low/Medium/High): Low**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Send message (Client)  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** create today date  **Requirement ID:**

**Description**: The function should produce the current date of the day

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Activating the function |  | Date (21/1220) | Date (21/1220) | Pass |  |

**Post-conditions:**

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Case ID:** 82 **Test Designed date:** 21/ 12/20

**Module Name:** Send message (Client)  **Test Executed by:** Omer Alali

**Test Execution date:** **:** 21/ 12/20

**Test Title:** Create a new message struct  **Requirement ID:**

**Description** Create a new message struct,. by title and content.

he default status is Unread. the default date is Today's date.

**Pre-conditions:** Enter all the parameters for the function; check the correct input before sending to the function

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Create a new message | "Title", "Content" | Massage (Title, Content, Unread, 21/12/20) | Massage (Title, Content, Unread, 21/12/20) | Pass |  |

**Post-conditions:**

**Test Case ID:**83

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Designed date:** 21/ 12/20

**Module Name:** Send message (Agent) **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Update agent response to customer message **Requirement ID:**

**Description**: The function receives the customer's name, message number (index) and agent's answer and updates the customer's answer

**Pre-conditions:** valid index;

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Update reply to message | "omer", "response", 1 | Response: " response"  Status: Answer | Response: " response"  Status: Answer | Pass |  |
| 2 | Update a reply to a message when a customer file does not exist | "omer", "response", 1 | Return 0 | Return 0 | Pass |  |
| 3 | Update an answer with an incorrect username | "tmer", "response", 1 | Return 0 | Return 0 | Pass |  |

**Post-conditions:**

**Test Priority (Low/Medium/High): High**  **Test Designed by:** Omer Alali

**Test Case ID:** 84 **Test Designed date:** 21/ 12/20

**Module Name:** Send message (Client)  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Add a new message by a customer  **Requirement ID:**

**Description**: Receive a username and message and update the message in the customer's message list

**Pre-conditions:**

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Update a message to a customer with an incorrect customer name | "Tmer", Message | Return 0 | Return 0 | Pass |  |
| 2 | Update a message to a customer | "omer", Message | The message was added to the client's message list | The message was added to the client's message list | Pass |  |

**Post-conditions:**

**Test Priority (Low/Medium/High): Medium** **Test Designed by:** Omer Alali

**Test Case ID:** 85 **Test Designed date:** 21/ 12/20

**Module Name:** Wish list (Client)   **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Add a package to a customer's wish list  **Requirement ID:**

**Description**: Receive a serial number of a package and username and update of the customer's wish list

**Pre-conditions:**

**Dependencies:** Clients.dat file, Packeges.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Update a package in the wish list | "test", 22440 | Update a package in the customer's wish list | Update a package in the customer's wish list | Pass |  |
| 2 | Updating a package in the customer's wish list with the wrong serial number | "test", 89542 | Return 0 | Package not added but no return value is 1 | Fail | Adding a condition that if a package is not found a return value is a lie |
| 3 | Updating a package in the customer's wish list with the wrong username | "omer", 22440 | Return 0 | Return 0 | Pass |  |

**Post-conditions:**

**Test Case ID:**86

**Test Priority (Low/Medium/High): Low**  **Test Designed by:** Omer Alali

**Test Designed date:** 22/ 12/20

**Module Name:** Message box  **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Print a message **Requirement ID:**

**Description**: Receives a message (struct) and prints it. If there is a response to the ad also prints the response

**Pre-conditions:**

**Dependencies:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Print a message without response | Message ("title", "Content") | \*\*\*title\*\*\*  22/12/2020  Content | \*\*\*title\*\*\*  22/12/2020  Content | Pass |  |
| 2 | Print a message with a response | Message ("title", "Content") | \*\*\*title\*\*\*  22/12/2020  Content  "Answer:"  Ans.. | \*\*\*title\*\*\*  22/12/2020  Content  "Answer:"  Ans.. | Pass |  |

**Post-conditions:**

**Test Priority (Low/Medium/High): High** **Test Designed by:** Omer Alali

**Test Case ID:** 87 **Test Designed date:** 24/ 12/20

**Module Name:** Massages box (Agent)  **Test Executed by:** Omer Alali

**Test Execution date:** 24/ 12/20

**Test Title:** Print all unread messages  **Requirement ID:**

**Description**: Check the screen print function of all unread messages. If a file does not exist, the function will return -1, if there are no messages the function should return 0. Check out all the different options

**Pre-conditions:**

**Dependencies:** Clients.dat file

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Print all unread messages when the file does not exist |  | Return -1 | Return -1 | Pass |  |
| 2 | Print all unread messages when there are no unread messages |  | Return 0 | Return 0 | Pass |  |
| 3 | Print all unread messages when there are two unread messages |  | Print 2 unread message | Print 2 unread message | Pass |  |

**Post-conditions:**

**Test Priority (Low/Medium/High): Medium** **Test Designed by:** Omer Alali

**Test Case ID:** 88 **Test Designed date:** 27/ 12/20

**Module Name:** Agent login **Test Executed by:** Omer Alali

**Test Execution date:** 21/ 12/20

**Test Title:** Connecting an agent to the system  **Requirement ID:**

**Description**: Checking an agent's login to the system by a valid username and password. And checking for invalid usernames or passwords as well as a malfunction in the agents file

**Pre-conditions:**

**Dependencies:** Agent.txt

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
|  |  |  |  |  |  |  |
| 1 | Login with valid username and password | "benny", "123" | Return 0 | Return 0 | Pass |  |
| 2 | Login when the file does not exist | "benny", "123" | Return -3 | Return -3 | Pass |  |
| 3 | Login with valid username and password incorrect | "benny", "1234" | Return -1 | Return -1 | Pass |  |
| 4 | Login with incorrect username | "omer", "123" | Return -2 | Return -2 | Pass |  |

**Post-conditions:**