Pseudocode

ADMIN:

#Same for Employee(CRUD)

Procedure AddInventory(item)

Initialize inventory\_list as list of items

Add item to inventory\_list

Set message = "Item added successfully."

ShowMessage(message, "Add Inventory")

End Procedure

Procedure DeleteInventory(item\_id)

Initialize inventory\_list as list of items

For each item in inventory\_list

If item.id equals item\_id Then

Remove item from inventory\_list

Set message = "Item deleted successfully."

ShowMessage(message, "Delete Inventory")

End Procedure

Procedure UpdateInventory(item\_id, updated\_item)

Initialize inventory\_list as list of items

For each item in inventory\_list

If item.id equals item\_id Then

Replace item with updated\_item in inventory\_list

Set message = "Item updated successfully."

ShowMessage(message, "Update Inventory")

End Procedure

Procedure ViewInventory()

Initialize inventory\_list as list of items

For each item in inventory\_list

Display item details

Next

Return inventory\_list

End Procedure

Procedure UpdatePassword(user\_id, old\_password, new\_password)

Initialize users\_list as list of users with their IDs, passwords, and roles

For each user in users\_list

If user.id equals user\_id AND user.password equals old\_password Then

Set user.password = new\_password

Set message = "Password updated successfully."

ShowMessage(message, "Update Password")

Return True

End If

Next

Set message = "User ID or password is incorrect."

ShowMessage(message, "Update Password")

Return False

End Procedure

Procedure Login(user\_id, password)

Initialize users\_list as list of users with their IDs and passwords

For each user in users\_list

If user.id equals user\_id AND user.password equals password Then

Set loggedInTitle = "Login"

Set message = "You have successfully logged in as " + NameofUser

ShowMessage(message, loggedInTitle)

Return True

End If

Next

Set loggedInTitle = "Login Incomplete"

Set message = "Your login credentials are incorrect."

ShowMessage(message, loggedInTitle)

Return False

End Procedure

Customer:

Procedure PlaceOrder(user\_id, order)

Initialize order\_list as list of orders

Initialize users\_list as list of users

For each user in users\_list

If user.id equals user\_id Then

Add order to order\_list with user\_id

Set message = "Your order has been placed."

ShowMessage(message, "Place Order")

Return True

End If

Next

Set message = "Order could not be placed."

ShowMessage(message, "Place Order")

Return False

End Procedure

Procedure ViewOrderHistory(user\_id)

Initialize order\_list as list of orders

Initialize user\_orders as empty list

For each order in order\_list

If order.user\_id equals user\_id Then

Add order to user\_orders

End If

Next

If user\_orders is not empty Then

Display orders in user\_orders

Return user\_orders

Else

Set message = "No orders found for this user."

ShowMessage(message, "View Order History")

Return user\_orders

End If

End Procedure