M.U.M BLACK BOX TESTING DRAFT

**TESTING PLANS**

Method: addUser()

Description of Method: This method registers a new customer into the system and then allows the user to interact with customer representatives as well as launch tickets.

Criterion:

* User must not have a pre-existing account
* User has an valid ID
* User password must me 8 or more characters

Valid Class:

* Customer ID has never been used to create an account

Invalid Class:

* Customer ID is already in use by an existing account or not registered

Figure1: Table Showing Black Box Test for addUser() Method

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid ID | User enters his previously unused ID to register | Notification to user that their account is created & prompted to choose a user name and password |
| 2 | Invalid  ID | User enters an ID used by another customer to register | Notification to user that ID entered is already in use & is prompted to enter a another ID |
| 3 | Invalid ID | User enters an ID unregistered by the system | Notification to the user that ID is not registered & is prompted to enter a valid ID |

**TESTING PLANS**

Method: removeTicket()

Description of Method: This method allows an agent to remove tickets deemed invalid from the system as well as flag potential resource wasting users.

Criterion:

* Issue already identified from the same user
* Incomplete ticket information

Valid Class:

* Spam

Invalid Class:

* User has raised serious & original issue

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Input Type | Input | Expected output |
| 1 | Valid  Spam ticket | Agent repeatedly receives ticket already documented information | Agent removes ticket |
| 2 | Valid Incomplete ticket | Agent receives incomplete ticket information from user | Agent removes ticket |
| 3 | Invalid Complete & original ticket | Agent receives ticket that highlights an undocumented issue | Agent forwards ticket to be handled by developers |

Figure2: Table Showing Black Box Test for removeTicket() Method

**TESTING PLANS**

Method: addToList()

Description: This method confirms the package the customer wishes to purchase the charges the customer’s account with the amount of credits equivalent to the cost of the package selected.

Criterion:

* Customer must be logged in
* Customer has selected a package to purchase
* Customer should have enough credits on their account to complete the purchase
* Device must be connected to the Internet

Valid Class:

* Package selected is less than or equal to the amount of credits on that user’s account

Invalid Class:

* Cost of package selected exceeds amount of credits user has on their account

Boundary: Amount of credits customer has on their account

Figure3: Table Showing Black Box Test for addToList() Method

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid PackageCost<AmtOfCredits | Cost of package is less than the amount of credits the customer has on their account | Notification to customer that their package has been successfully purchased and their account has been reduced by the cost of the package |
| 2 | Invalid PackageCost>AmtOfCredits | Cost of package exceeds the amount of credits the customer has on their account | Notification to customer that they do not have sufficient funds to complete the transaction and is prompted to purchase more credits |
| 3 | Invalid PackageCost=0 | No package selected | Notification to customer that no package is selected |

**TESTING PLANS**

Method: authenticate()

Description of Method: This method allows the user to authenticate themselves by entering a valid ID and password. The system will then check these credentials for validity.

Criterion:

* User must have a pre-existing account
* User has an valid ID
* User has a valid password

Valid Class:

* User enters a valid ID and password

Invalid Class:

* User does not enter a valid ID

Figure4: Table Showing Black Box Test for authenticate() Method

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid Verified ID & password | User enters an ID and password that matches one saved in the system database | Notification to customer that their login has been successful and allowed to move on to the next page |
| 2 | Invalid  Verified ID & inccorect password | User enters a valid ID but wrong password | Notification to customer that their password is incorrect and to recheck their spelling |
| 3 | Invalid Unverified ID | User enters an invalid ID | Notification to customer that no such ID exists |

**TESTING PLANS**

Method: selectPackage()

Description: This method presents the user with a list of upgrade packages for him to choose from then calls the addToList() method once a choice is made

Criterion:

* Customer must be logged in
* Customer should have enough credits on their account to complete the purchase

Valid Class:

* User selects package with a higher connection speed than their current package

Invalid Class:

* User selects package with a lower connection speed than their current package

Boundary: Amount of credits customer has on their account

Figure5: Table Showing Black Box Test for selectPackage() Method

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid Package with higher connection speed than current package | User selects a package with higher connection speed that current package | Notification to customer that their package has been successfully chosen |
| 2 | Invalid PackageCost>AmtOfCredits | Cost of package exceeds the amount of credits the customer has on their account | Notification to customer that they do not have sufficient funds to complete the transaction and is prompted to purchase more credits |
| 3 | Invalid Package with lower connection speed than current package | User selects a package with lower connection speed that current package | Notification to customer that a package with lower connection speed that current package and asked to confirm whether they made the right decision |