## Manage Product – Load

User loads an existing product’s information from the database.

|  |  |  |
| --- | --- | --- |
| **Client** | **System** | **Server** |
| Click the “Manage Product” button | Display “Manage Product” Screen |  |
| Fill in the “Product ID” field and click the “Load Product” button | Display the loaded product information in the corresponding fields | The client sends the server a ProductModel with the product id populated as data in a MessageModel with the GET\_PRODUCT code and the server sends back the corresponding product from the database. If the product does not exist, this error message will be displayed: |

## Manage Customer – Load

User loads an existing customer’s information from the database.

|  |  |  |
| --- | --- | --- |
| **Client** | **System** | **Server** |
| Click the “Manage Customer” button | Display “Manage Customer” Screen |  |
| Fill in the “Customer ID” field and click the “Load Customer” button | Display the loaded customer information in the corresponding fields | The client sends the server a CustomerModel with the customer id populated as data in a MessageModel with the GET\_CUSTOMER code and the server sends back the corresponding customer from the database. If the customer does not exist, this error message will be displayed: |

## Manage Purchase – Load

User loads an existing purchase’s information from the database.

|  |  |  |
| --- | --- | --- |
| **Client** | **System** | **Server** |
| Click the “Manage Purchase” button | Display “Manage Purchase” Screen |  |
| Fill in the “Purchase ID” field and click the “Load Purchase” button | Display the loaded purchase information in the corresponding fields and populate the label information. | The client sends the server a PurchaseModel with the purchase id populated as data in a MessageModel with the GET\_PURCHASE code and the server sends back the corresponding purchase from the database. If the purchase does not exist, this error message will be displayed: |

## Manage Product – Save

User saves a product to the database.

|  |  |  |
| --- | --- | --- |
| **Client** | **System** | **Server** |
| Click the “Manage Product” button | Display “Manage Product” Screen |  |
| Fill in the all of the product information and click the “Save Product” button | A message will be displayed showing the action was successful | The client sends the server the populated ProductModel as data in a MessageModel with the PUT\_PRODUCT code through Gson. The server checks to see if the product id matches a preexisting product id. If it does, then the old product is deleted. The product is then inserted into the database and the server sends back confirmation if the insertion was a success. |
| Click the “OK” button | Display the Home Screen |  |

## Manage Customer – Save

User saves a customer to the database.

|  |  |  |
| --- | --- | --- |
| **Client** | **System** | **Server** |
| Click the “Manage Customer” button | Display “Manage Customer” Screen |  |
| Fill in the all of the customer information and click the “Save Customer” button | A message will be displayed showing the action was successful | The client sends the server the populated CustomerModel as data in a MessageModel with the PUT\_CUSTOMER code through Gson. The server checks to see if the customer id matches a preexisting customer id. If it does, then the old customer is deleted. The customer is then inserted into the database and the server sends back confirmation if the insertion was a success. |
| Click the “OK” button | Display the Home Screen |  |

## Manage Purchase – Save

User saves a purchase to the database.

|  |  |  |
| --- | --- | --- |
| **Client** | **System** | **Server** |
| Click the “Manage Purchase” button | Display “Manage Purchase” Screen |  |
| Fill in the “Product ID” field | The “Product Name” and “Product Price” labels update | The client sends the server the product id and the server sends back the corresponding product from the database. If the product does not exist, the label will be filled accordingly: |
| Fill in the “Customer ID” field | The “Customer Name” label updates | The client sends the server the customer id and the server sends back the corresponding customer from the database. If the product does not exist, the label will be filled accordingly: |
| Fill in the “Quantity” field | The “Cost”, “Tax”, and “Total Cost” labels update    If there are not enough items in stock, a message displays:    If the quantity is invalid: |  |
| Fill in the “Purchase ID” field and click the “Save Purchase” button | Display the Home Screen | The client sends the server the populated PurchaseModel as data in a MessageModel with the PUT\_PURCHASE code through Gson. The server checks to see if the purchase id matches a preexisting purchase id. If it does, then the old purchase is deleted. The purchase is then inserted into the database and the server sends back confirmation if the insertion was a success. |