Branch Class

```
public void branchOrderlist() {
		for (int i = 0; i < officeProductList.getSize(); i++) {		O(n)
			System.out.println(officeProductList.get(i).toString());	O(1)
		}
	}
	Time complexity branchOrderList:
		Linear time

public KWArrayList<BranchEmployee> getBranchEmployeeList() {
		return branchEmployeeList;		O(1)
	}
	Time complexity getBranchEmployeeList:
		Constant time


public void setBranchEmployeeList(KWArrayList<BranchEmployee> branchEmployeeList) {
		this.branchEmployeeList.cloneList(branchEmployeeList); O(n)
	}
	Time complexity setBranchEmployeeList:
		Linear time

public HybridList<OfficeProduct> getOfficeProductList() {
		return officeProductList; O(1)
	}
	Time complexity getOfficeProductList:
		Constant time



public void setOfficeProductList(HybridList<OfficeProduct> officeProductList) {
		this.officeProductList = officeProductList; O(1)
	}

	Time complexity setOfficeProductList:
		Constant time
```

Administrator Class

```
public void addBranch(Company company, Branch branch) {
		company.getBranchList().add(branch);		O(1)
	}

	Time Complexity:
		Constant time

public void addBranchIndex(Company company, Branch branch, int index) {
		company.getBranchList().add(index, branch); O(n)
	}
	Time Complexity:
		Linear time


public void removeBranch(Company company, Branch branch) {
		company.getBranchList().remove(branch); O(n)
	}
	Time Complexity:
		Linear time


public void addBranchEmployee(Branch branch, BranchEmployee branchEmployee) {
		branch.getBranchEmployeeList().add(branchEmployee); O(1)

	}

	Time Complexity:
		Constant time

public void removeBranchEmployee(Branch branch, BranchEmployee branchEmployee) {
		branch.getBranchEmployeeList().remove(branchEmployee); O(n)
	}

	Time Complexity:
		Linear time

public void addAdmin(Company company, Administrator newAdmin) {
		company.getUserList().add(newAdmin); O(1)
	}

	Time Complexity:
		Constant time
```

BranchEmployee Class:

```
public void addProductHead(Company company, OfficeProduct newProduct) {
		for (int i = 0; i < company.getBranchList().getSize(); i++) {	O(n)
			if (company.getBranchList().get(i).getName().equals(branchName))
O(1) {

	O(1)company.getBranchList().get(i).getOfficeProductList().addFirst(newProduct)

				}
		}
	}
Time Complexity:
	Linear Time


public void addProductIndex(Company company, OfficeProduct newProduct, int index) {
		for (int i = 0; i < company.getBranchList().getSize(); i++) { O(n)
			if (company.getBranchList().get(i).getName().equals(branchName))
{

	company.getBranchList().get(i).getOfficeProductList().add(index, newProduct);
O(n)

				}
		}

	}

Time Complexity:
	Quadratic Time  T(n)=O(n^2)

public void addProductTail(Company company, OfficeProduct newProduct) {
		for (int i = 0; i < company.getBranchList().getSize(); i++) { O(n)
			if (company.getBranchList().get(i).getName().equals(branchName))
{

	company.getBranchList().get(i).getOfficeProductList().addLast(newProduct);
O(1)

				}
		}
	}
Time Complexity:
	Linear Time
```

```
public void removeProduct(Company company, OfficeProduct newProduct) {
		for (int i = 0; i < company.getBranchList().getSize(); i++) { O(n)
			if (company.getBranchList().get(i).getName().equals(branchName))
{

	company.getBranchList().get(i).getOfficeProductList().remove(newProduct); O(n)
			}
		}
	}
```

Time Complexity:

Quadratic Time T(n)=O(n^2)

```
public void removeProductIndex(Company company, int index) {
		for (int i = 0; i < company.getBranchList().getSize(); i++) { O(n)
			if (company.getBranchList().get(i).getName().equals(branchName))
{

	company.getBranchList().get(i).getOfficeProductList().remove(index); O(n)
			}
		}
	}
```

Time Complexity:

Quadratic Time T(n)=O(n^2)

Customer Class

```
public int generateRandomCustomerId() {
		int tempCustomerId;
		Random rd = new Random();
		tempCustomerId = 1000000 + rd.nextInt(9000000);
		return tempCustomerId;
	}
```

Time Complexity:
Constant Time

```
public boolean customerValidate(Company company, String name, String pass) {
		for (int i = 0; i < company.getUserList().getSize(); i++) {
			if
(company.getUserList().get(i).getUserStatus().equals("Customer")) {
				if (company.getUserList().get(i).getName().equals(name)
&&
company.getUserList().get(i).getName().equals(pass)) {
					return true;
				}
			}

		}
		return false;
	}
```

Time Complexity:
Linear Time

```
public void customerRegister(Company company) {
		company.getUserList().add(this);
	}
```

Time complexity:
Constant Time

```
public void customerProductList() {

		System.out.print("WELCOME TO SYSTEM " + getName() + " old ordes \n");
		if (orders.getSize() == 0) {
			System.out.println("Order is null");
		}
		for (int i = 0; i < orders.getSize(); i++) {
			System.out.println(orders.get(i).toString());
		}
	}
```

Time complexity:
Linear time

```
public void customerSell(Company company, OfficeProduct product, String branchName) {
		orders.addFirst(product);
		for (int i = 0; i < company.getBranchList().getSize(); i++) { O(n)
			if (company.getBranchList().get(i).getName().equals(branchName))
{

	company.getBranchList().get(i).getBranchEmployeeList().get(0).removeProduct(co
mpany, product); O(n)
			}
		}

	}
```

Time Complexity:

Quadratic Time T(n)=O(n^2)

Company Class

```
public void branchList() {
		System.out.println("Branches List: ");
		int counter = 1;
		for (int i = 0; i < branchList.getSize(); i++) {
			System.out.println(counter + " : " +
branchList.get(i).getName());
			counter++;
		}
	}
```

Time complexity:

Linear Time

```
public void userList() {
		System.out.println("User List: ");
		int counter = 1;
for (int i = 0; i < userList.getSize(); i++) {
			System.out.println(counter + " - " + "Name: " +
userList.get(i).getName() + ", Surname: "
				+ userList.get(i).getSurname() + ", User status: " +
userList.get(i).getUserStatus());
}
for (int i = 0; i < branchList.getSize(); i++) {
			for (int j = 0; j <
branchList.get(i).getBranchEmployeeList().getSize(); j++) {
				System.out
					.println(counter + " - " + "Name: " +
branchList.get(i).getBranchEmployeeList().get(j).getName()
						+ ", Surname: " +
branchList.get(i).getBranchEmployeeList().get(j).getSurname()
						+ ", User Status: " +
branchList.get(i).getBranchEmployeeList().get(j).getUserStatus()
						+ "Branch Name: " +
branchList.get(i).getName());
			}
```

Time complexity:

Linear Time