

Gebze Technical University Faculty of Engineering



Final Project Report

Group 14

Submission Date: June 14, 2022

Supervisors: Fatih Erdogan Sevilgen

Github Link: [Sales Control System](#)

SALES CONTROL SYSTEM

1. Group Members

- 200104004036 ERAY ÖZKAN
- 1901042621 EMRE GÜVEN
- 1901042605 BURAK KOCAUSTA
- 1901042659 EMİRKAN BURAK YILMAZ
- 1901042616 İREM ÇAĞIN YURTTÜRK
- 1901042668 ŞEVVAL ATEŞ
- 1801042630 MEHMET AVNİ ÇELİK
- 161044026 EMİNE ESRA KARAOSMANOĞLU

2. Problem Definition

In the free markets, companies grow and grow if they fill a gap in the market. Because of the increasing population and globalized world, world-wide brands open branches all over the world. Thus, it is getting more and more difficult to keep track of the system which includes employees,products,customers etc.

Not just for the most known companies, also small and medium-sized enterprises also known as sme, need more modern systems for their business to compete with industrial giants and stay alive in the market.

It is experienced that the markets which did not pay attention to modernization of the shopping systems had a lot of troubles with banks,taxes,customers,employees and many things related to lack of tracking. Expanding the capacity of any market and enhancing the service and product quality becomes more difficult without tracking systems.

The “Sales Control System” application makes available some essential properties,such as inventory and branch control, employee and ordering information for the companies. By this means, the system helps owners who want to grow in the market or who need to track their products with the least loss ratio. The system also helps companies by providing user friendly accessibility for its customers.

3. Users of the System

The Sales Control System consists of four users. These users are Administrators , branch manager, branch employees and customers.

3.1 Administrator

Administrators manage the whole system by adding or removing branches and their branch managers and customers to the system.

3.2 Branch Manager

Branch Managers manage their branch only by adding and removing branch employees to the system.

3.3 Branch Employees

Branch Employees can see product requests from branches and accept or reject them, and can add or remove products to the system.

3.4 Customers

- Customers can see/buy products and their prices and information.
- Suggest the closest branch that has gas stock for a specific product based on its location.

4. Requirements in Details

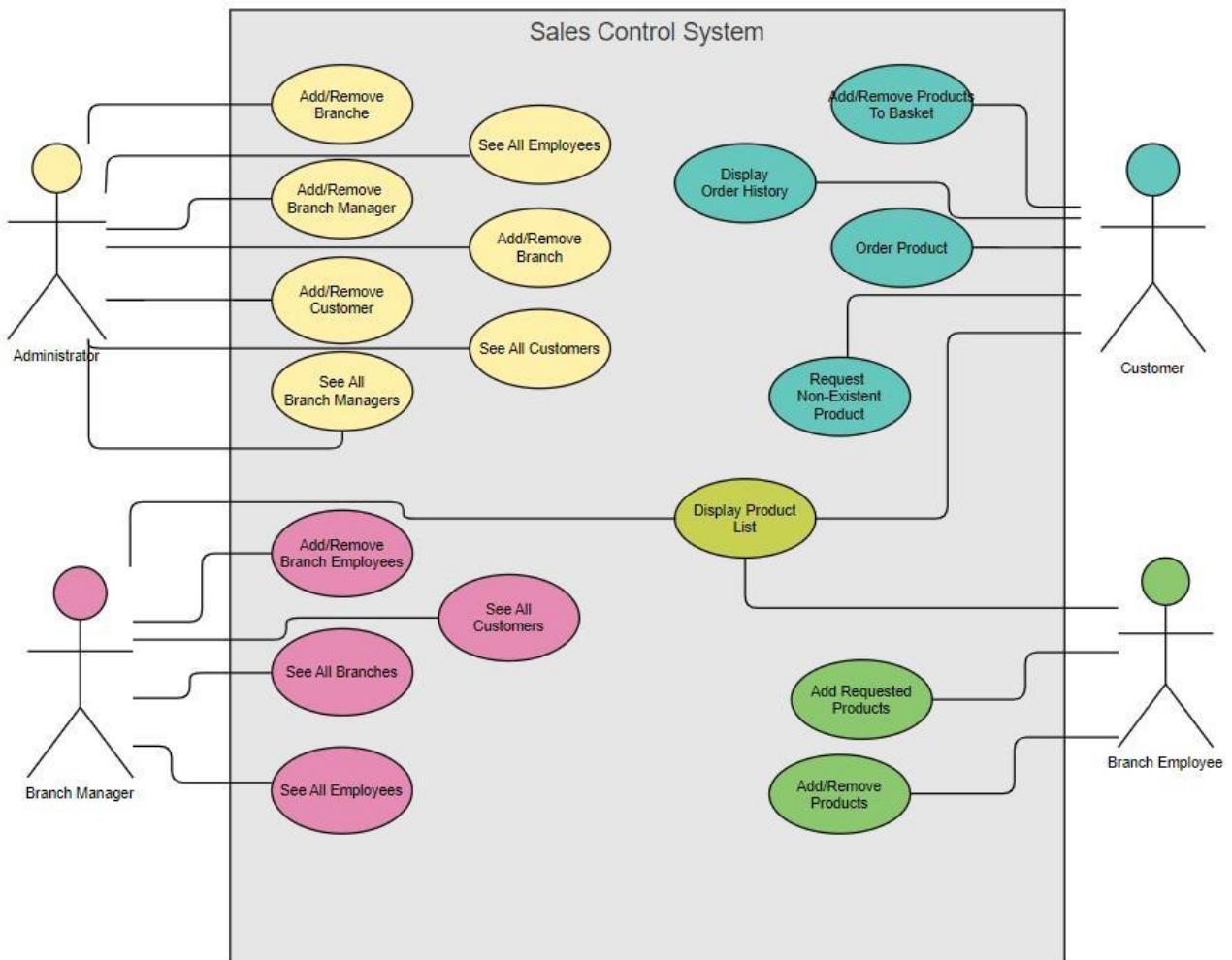
4.1 Functional Requirements

| Requirement ID | Description |
|----------------|---|
| FR_1 | System is provided for 4 different types of user; administrator, branch manager, branch employees and customers of a company. And the system shall let users to login to the system with their information. |
| FR_2 | System must have a company and its unique administrator. |
| FR_3 | System shall let the administrator add new branches to the company and remove the existing branches from the company. |
| FR_4 | System shall let the administrator set the managers of branches. |
| FR_5 | System shall let the administrator to display the information of all branches, all branch managers, all branch employees and all customers. |
| FR_6 | System shall provide a unique manager for a branch. |
| FR_7 | System shall let branch managers add new employees to their branches, and remove the working employees from their branches. |
| FR_8 | System shall let branch managers add new customers to their branches, and remove the existing customers from their branches. |
| FR_9 | System shall let the branch manager to display the information of employees and products of their branches ,and the Branch Sales Price. |
| FR_10 | System shall let branch employees to add new products to their branches and remove the existing products from their branches.Branch employee displays the products. |
| FR_11 | System shall let branch employees to add requested products,requested from customers, to their branches. |
| FR_12 | System shall let customers add products to the basket and remove products from the basket. Also customer can buy the basket. |
| FR_13 | System shall let customers request non existing products from their branches. |
| FR_14 | System shall let customers to display the products in the branch. |
| FR_15 | System shall let customers display their orders in historical order. |
| FR_16 | System shall let administrators approve and reject all of the waiting registrations , <u>approve and pass and reject waiting registrations individually</u> |
| FR_17 | System shall let administrator displays company sales price. |

4.2 Non-Functional Requirements

| Requirement ID | Requirement | Description |
|----------------|---------------|--|
| NFR_1 | Usability | System shall provide an interactive menu for users to perform their operations user friendly |
| NFR_2 | Security | Username and password is required to sign in to the system. |
| NFR_3 | Performance | The data in the system will be dynamic and this data will expand as users log in. Also, the used algorithms will improve the performance. |
| NFR_4 | Space | The data in the system will be dynamic, information of users and products are added as long as the system is working. Because of these, an expandable space is required. |
| NFR_5 | Operational | The system must be in communication with the users in order to perform the necessary operations. User data and interaction is one of the most fundamental requirements in this system. |
| NFR_6 | Environmental | The program is designed for the computer and will be written in a way to run on the computer terminal. |
| NFR_7 | Learnability | When users see the interface, they can understand and implement the main actions. |
| NFR_8 | Accounting | The system must have an administrator. Administrators can confirm employees, managers and customers to register in the system. |
| NFR_9 | Development | This system will be developed by many developers at the same time, and the development environment must be suitable for it. |
| NFR_10 | Manageability | When editing the code for customer, branch employee or branch manager, the rest of the part stays up and running. |

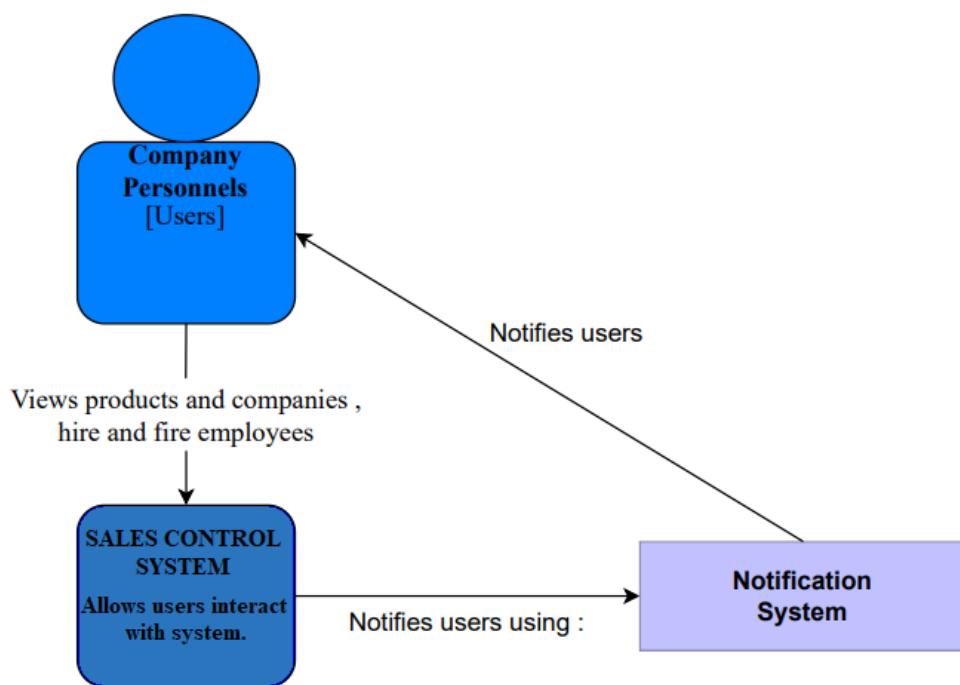
5. Use-Case Diagrams



6. The C4 Model of the System

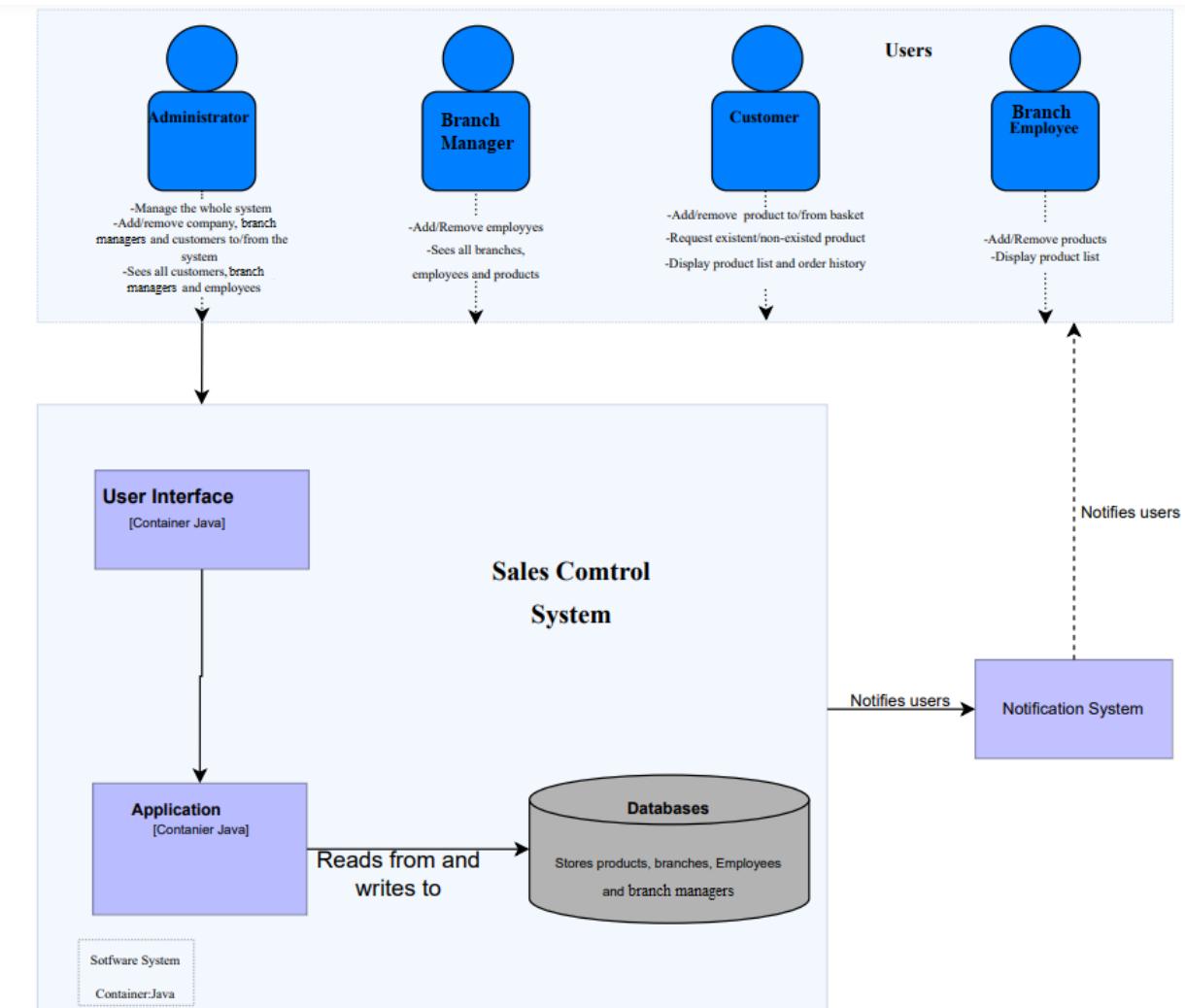
Level 1 : System Context Diagram

Projects system can be seen in a box in the center, surrounded by its users and the other system that it interacts with.



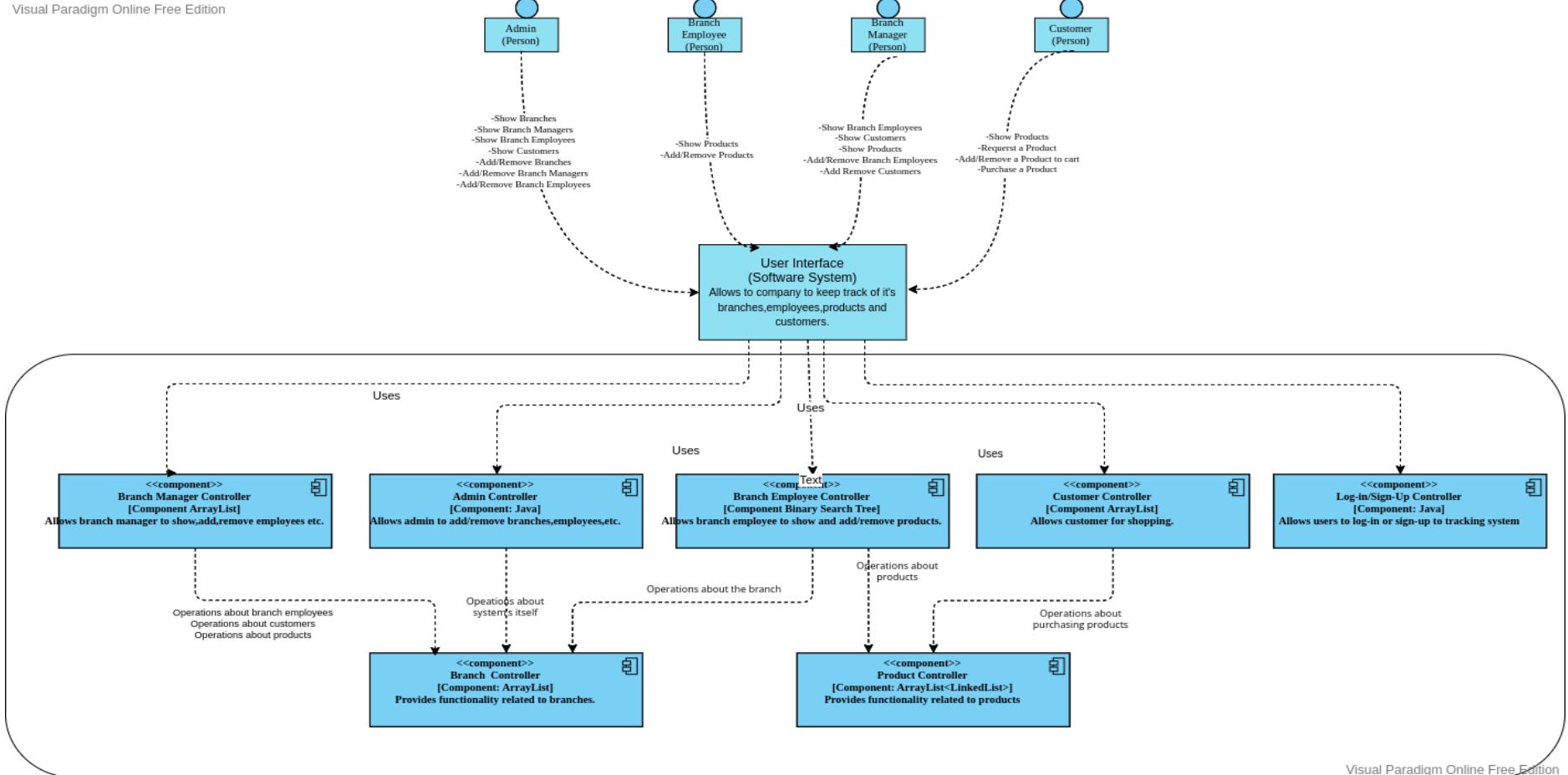
Level 2 : Container Diagram

Container diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it.



Level 3 : Component Diagram

A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.



Use of Data Structures:**1. List:**

List data structure is used to keep the items that don't have specific order. The ArrayList or LinkedList data structure is selected because of the frequent use of add/remove product operation for basket. Since it's easy to add/remove items in LinkedList compared to ArrayList, products are kept in an arraylist of linked lists. Other than this ArrayList is used for sorting operations in the system.

2. Stack:

Stack data structure is used to organize data according to LIFO (Last In, First Out). The customer order history is kept in stack to provide better system usability for customers.

3. Binary Search Tree (BST):

BST data structure is used to keep the users of a system such to provide an efficient searching basis on the name of the user. It is explained below in detail. Because in the final report, they are converted to a balanced tree.

4. Priority Queue:

Binary Heap data structure is used as Priority Queue to set priority between objects. That type of priority is needed to keep the requested products in a local branch. To not miss high profit orders the requested products are kept in max-heap basis on entry price of the product.

5. TreeSet(Navigable):

TreeSet is used for holding products which have the same product type. TreeSet is held in HashMap as a value. TreeSet is useful because products should be accessed in name order. With this structure; searching, insertion, removal becomes logarithmic time complexity due to the Red Black Tree. It is the underlying data structure of TreeSet. Navigable property is useful for Products, because products might be traversed in reverse order, or floor(), ceiling() methods can be useful. Main reason is customers definitely want to view products in various ways. For example, customers might want to sort products in reverse order by name, or bigger than(some price, etc..), similarly fewer than requests can be made. Navigable property can handle all these operations.

6. Hash Map:

Hash Map is the most used data structure in this report. Firstly, it is most useful in holding the (username, password) as pair in the system, and (username, user) pair also in the system. HashMap has average constant time access, insertion, and removal. For holding pairs like these, arrays or other data structures can cause pain. Using username as a key for password is useful for checking login. Using username as a key again for users is useful for accessing User class in constant time when it is needed. With this access, this class's operations can be used just using username as a key. HashMap is used in graph implementation also, because of it being a Dynamic Graph, Array is not enough. Removal and insertion to a HashMap based graph is much smarter than an array, because when a vertex is removed, all IDs must be changed or unnecessary space must be used. ID's can be nonconsecutive, therefore HashMap is much more useful than an array for this kind of implementation. When implementing the graph, LinkedHashMap is used because Iterator is used so much, while traversing the whole LinkedHashMap with iterator its time complexity is linear with number of elements. It is not like that in regular HashMap, in regular HashMap it is linear with capacity of the Hash Table, therefore using LinkedHashMap is better.

7. Skip List:

Customers are held in Skip List according to their name, in this way accessing, inserting, and removing customers takes logarithmic time. It is also concurrent which is useful for multiple access during sales. Probability of more than one customer accessing is high, therefore ConcurrentSkipList is used.

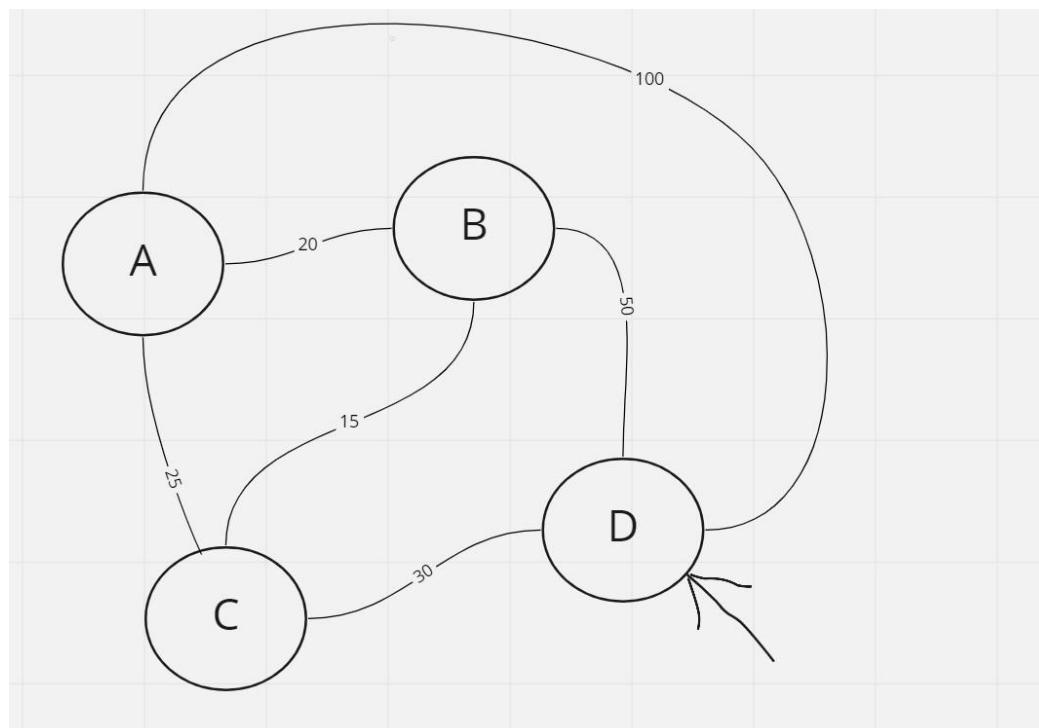
8. Balanced Binary Search Tree:

Red Black Tree is used for holding the employees according to their name. Red Black Tree provides an efficient search, insertion, and removal. All of these operations are done in logarithmic time complexity. It is preferred over other balanced trees because insertion, and removal is better for Red Black Trees compared to other balanced trees. Also Red Black Tree has better space complexity than AVL Tree for example. Because of these memory, and performance reasons, Red Black Tree is preferred. Also TreeSet is used for holding products because of the reason explained above. TreeSet also has an underlying data structure as Red Black Tree. TreeSet is better for products than this Red Black Tree, because of the advantages of Navigable property explained. For the employees, these flexibilities are not necessary. Using Simple implementation of Red Black Tree is enough. Only an iterator is needed, and it is implemented.

9. Graph:

Graph is used for holding branches. Main reason to hold branches on a graph is, every branch's distance between each is important for recommending a branch to a customer. While recommending a branch to a customer, the closest branch must be recommended. With graph implementation, using the Dijkstra Algorithm opportunity shows up. When Customer requests a suggestion of another branch with some condition(in our implementation it is nonexistence of a wanted item on a specific branch), the closest branch which provides the customer's needs. Graph is dynamic, because of the reason explained above. When a customer wants to buy a nonexistent item, the suggestion algorithm checks the closest branch which has that item, and recommends it to the customer. All connections must be inserted by the Administrator. Graph only maps(routes) the current company's branches, so it is an administrator's job. Branches don't have much insertion, removal, search operation, it is not expected that these operations occur frequently as other data like Products or Customers. Adding removing products, customers are much more frequent than adding, removing branches. So the disadvantages of the time complexity of removal operation don't matter as much as others. Still insertion of a branch is constant time, accessing with id's makes these operations faster. So holding them inside a graph for the reason of maps, routes is much better.

Suggesting Branch is simply works like this:

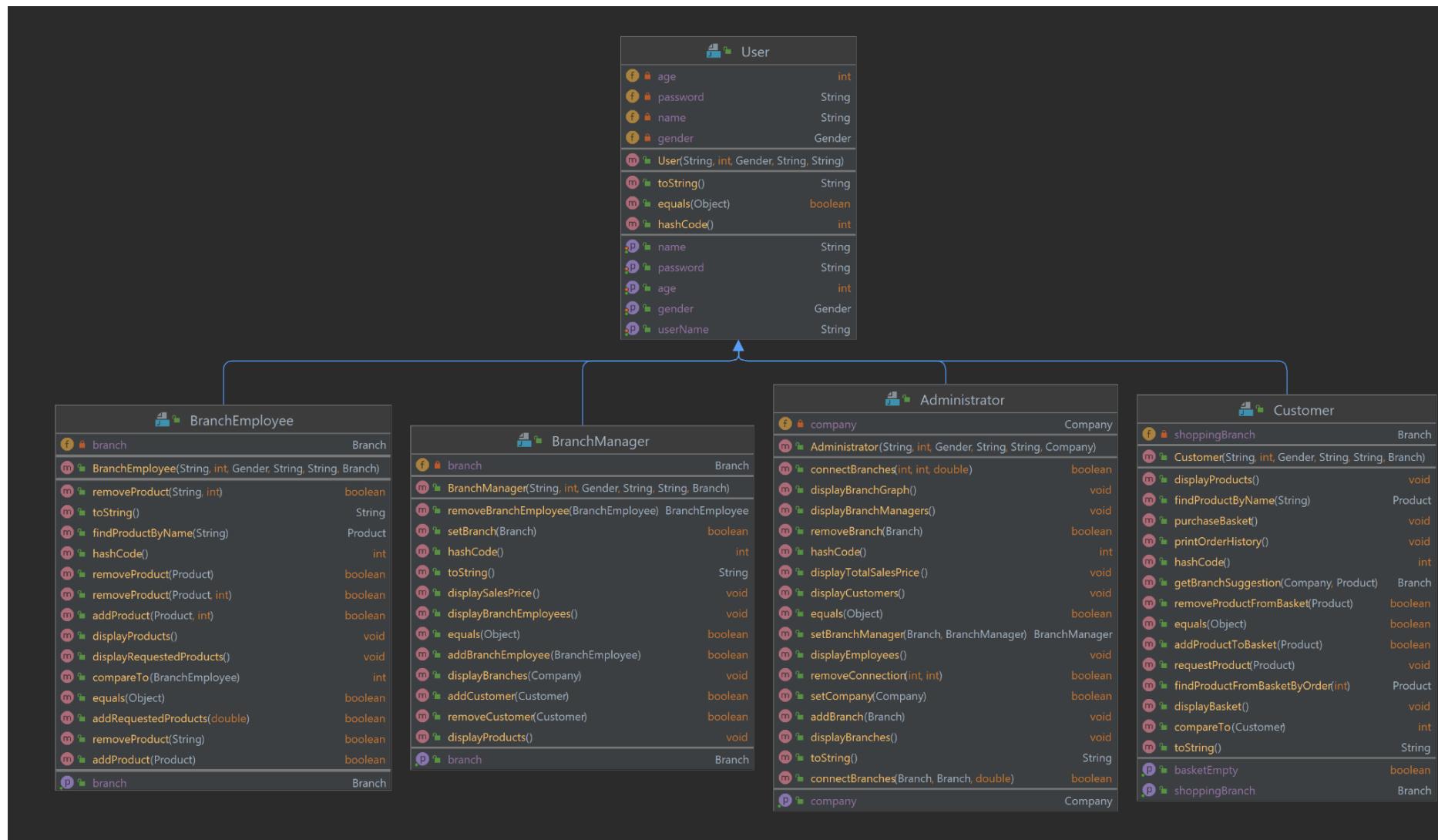


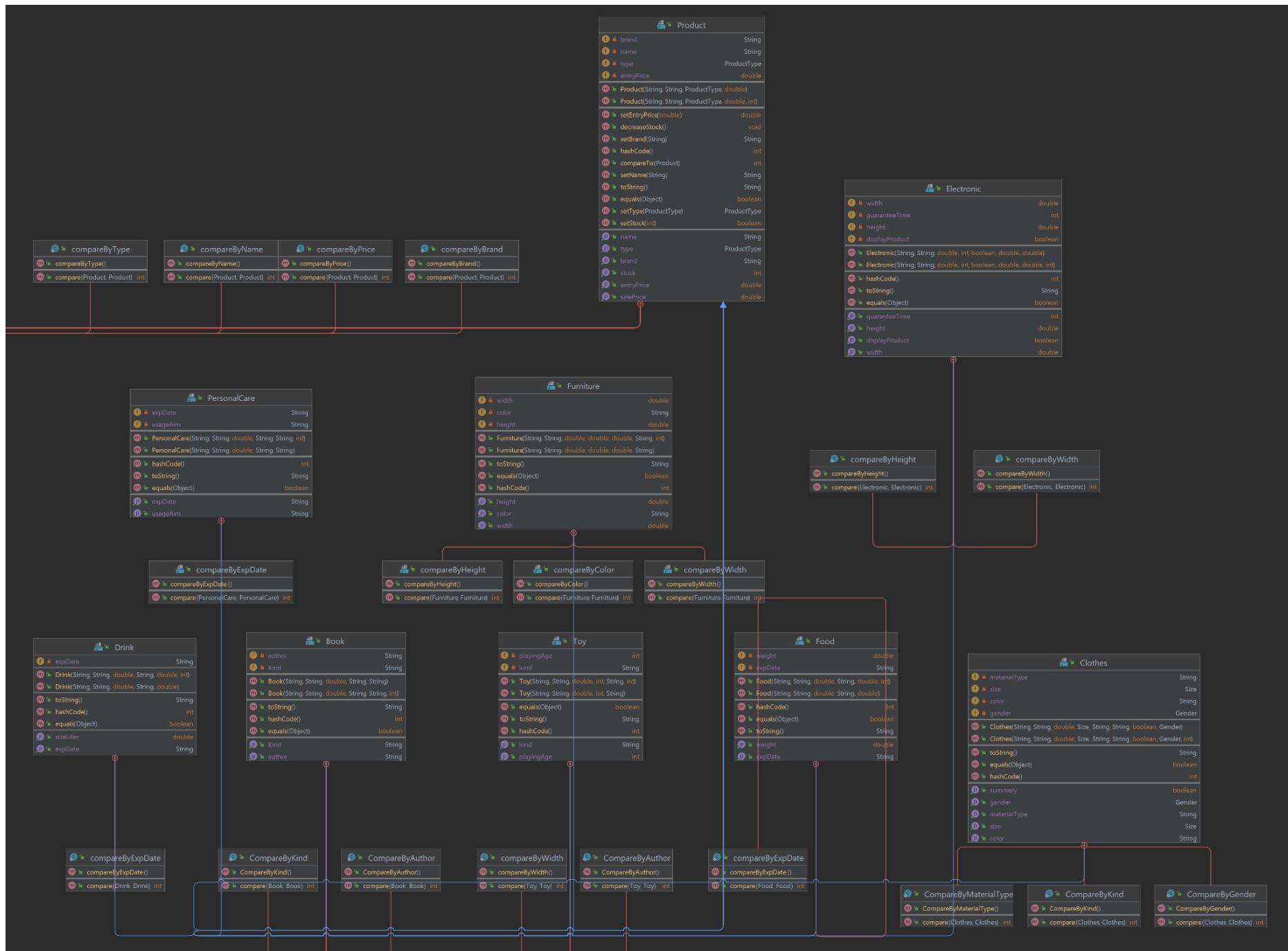
In this case, Customers current branch is Branch D, and this branch does not have the product that custom wants. Branch A, B have that product. After that, the System suggests the closest branch and Branch which has the desired product using the Dijkstra Algorithm. Therefore, Branch B is suggested.

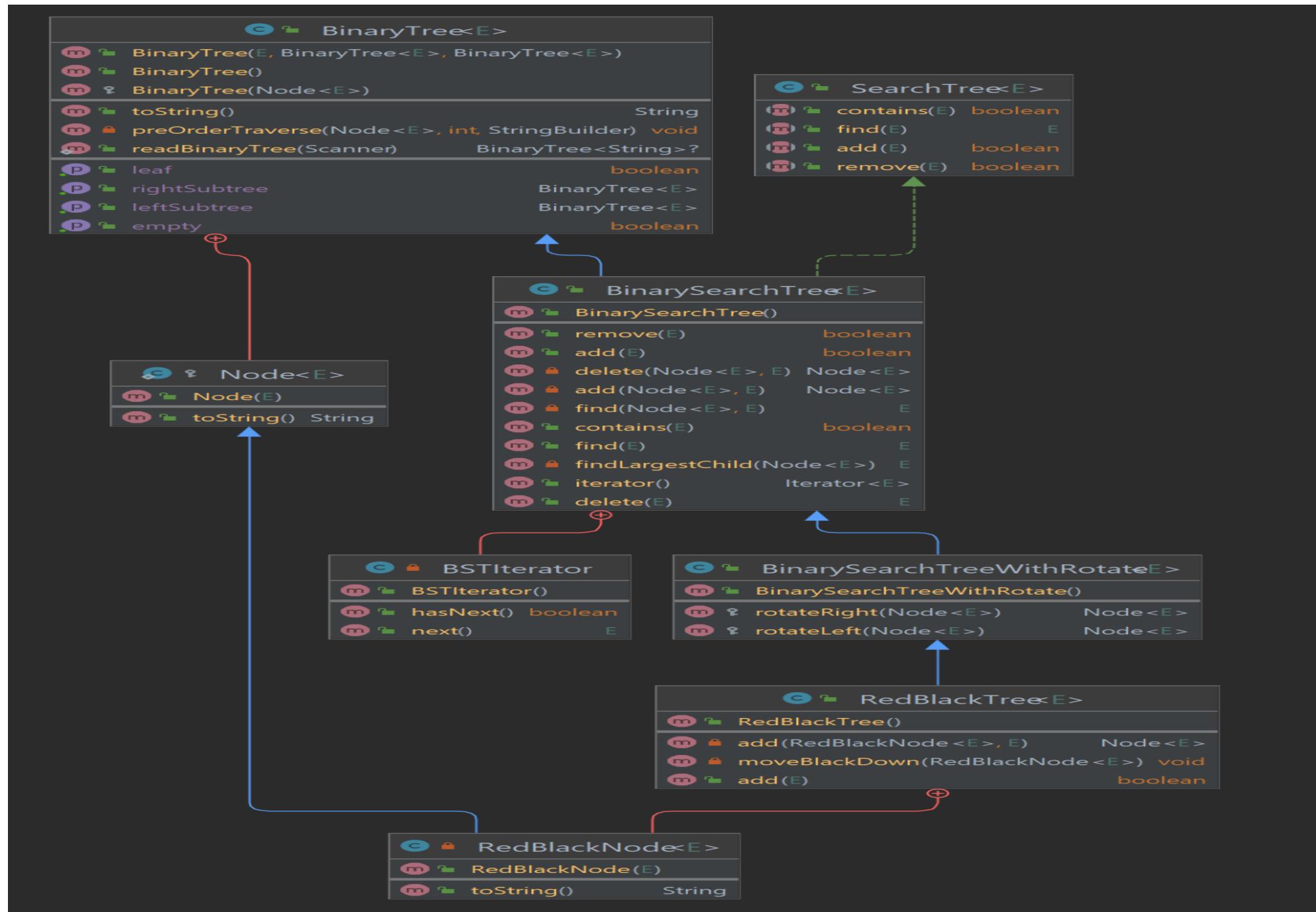
10. Sorting Algorithm(Quick Sort):

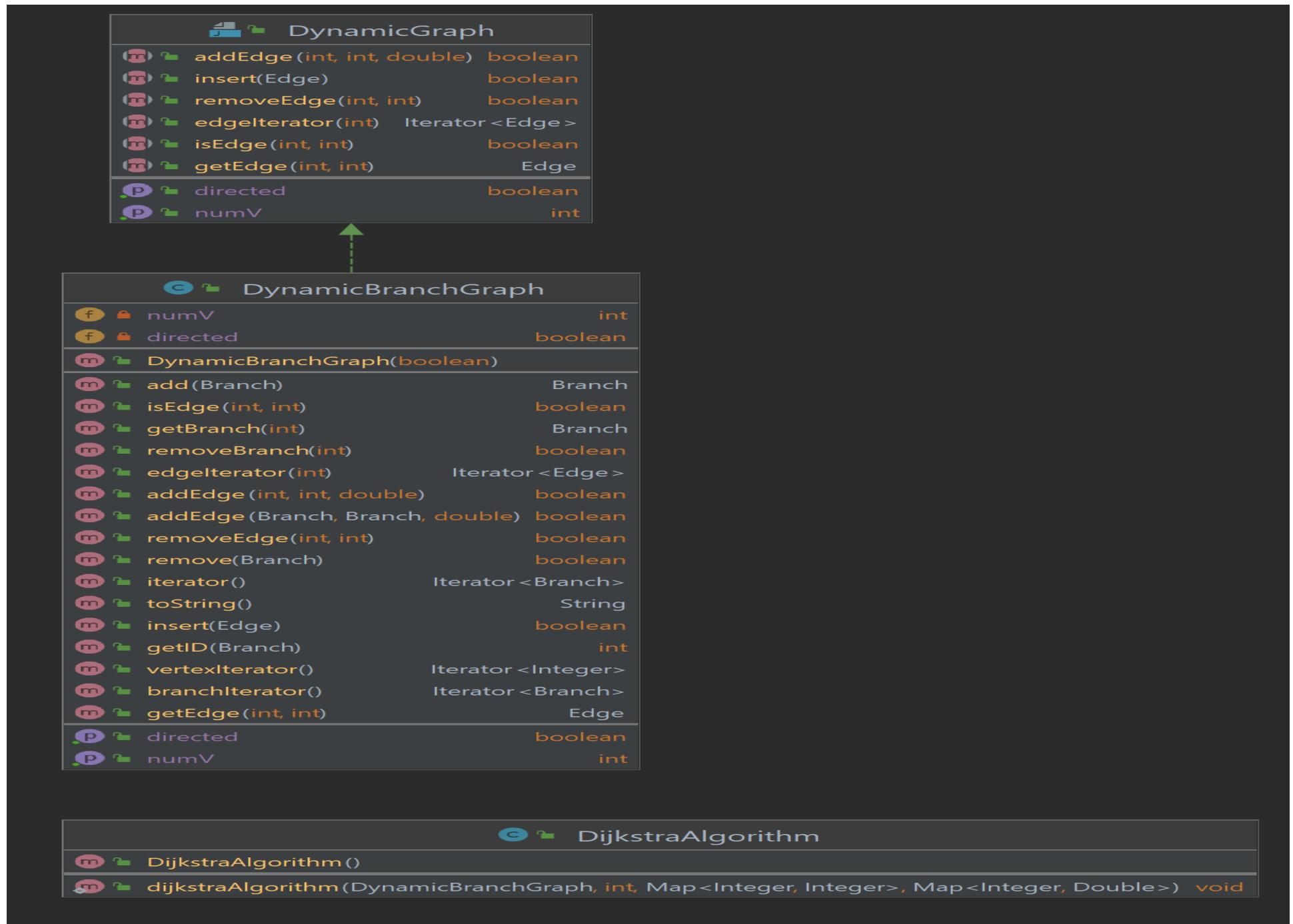
Customers might want to sort products specifically, there could be many products, therefore sorting algorithms which are logarithmic must be used. Between logarithmic sorts Quick Sort is preferred. Speed is more important than memory concerns, therefore Quick Sort is preferred above Heap Sort. Heap Sort has better memory usage, but it is empirically known that Quick Sort is faster than Heap Sort. Merge Sort, and Quick Sort are similar for speed, but Merge Sort has more memory usage than Quick Sort. Memory is not that much concern which causes using heap sort, but between Quick Sort, and Merge Sort. Quick Sort is preferred because of the memory usage advantage over Merge Sort. Quick Sort's worst case is avoided with randomized pivot choosing. Consequently, reaching Quick Sort's worst case has very low probability.

Class Diagram(sub-divided schema):

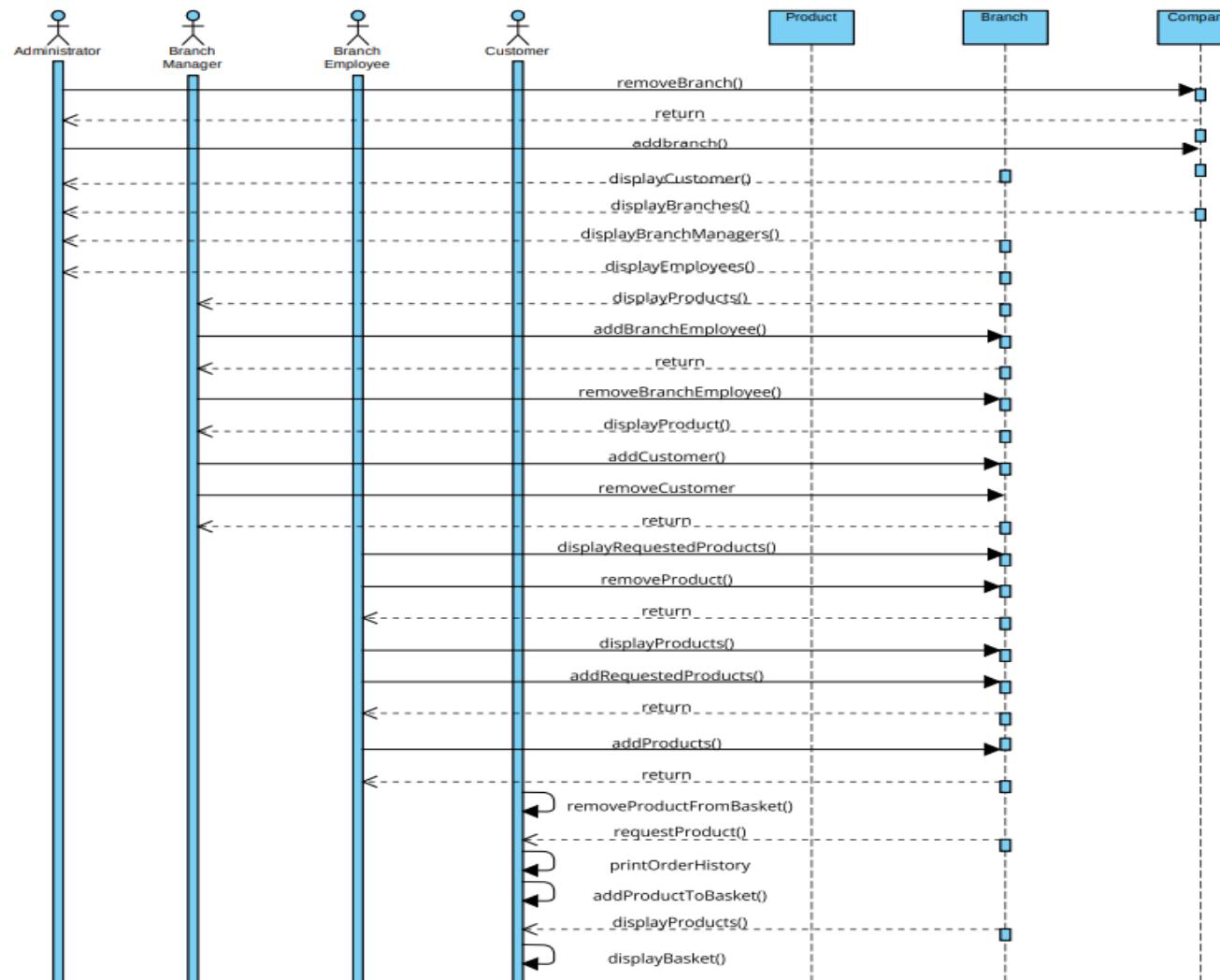




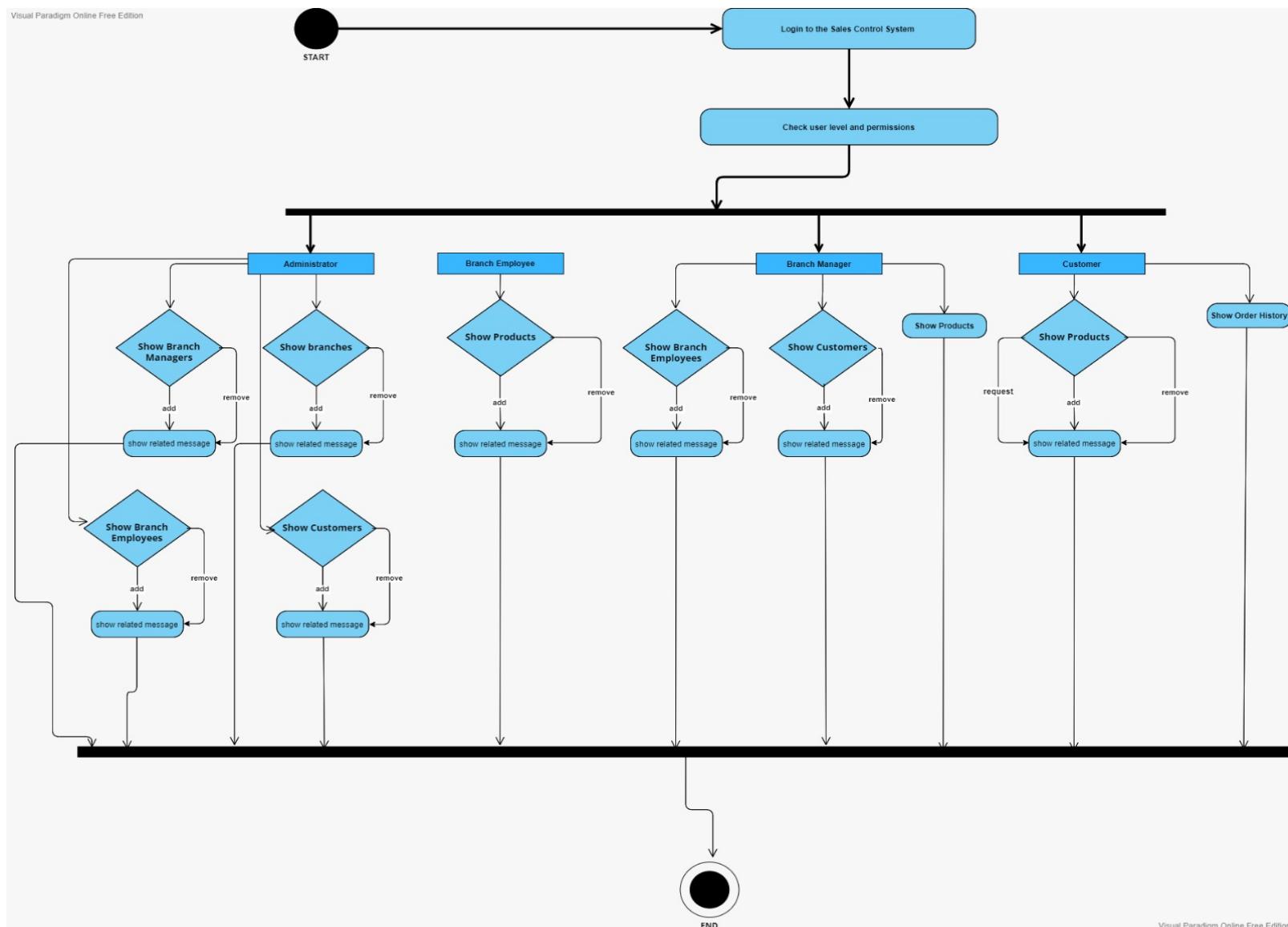




Sequence Diagram



Activity Diagram



Non-Trivial Implementation Details

Company: Holds the name, Administrator and list of branches as fields.

- **Company(String):** Constructs Company object by setting its name.
- **getCompanyName():** Returns the name of the company.
- **getBranches():** Returns the graph of branches.
- **getAdministrator():** Returns the Administrator of the company.
- **setAdministrator(Administrator):** Sets indicated Administrator as the Administrator of this company.

Branch: Holds the name of branch, branch manager, Red Black Tree of employees, HashMap of product type , and products(products are key as a TreeSet) , Skip List of customers, PriorityQueue of requested products as field.

- **Branch(String):** Constructs Branch object by setting the name of branch.
- **getBranchName():** Returns the name of the branch.
- **setBranchManager(BranchManager):** Sets the indicated branch manager as the manager of this branch, if the branch has not a manager.
- **getBranchManager():** Returns the manager of this branch.
- **getBranchEmployees():** Returns the BST(Red Black) of employees in this branch.
- **getCustomers():** Returns the Skip List of customers of this branch.
- **getProducts():** Returns the HashMap of products that are sold in this branch.
- **getProductsList(ProductType):** Returns the ArrayList of Products for specific type.
- **addSale(double):** Increments the sale price.
- **hasProduct(Product):** Checks if the branch has that product.
- **getStringEmployees():** Returns the String representation of Employees.
- **getStringCustomers():** Returns the String representation of Customers.
- **getStringProducts():** Returns the String representation of Products.
- **getRequestedProducts():** Returns the Priority Queue of requested products.

Product: Holds general information of products such as name, brand, type as String and entry price as double, number of stock as int. Also general profit rate is hold. It have inner comparator classes for sorting.

- **Product(String, String, String, double):** Constructs a Product object by setting name, brand, type and entry price.
- **getSalePrice():** Returns the sale price that is obtained by multiplying profit rate and entry price.
- **setEntryPrice(double):** Sets the indicated price as the entry price of this product.
- **getEntryPrice():** Returns the entry price of product.
- **setName(String):** Sets the given name as the name of product.
- **getName():** Returns the product name.
- **setBrand(String):** Sets the indicated brand name as the brand of this product.
- **getBrand():** Returns the brand of product.
- **setType(String):** Sets the indicated type as the type of this product.
- **decreaseStock():** Decreases the number of stocks.
- **getType():** Returns the type of product.

All type of product classes extends Product. Their constructors construct the objects by setting indicated informations. Getters returns the related informations.

Book: It has extra informations as kind and author.

Clothes: It has extra informations as size, material type, color, gender and is summery or not.

Drink: It has extra informations as expire date and liter.

Electronic: It has extra informations as guarantee time, width, height and is display product or not.

Food: It has extra informations as expire date and weight.

Furniture: It has extra informations as width, height and color.

PersonalCare: It has extra informations as usage area and expire date.

Toy: It has extra informations as playing age and kind.

User: Holds information of users such as name as String, age as int, gender as Gender enum.

- **User(String, int, Gender):** Constructs an User object by setting name, age and gender.
- **setName(String):** Sets the name of user.
- **getName():** Returns the name of user.
- **setAge(int):** Sets the age of user.
- **getAge():** Returns the age of user.
- **setGender(Gender):** Sets the gender of user.
- **getGender():** Returns the gender of user.

Administrator: Extends User. Holds the company, which it is responsible for, as field, gets informations through this company field and makes operations. Also creates branch graph. Admin can insert edges(roads) between branches to suggest closest branch for customer.

- **Administrator(String, int, Gender, String, String, Company):** Constructs an Administrator object by setting name, age, gender, username, password, and company. Sets itself as the administrator of the company.
- **setCompany(Company):** Sets this administrator as the administrator of the indicated company, if the company has not an administrator.
- **getCompany():** Returns the company that this administrator is responsible for.
- **addBranch(Branch):** Gets a branch and adds it to the branches list of company, if the branch is not added.
- **removeBranch(Branch):** Removes the indicated branch from branches list of company, if it is added.
- **setBranchManager(Branch, BranchManager):** Sets the indicated branch manager as the branch manager of the indicated branch.
- **connectBranches(Branch, Branch, double):** Adds an undirected edge between given branch references.
- **connectBranches(int, int, double):** Adds an undirected edge between given ids.
- **removeConnection(int, int):** Removes the edge between given branch ids.
- **displayBranchGraph():** Prints the edges, and vertices in adjacency list format.
- **displayBranches():** Prints the informations of all branches of this company.
- **displayTotalSalesPrice():** Prints the total sales price.
- **displayBranchManagers():** Prints the informations of branch managers of all branches.
- **displayBranchEmployees():** Prints the informations of branch employees for each branch.
- **displayCustomers():** Prints the informations of customers for each branch.

Branch Employee: Extends User. Holds the branch which employee belongs to, as field, gets informations through this branch field and makes operations.

- **BranchEmployee(String, int, Gender, String, String, Company):** Constructs a BranchEmployee object by setting name, age, gender, username, password, and company. It is set as an employee of the indicated branch.
- **addProduct(Product):** Adds the indicated product to the products list of branch.
- **removeProduct(Product):** Removes the indicated product with all its stocks from the products list of branch.
- **removeProduct(Product, int):** Removes the indicated product with indicated number of that product.
- **removeProduct(String):** Removes the indicated product according to the name with all its stocks from the products list of branch.
- **removeProduct(String, int):** Removes the indicated product according to the name with indicated number of that product.
- **findProductByName(String):** Returns the Product reference which has the name of the given parameter.
- **addRequestedProducts():** Gets the PriorityQueue of requested products from the branch. Adds the most requested one to the products list of branches.
- **displayProducts():** Prints the information of all products in this branch.
- **displayRequestedProducts():** Prints the information of requested products in this branch.
- **getBranch():** Returns the branch that this employee belongs to.

Branch Manager: Extends User. Holds the branch, which it is responsible for, as field, gets informations through this branch field and makes operations.

- **BranchManager(String, Gender, String, String, Branch):** Constructs a BranchManager object by setting name, age, gender, username, password, and branch. Sets itself as the manager of the branch.
- **setBranch(Branch):** Sets this branch manager as the manager of the indicated branch, if the branch has not a manager.
- **getBranch():** Returns the branch that this branch manager is responsible for.
- **displayBranches(Company):** Prints the all branches of the company.
- **displayBranchEmployees():** Prints the information of all branch employees in this branch.
- **displayProducts():** Prints the information of all products in this branch.
- **addCustomer(Customer):** Adds the indicated customer to the customers BST(Red Black) of the branch which this branch manager is responsible for.
- **removeCustomer(Customer):** Removes the indicated customer from the customers BST(Red Black) of the branch which this branch manager is responsible for.
- **addBranchEmployee(BranchEmployee):** Adds the indicated branch employee to the branch employees Skip List of the branch which this branch manager is responsible for.
- **removeBranchEmployee(BranchEmployee):** Removes the indicated branch employee from the branch employees Skip List of the branch which this branch manager is responsible for.

Customer: Extends User. Fields of this customer are the branch which customer does shopping, shopping basket as list of products, added products as stack and removed products as stack.

- **Customer(String, Gender, String, String, Branch):** Constructs a Customer object by setting name, age, gender, username, password, and branch. It is set as a customer of the indicated branch.
- **addProductToBasket(Product):** Adds the indicated product to the shopping basket of this customer.
- **removeProductFromBasket(Product):** Removes the indicated product from the shopping basket of this customer.
- **getBranchSuggestion(Company, Product):** Returns the closest Branch in the Company, which has wanted product. While finding the closest branch it uses the Dijkstra Algorithm.
- **findProductByName(String):** Finds the product according to the given name.
- **findProductFromBasketByOrder(int):** Finds the product from basket according to the index.
- **purchaseBasket():** Purchases all products in the basket.
- **requestProduct(Product):** Adds the indicated product to the PriorityQueue of requested products in branch.
- **displayProducts():** Prints the informations of all products in this branch.
- **printOrderHistory():** Prints the order history of this customer.
- **getShoppingBranch():** Returns the branch that this customer does shopping.

SCSystem: System class which provides an interface for a company's users. Administrator, Branch Managers, Branch Employees, Customers can manipulate the data with their restrictions in the system. Users are held in 2 HashMap, one of them has pairs username as key, password as value. Other one is username as key, User class as value.

Test Cases

| Test ID | Requirement ID | Test Scenario | Test Steps | Test Data | Expected Results | Actual Results | Pass /Fail |
|---------|----------------|--|--|--|---|----------------|------------|
| T01 | FR_1 | Check administrator login with valid data. | 1.Login the system with valid Admin data | 1.Admin's username 2.Admin's password | User should login into system as an administrator. | SHOWED BELOW | PASS |
| T02 | FR_1 | Check administrator login with invalid username. | 1.Enter invalid input for the username | 1.Admin's username | User should login into system as an administrator. | SHOWED BELOW | PASS |
| T03 | FR_1 | Check administrator login with invalid password. | 1.Enter valid input for the username 2.Enter invalid input for the password | 1.Admin's username 2.Admin's password | System should give an error message and should not login. | SHOWED BELOW | PASS |
| T04 | FR_1 | Check branch employee login with valid data. | 1.Login the system with valid Branch employee data | 1.Branch Employee's username 2.Branch Employee's password | User should login into system as a branch employee. | SHOWED BELOW | PASS |
| T05 | FR_1 | Check branch employee login with invalid username. | 1.Enter invalid input for the username | 1.Branch Employee's username | System should give an error message and should not login. | SHOWED BELOW | PASS |
| T06 | FR_1 | Check branch employee login with invalid password. | 1.Enter valid input for the username 2.Enter invalid input for the password | 1.Branch Employee's username 2.Branch Employee's password | System should give an error message and should not login. | SHOWED BELOW | PASS |
| T07 | FR_1 | Check branch manager login with valid data. | 1.Enter valid input for the username. 2. Enter valid input for password. | 1.Branch manager's username 2.Branch managers's password | User should login into system as a branch manager. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|--|---|--|---|--------------|------|
| T08 | FR_1 | Check branch manager login with invalid username. | 1.Enter invalid input for the username. | 1.Branch manager's username | System should give an error message and should not login. | SHOWED BELOW | PASS |
| T09 | FR_1 | Check branch manager login with invalid password. | 1.Enter valid input for the username. 2. Enter invalid input for password. | 1.Branch manager's username 2.Branch managers's password | System should give an error message and should not login. | SHOWED BELOW | PASS |
| T10 | FR_1 | Check customer login with valid data. | 1.Enter valid input for the username. 2. Enter valid input for password. | 1.Customer' s username 2.Customers 's password | User should login into system as a customer. | SHOWED BELOW | PASS |
| T11 | FR_1 | Check customer login with invalid username. | 1.Enter invalid input for the username. | 1.Customer' s username | System should give an error message and should not login. | SHOWED BELOW | PASS |
| T12 | FR_1 | Check customer login with invalid password. | 1.Enter valid input for the username. 2. Enter invalid input for password. | 1.Customer' s username 2.Customers 's password | System should give an error message and should not login. | SHOWED BELOW | PASS |
| T13 | | Administrator see the graph of branches. | 1.Login to the system as administrator. 2. Select - Connect Branches | 1.Admin's username 2.Admin's password 3.New user's -username -password | System should display the graph. | SHOWED BELOW | PASS |
| T14 | | Administrator connects the branches. | 1.Login to the system as administrator. 2. Select - Connect Branches 3. Enter the branches that wanted to connect | 1.Admin's username 2.Admin's password 3.New user's -username -password | The branches are should be connected. | SHOWED BELOW | PASS |
| T15 | FR_16 | Administrator approves all of the waiting registrations. | 1.Login to the system as administrator. | 1.Admin's username 2.Admin's | Administrator should successfully approve all of the waiting registrations. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|--|---|--|---|--------------|------|
| | | | 2. Select - Approve Waiting Registrations - from the menu 3. Select Approve All from the menu | password 3.New user's -username -password | | | |
| T16 | FR_16 | Administrator approves waiting registrations individually. | 1.Login to the system as administrator. 2. Select - Approve Waiting Registrations - from the menu 3. Select Approve individually from the menu 4. Select Approve from the menu | 1.Admin's username 2.Admin's password 3.New user's -username -password | Administrator should successfully approve waiting registrations individually. | SHOWED BELOW | PASS |
| T17 | FR_16 | Administrator rejects waiting registrations individually. | 1.Login to the system as administrator. 2. Select - Approve Waiting Registrations - from the menu 3. Select Approve individually from the menu 4. Select reject from the menu | 1.Admin's username 2.Admin's password 3.New user's -username -password | Administrator should successfully reject waiting registrations individually. | SHOWED BELOW | PASS |
| T18 | FR_16 | Administrator passes waiting registrations individually. | 1.Login to the system as administrator. 2. Select - Approve Waiting Registrations - from the menu 3. Select Approve individually from the menu 4. Select pass from the menu | 1.Admin's username 2.Admin's password 3.New user's -username -password | Administrator should successfully pass waiting registrations individually. | SHOWED BELOW | PASS |
| T19 | FR_16 | Administrator rejects all of the waiting registrations. | 1.Login to the system as administrator. 2. Select -Approve Waiting | 1.Admin's username 2.Admin's password 3.New | Administrator should successfully reject all of the waiting registrations. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|--|---|--|---|--------------|------|
| | | | Registrations - from the menu. 3. Select reject all from the menu | user's -username -password | | | |
| T20 | FR_17 | Administrator displays company sales price. | 1.Login to the system as administrator. 2. Select - Display Company Sales Price - from the menu. | 1.Admin's username 2.Admin's password | Administrator should see company's sales price. | SHOWED BELOW | PASS |
| T21 | FR_4 | Administrator sets branch Manager of the branch. | 1.Login to the system as administrator. 2. Select -Set Branch Manager - from the menu. | 1.Admin's username 2.Admin's password 3.Branch name 4.new Branch manager's -name -age -gender -username -password | Administrator should set branch manager of the branch in the company. | SHOWED BELOW | PASS |
| T22 | FR_3 | Administrator adds a branch to the system. | 1.Login to the system as administrator. 2. Select - Add Branch - from the menu. | 1.Admin's username 2.Admin's password 3.Branch name 4.Branch manager -name -age -gender -username -password | Administrator should successfully add branch to the system. | SHOWED BELOW | PASS |
| T23 | FR_3 | Administrator adds a branch with an existing name to the system. | 1.Login to the system as administrator. 2. Select - Add Branch - from the menu. | 1.Admin's username 2.Admin's password 3.Branch name 4.Branch manager -name | System should give an error message that says the name is already existing and should not add the Branch. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|---|---|--|--|--------------|------|
| | | | | -age -gender | | | |
| T24 | FR_3 | Administrator removes a branch from the system | 1.Login to the system as administrator. 2. Select - Remove Branch - from the menu. | 1.Admin's username 2.Admin's password 3.Branch name | Administrator should remove branch from the system. | SHOWED BELOW | PASS |
| T25 | FR_5 | Administrator displays branches. | 1.Login the system as an admin. 2. Select -Display Branches- from the menu. | 1.Admin's username 2.Admin's password | Administrator should see that branch's informations. | SHOWED BELOW | PASS |
| T26 | FR_5 | Administrator displays branch managers. | 1.Login the system as an admin. 2. Select -Display Branch Managers- from the menu. | 1.Admin's username 2.Admin's password | Administrator should see all branch managers. | SHOWED BELOW | PASS |
| T27 | FR_5 | Administrator displays branch employees. | 1.Login the system as an admin. 2. Select -Display Branch Employees- from the menu. | 1.Admin's username 2.Admin's password | Administrator should see all branch employees. | SHOWED BELOW | PASS |
| T28 | FR_5 | Administrator displays branch customers. | 1.Login the system as an admin. 2. Select -Display Branch Customers- from the menu. | 1.Admin's username 2.Admin's password | Administrator should see all Customers. | SHOWED BELOW | PASS |
| T29 | FR_10 | Branch employee adds a new product to the system. | 1.Login the system. 2.Enter Product name 3. Enter Brand name 4. Enter Type 5.Enter Price 6.Enter Stock | 1.Branch employee's username and password 2.Product information | Branch employee should successfully add product to the system. | SHOWED BELOW | PASS |
| T30 | FR_10 | Branch employee adds a new product with an existing name. | 1.Login the system. 2.Enter Product name 3. Enter Brand name 4. Enter Type | 1.Branch employee's username and password 2.Product information | System should give an error message that says the name is already existing and should not add the product. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|---|--|--|---|--------------|------|
| | | | 5.Enter Price 6.Enter Stock | s | | | |
| T31 | FR_10 | Branch employee removes a product from the system. | 1.Login the system. 2.Enter Product name | 1.Branch employee's username and password 2.Product name | Branch employee should remove that product from the system. | SHOWED BELOW | PASS |
| T32 | FR_10 | Branch employee displays the products. | 1.Login the system. 2.Select -Display Products- from the menu. | 1.Branch employee's username and | Branch employee should see all the products. | SHOWED BELOW | PASS |
| T33 | FR_9 | Branch manager displays the branch employees. | 1.Go to software 2.Login as a branch manager 3. Select -Display Branch Employees- from the menu. | 1.Branch manager's -password -username | Branch manager should see all the branch employees. | SHOWED BELOW | PASS |
| T34 | FR_9 | Branch manager displays the products. | 1.Go to software 2.Login as a branch manager 3. Select -Display the Products- from the menu. | 1.Branch manager's -password -username | Branch manager should see all the products. | SHOWED BELOW | PASS |
| T35 | FR_8 | Branch manager adds a new customer to the system. | 1.Go to software 2.Login as a branch manager 3. Select -Add Customer - from the menu. | 1.Branch manager's -password -username 2.New customer's -name -password -username | Branch manager should successfully add customer to the system. | SHOWED BELOW | PASS |
| T36 | FR_8 | Branch manager adds a new customer with an existing username. | 1.Go to software 2.Login as a branch manager 3. Select -Add Customer - from the menu. | 1.Branch manager's -username -password 2.An existed customer's -username | System should give an error message that says the name is already existing and should not add the customer. | SHOWED BELOW | PASS |
| T37 | FR_8 | Branch manager removes a customer from | 1.Go to software 2.Login as a branch | 1.Branch manager's | Branch manager should remove that customer from the system. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|--|---|--|---|--------------|------|
| | | the system. | manager 3. Select -Remove Customer - from the menu. | -username -password 2.An existed customer's -name | | | |
| T38 | FR_7 | Branch manager adds a new branch employee to the system. | 1.Go to software 2.Login as a branch manager 3. Select -Add Branch Employee - from the menu. | 1.Branch manager's -password -username 2.New branch employee's -name -password -username | Branch manager should successfully add branch employee to the system. | SHOWED BELOW | PASS |
| T39 | FR_7 | Branch manager adds a new branch employee with an existing username. | 1.Go to software 2.Login as a branch manager 3. Select -Add Branch Employee - from the menu. | 1.Branch manager's -password -username 2.An existed branch employee's -name | System should give an error message that says the name is already existing and should not add the branch employee . | SHOWED BELOW | PASS |
| T40 | FR_7 | Branch manager removes a branch employee from the system. | 1.Go to software 2.Login as a branch manager 3. Select -Remove Branch Employee - from the menu. | 1.Branch manager's -password -username 2.An existed branch employee's -name | Branch manager should remove that branch employee from the system. | SHOWED BELOW | PASS |
| T41 | FR_9 | Branch manager displays the Branch Sales Price | 1.Go to software 2.Login as a branch manager 3. Select - Branch Sales Price - from the menu | 1.Branch manager's -password -username | Branch manager should see the Branch Sales Price . | SHOWED BELOW | PASS |
| T42 | FR-13 | Customer requests a product. | 1.Go to software 2.Login as a customer 3. Select - display products 4. Select - category 5. Select - request product. | 1.Customer's password -username 2.Product information | Success feedback is given. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|---|---|---|--|--------------|------|
| | | | 6. Input wanted product information. | | | | |
| T43 | FR_14 | Customer sees the products. | 1.Go to software 2.Login as a customer 3. Select - display products 4. Select - category | 1.Customer's password -username | Customer should see the products in selected category. | SHOWED BELOW | PASS |
| T44 | FR_15 | Customer sees order history. | 1.Go to software 2.Login as a customer 3. Select - display order history | 1.Customer's password -username | Customer should see all the order history. | SHOWED BELOW | PASS |
| T45 | FR_12 | Customer can buy products. | 1.Go to software 2.Login as a customer 3. Select - display basket 4.Select - buy products | 1.Customer's password -username | Customer should purchase the products. | SHOWED BELOW | PASS |
| T46 | FR_13 | Customer gets branch suggestion of nonexistent product. | 1.Go to software 2.Login as a customer 3. Select - display products 4.Select -choose category 5. Select branch suggestion 6. Enter the existed product | 1.Customer's -password -username 2.Existed product's information | If there is product nearby branch , customer should see the closest branch that has the product. | SHOWED BELOW | PASS |
| T47 | FR_12 | Customer adds product to basket. | 1.Go to software 2.Login as a customer 3. Select - display products 4.Select - choose category. 5. Select - add | 1.Customer's password -username | The product should added in basket. | SHOWED BELOW | PASS |

| | | | | | | | |
|-----|-------|--|--|--|--|--------------|------|
| | | | product to basket 6.Enter valid product information | | | | |
| T48 | FR_12 | Customer removes product from basket. | 1.Go to software 2.Login as a customer 3. Select - display basket 4.Select - remove product 5. Choose the item that wanted to remove | 1.Customer's -password -username | The product should removed from basket. | SHOWED BELOW | PASS |
| T49 | FR_14 | Customer see sorted products. | 1.Go to software 2.Login as a customer 3. Select - display products 4. Select - category 5. Select sorting method | 1.Customer's -password -username | Customer should see all the products in order that customer wants. | SHOWED BELOW | PASS |
| T50 | | Customer try to add invalid product to basket. | 1.Go to software 2.Login as a customer 3. Select - display products 4.Select - choose category. 5. Select - add product to basket 6.Enter invalid product information | 1.Customer's -password -username | System should give error message. | SHOWED BELOW | PASS |
| T51 | | Customer try to remove invalid product. | 1.Go to software 2.Login as a customer 3. Select - display basket 4.Select - remove product 5. Choose invalid | | System should give error message. | SHOWED BELOW | PASS |

| | | | item | | | | |
|-----|--|-----------------------|---|---------------------------------|-------------------------------------|---------------------|------|
| T52 | | Customer sees basket. | 1.Go to software 2.Login as a customer 3. Select - display basket | 1.Customer's password -username | Customer should see all the basket. | SHOWED BELOW | PASS |

System(Menu) Test

Results

- T01

```
Enter username: ADMIN
Enter password: adminscs.

-----
ADMIN SIGNED IN SUCCESSFULLY.
```

- T02

```
----- USER SIGN IN -----
- 0 - Exit

Enter username: ADMIn
THERE IS NO USER IN SYSTEM WITH THIS USERNAME!

Enter username: █
```

- T03

```
----- USER SIGN IN -----
- 0 - Exit

Enter username: ADMIN
Enter password: adminscts
WRONG PASSWORD!

Enter password:
```

- T04

```
----- BRANCH MANAGER MENU -----
- 1 - Add Branch Employee
- 2 - Remove Branch Employee
- 3 - Add Customer
- 4 - Remove Customer
- 5 - Display Branch Employees
- 6 - Display Products
- 7 - Display Branch Sales Price
- 0 - Exit

Choice: 1
[H[2]
Enter Branch Employee Name To Add:
Enter Name: Parker

Enter Age: 25

Male: M-m   Female: F-f   Other: O-o
Enter Gender: f

Enter Username: parkerk
Enter Password(Min 8 Charachters): parkerKATE

----- SALES CONTROL SYSTEM -----
- COMPANY : BIG SUPERMARKET
- 1 - Sign Up
- 2 - Sign In
- 0 - Exit

Choice: 2
[H[2]
----- USER SIGN IN -----
- 0 - Exit

Enter username: parkerk
Enter password: parkerKATE
[H[2]
parkerk SIGNED IN SUCCESSFULLY.
[H[2]
----- BRANCH EMPLOYEE MENU -----
- 1 - Add Product
- 2 - Remove Product
- 3 - Add Requested Products
- 4 - Display Products
- 5 - Display Requested Products
- 0 - Exit

Choice:
```

- T05 / T08 / T11

```
----- USER SIGN IN -----  
- 0 - Exit  
  
Enter username: erayozkan  
THERE IS NO USER IN SYSTEM WITH THIS USERNAME!  
  
Enter username: [REDACTED]
```

- T06

```
----- USER SIGN IN -----  
- 0 - Exit  
  
Enter username: Sevim7  
  
Enter password: S_evim2001  
WRONG PASSWORD!  
  
Enter password: [REDACTED]
```

- T07

```
----- SALES CONTROL SYSTEM -----  
- COMPANY : BIG SUPERMARKET -  
- 1 - Sign Up -  
- 2 - Sign In -  
- 0 - Exit -  
  
Choice: 2  
[H[2J  
----- USER SIGN IN -----  
- 0 - Exit -  
  
Enter username: mngrArda  
Enter password: 1903arda.  
[H[2J  
mngrArda SIGNED IN SUCCESSFULLY.  
[H[2J  
----- BRANCH MANAGER MENU -----  
- 1 - Add Branch Employee -  
- 2 - Remove Branch Employee -  
- 3 - Add Customer -  
- 4 - Remove Customer -  
- 5 - Display Branch Employees -  
- 6 - Display Products -  
- 7 - Display Branch Sales Price -  
- 0 - Exit -  
  
Choice:
```

● T09

```
----- USER SIGN IN -----  
- 0 - Exit  
  
Enter username: tarik.mng  
Enter password: 123456789  
WRONG PASSWORD!  
Enter password: █
```

● T10

```
----- SALES CONTROL SYSTEM -----  
- COMPANY : BIG SUPERMARKET -  
- 1 - Sign Up -----  
- 2 - Sign In -----  
- 0 - Exit -----  
Choice: 2  
[H[2]  
----- USER SIGN IN -----  
- 0 - Exit -----  
  
Enter username: avni.celik  
Enter password: avn_1234  
[H[2]  
avni.celik SIGNED IN SUCCESSFULLY.  
[H[2]  
----- CUSTOMER MENU -----  
- 1 - Display Products -----  
- 2 - Display Basket -----  
- 3 - Display Order History -----  
- 0 - Exit -----  
Choice:
```

● T12

```

----- USER SIGN IN -----
- 0 - Exit

Enter username: avni.celik
Enter password: avn_123214
WRONG PASSWORD!

Enter password: █

```

- T13

```

Adjacency List
vertex (id = 1, branch name = branch1)-->
    [(1, 2): 12.2]
    [(1, 4): 15.6]
vertex (id = 2, branch name = branch2)-->
    [(2, 1): 12.2]
    [(2, 3): 10.9]
    [(2, 4): 8.7]
vertex (id = 3, branch name = branch3)-->
    [(3, 2): 10.9]
vertex (id = 4, branch name = branch4)-->
    [(4, 2): 8.7]
    [(4, 1): 15.6]

```

Choose Source Branch:
Existing Branches

| ----- Branches ----- | |
|----------------------|----------------|
| Branch Name | Branch Manager |
| branch1 | tarık |
| branch2 | osman |
| branch3 | irem |
| branch4 | arda |

Enter Branch Name: █

- T14

```

Choose Source Branch:
Existing Branches

----- Branches -----
Branch Name      Branch Manager
branch1          tarık
branch2          osman
branch3          irem
branch4          arda

----- Enter Branch Name: branch1 -----



Choose Destination Branch:
Existing Branches

----- Branches -----
Branch Name      Branch Manager
branch1          tarık
branch2          osman
branch3          irem
branch4          arda

----- Enter Branch Name: branch3 -----



Enter Weight: 18.3
BRANCHES ARE CONNECTED SUCCESSFULLY!

```

```

Adjacency List
vertex (id = 1, branch name = branch1)-->
    [(1, 2): 12.2]
    [(1, 4): 15.6]
    [(1, 3): 18.3]
vertex (id = 2, branch name = branch2)-->
    [(2, 1): 12.2]
    [(2, 3): 10.9]
    [(2, 4): 8.7]
vertex (id = 3, branch name = branch3)-->
    [(3, 2): 10.9]
    [(3, 1): 18.3]
vertex (id = 4, branch name = branch4)-->
    [(4, 2): 8.7]
    [(4, 1): 15.6]

```

- T15

Waiting Registrations :

```
User Type: Branch Employee
Branch: branch1
Name: irem
Username: icy
Age: 21
Gender: FEMALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

```
User Type: Branch Employee
Branch: branch3
Name: Kemal
Username: mkemal
Age: 32
Gender: MALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

Approving all:

```
1 - Approve All
2 - Approve Individually
3 - Reject All
Choice: 1
ALL REGISTRATION REQUESTS ARE APPROVED SUCCESSFULLY!
```

```
----- ADMINISTRATOR MENU -----
```

```
- 1 - Approve Waiting Registrations
- 2 - Add Branch
- 3 - Remove Branch
- 4 - Set Branch Manager
- 5 - Connect Branches
- 6 - Display Branches
- 7 - Display Branch Managers
- 8 - Display Branch Employees
- 9 - Display Customers
- 10 - Display Company Sales Price
- 0 - Exit
```

```
Choice: 1
```

```
THERE IS NO REGISTRATION REQUEST!
```

- T16

Waiting Registrations :

```
User Type: Branch Employee
Branch: branch1
Name: irem
Username: icy
Age: 21
Gender: FEMALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

```
User Type: Branch Employee
Branch: branch3
Name: Kemal
Username: mkemal
Age: 32
Gender: MALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

```
User Type: Customer
Branch: branch4
Name: Aslı
Username: asli
Age: 22
Gender: FEMALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

Approving Individually:

```
1 - Approve All
2 - Approve Individually
3 - Reject All
Choice: 2

User Type: Branch Employee
Branch: branch4
Name: Aslı
Username: asli
Age: 22
Gender: FEMALE
```

```
1 - Approve
2 - Reject
3 - Pass
Choice: 1
REGISTRATION IS APPROVED!
```

- T17

Waiting Registrations :

```
User Type: Branch Employee
Branch: branch3
Name: Kemal
Username: mkemal
Age: 32
Gender: MALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

```
User Type: Customer
Branch: branch4
Name: Asli
Username: asli
Age: 22
Gender: FEMALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

Rejecting Individually:

```
1 - Approve All
2 - Approve Individually
3 - Reject All
Choice: 2

User Type: Branch Employee
Branch: branch3
Name: Kemal
Username: mkemal
Age: 32
Gender: MALE
```

```
1 - Approve
2 - Reject
3 - Pass
Choice: 2
REGISTRATION IS REJECTED!
```

```
User Type: Customer
Branch: branch4
Name: Asli
Username: asli
Age: 22
Gender: FEMALE
```

```
1 - Approve
2 - Reject
3 - Pass
Choice: 2
REGISTRATION IS REJECTED!
```

- T18

Waiting Registrations :

```
User Type: Branch Employee
Branch: branch1
Name: İrem
Username: icy
Age: 21
Gender: FEMALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

```
User Type: Branch Employee
Branch: branch3
Name: Kemal
Username: mkemal
Age: 32
Gender: MALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

Passing Individually:

```
User Type: Branch Manager
Branch: branch3
Name: Kemal
Username: mstfkml
Age: 32
Gender: MALE
```

```
1 - Approve
2 - Reject
3 - Pass
Choice: 3
USER WILL BE EXAMINED LATER!
```

```
User Type: Branch Employee
Branch: branch1
Name: İrem
Username: icy
Age: 21
Gender: FEMALE
```

```
1 - Approve
2 - Reject
3 - Pass
Choice: 3
USER WILL BE EXAMINED LATER!
```

- T19

Waiting Registrations :

```
User Type: Branch Employee
Branch: branch1
Name: İrem
Username: icy
Age: 21
Gender: FEMALE
REGISTRATION REQUEST RECEIVED SUCCESSFULLY.
```

Rejecting All:

```
1 - Approve All
2 - Approve Individually
3 - Reject All
Choice: 3
ALL REGISTRATION REQUESTS ARE REJECTED SUCCESSFULLY!

-----
----- ADMINISTRATOR MENU -----
1 - Approve Waiting Registrations
2 - Add Branch
3 - Remove Branch
4 - Set Branch Manager
5 - Connect Branches
6 - Display Branches
7 - Display Branch Managers
8 - Display Branch Employees
9 - Display Customers
10 - Display Company Sales Price
0 - Exit

Choice: 1

THERE IS NO REGISTRATION REQUEST!
```

- T20

before the sale:

```
----- ADMINISTRATOR MENU -----
- 1 - Approve Waiting Registrations
- 2 - Add Branch
- 3 - Remove Branch
- 4 - Set Branch Manager
- 5 - Display Branches
- 6 - Display Branch Managers
- 7 - Display Branch Employees
- 8 - Display Customers
- 9 - Display Company Sales Price
- 0 - Exit

choice: 9

----- Total Amount Of Sales Of Company : 0.00 -----
```

Customer buys product:

```
----- Basket of Customer Umut -----
Product Type    Brand Name      Product Name     Price      Stock
-----
Clothes         mavi           Short            145.34     4
-----
Total Amount : 145.34
-----
1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu
Choice: 1
```

After the sale:

```
-----
Total Amount Of Sales Of Company : 145.34
-----
```

● T21

```
Existing Branches
----- Branches -----
Branch Name        Branch Manager
-----
branch1           tarık
branch3           irem
branch4           arda
newBranch         tarık
-----
Enter Branch Name: branch1
Enter Informations Of Branch Manager:
Enter Name: new Manager
Enter Age: 24
Male: M-m   Female: F-f   Other: O-o
Enter Gender: f
Enter Username: newmngr
Enter Password(Min 8 Charachters): newmngr.
BRANCH MANAGER new Manager IS SET SUCCESSFULLY!
```

- T22

```
Enter Branch Name: newBranch

Enter Informations Of Branch Manager:
Enter Name: tarık

Enter Age: 43

Male: M-m    Female: F-f    Other: O-o
Enter Gender: m

Enter Username: trkmngr

Enter Password(Min 8 Charachters): trkmngr.
BRANCH IS ADDED SUCCESSFULLY!
```

- T23

| Branch Name | Branch Manager |
|-------------|----------------|
| branch1 | tarık |
| branch2 | osman |
| branch3 | irem |
| branch4 | arda |

```
----- ADMINISTRATOR MENU -----
1 - Approve Waiting Registrations
2 - Add Branch
3 - Remove Branch
4 - Set Branch Manager
5 - Connect Branches
6 - Display Branches
7 - Display Branch Managers
8 - Display Branch Employees
9 - Display Customers
10 - Display Company Sales Price
0 - Exit

Choice: 2

Enter Branch Name: branch4
BRANCH IS ALREADY ADDED!
```

- T24

```
Existing Branches
----- Branches -----
Branch Name      Branch Manager
-----
branch1          tarık
branch2          osman
branch3          irem
branch4          arda
newBranch        tarık
-----
Enter Branch Name: branch2
BRANCH IS REMOVED SUCCESSFULLY!
```

- T25

```
----- Branches -----
Branch Name      Branch Manager
-----
branch1          tarık
branch2          osman
branch3          irem
branch4          arda
-----
```

- T26

```
User Type: Branch Manager
Branch: branch1
Name: tarık
Username: tarik.mng
Age: 31
Gender: MALE

User Type: Branch Manager
Branch: branch2
Name: osman
Username: mngrOsman
Age: 45
Gender: MALE

User Type: Branch Manager
Branch: branch3
Name: irem
Username: mngrIrem
Age: 39
Gender: FEMALE

User Type: Branch Manager
Branch: branch4
Name: arda
Username: mngrArda
Age: 43
Gender: MALE
```

- T27

| Employees of BIG SUPERMARKET | | |
|------------------------------|---------------|-------------------|
| Branch Name | Employee Name | Employee Username |
| branch1 | osman | Osmanpzs |
| branch1 | asli | aslisi |
| branch1 | sevim | Sevim7 |
| branch3 | eray | Eray7 |
| branch3 | M.emre | emre07 |
| branch3 | esra | Esra |
| branch4 | furkan | furki |
| branch4 | eray | erayoz |
| branch4 | mehmet | memo |

- T28

| Branch Name | Customer Name | Customer Username |
|-------------|---------------|-------------------|
| branch1 | Umut | umut |
| branch1 | ali | ali53 |
| branch1 | alper | alper |
| branch1 | avni | avni.celik |
| branch1 | emir | emir.efe34 |
| branch1 | kerem | kerem |
| branch1 | necip | neco |
| branch1 | oğuzhan | ozi |
| branch1 | tümay | tümay |
| branch3 | elif | Elif |
| branch3 | necip | neco |
| branch3 | oğuzhan | ozi |
| branch4 | Umut | umut |
| branch4 | burcu | burcu |
| branch4 | kerem | kerem |

- T29

Enter Product Name: chocolate

Enter Brand Name: ulker

- 1 - Book
- 2 - Clothes
- 3 - Drink
- 4 - Electronic
- 5 - Food
- 6 - Furniture
- 7 - Personal Care
- 8 - Toy

Enter Type: 5

Enter Entry Price: 5.0

Enter Stock: 100

- BRANCH EMPLOYEE MENU

- 1 - Add Product
- 2 - Remove Product
- 3 - Add Requested Products
- 4 - Display Products
- 5 - Display Requested Products
- 0 - Exit

Choice: 4

| Products | | | | |
|-------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Personal Care | blendax | shampoo | 52,00 | 3 |
| 2 - Furniture | ikea | table | 260,00 | 4 |
| 3 - Drink | coca cola | coke | 14,30 | 45 |
| 4 - Food | ulker | chocolate | 6,50 | 100 |
| 5 - Food | algida | ice-cream | 11,05 | 16 |
| 6 - Clothes | mavi | T_shirt | 130,00 | 15 |
| 7 - Clothes | mavi | Jeans | 260,00 | 14 |

- T30

| Products | | | | | |
|------------------|------------|--------------|--------|-------|--|
| Product Type | Brand Name | Product Name | Price | Stock | |
| 1 - Drink | coca cola | coke | 14,30 | 45 | |
| 2 - Clothes | mavi | T_shirt | 130,00 | 16 | |
| 3 - Clothes | mavi | Jeans | 260,00 | 14 | |
| 4 - Personal Car | blendax | shampoo | 52,00 | 3 | |
| 5 - Food | algida | ice-cream | 11,05 | 16 | |
| 6 - Furniture | ikea | table | 260,00 | 4 | |

- BRANCH EMPLOYEE MENU

- 1 - Add Product
- 2 - Remove Product
- 3 - Add Requested Products
- 4 - Display Products
- 5 - Display Requested Products
- 0 - Exit

Choice:

```

Enter Product Name: coke
Enter Brand Name: coca cola
1 - Book
2 - Clothes
3 - Drink
4 - Electronic
5 - Food
6 - Furniture
7 - Personal Care
8 - Toy
Enter Type: 3
Enter Entry Price: 11
Enter Stock: 15
----- BRANCH EMPLOYEE MENU -----
1 - Add Product
2 - Remove Product
3 - Add Requested Products
4 - Display Products
5 - Display Requested Products
0 - Exit
Choice: 1

```

| Products | | | | |
|------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Drink | coca cola | coke | 14,30 | 60 |
| 2 - Clothes | mavi | T_shirt | 130,00 | 16 |
| 3 - Clothes | mavi | Jeans | 260,00 | 14 |
| 4 - Personal Car | blendax | shampoo | 52,00 | 3 |
| 5 - Food | algida | ice-cream | 11,05 | 16 |
| 6 - Furniture | ikea | table | 260,00 | 4 |

• T31

| Products | | | | |
|------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Personal Car | blendax | shampoo | 52,00 | 3 |
| 2 - Furniture | ikea | table | 260,00 | 4 |
| 3 - Drink | coca cola | coke | 14,30 | 45 |
| 4 - Food | ulker | chocolate | 6,50 | 100 |
| 5 - Food | algida | ice-cream | 11,05 | 16 |
| 6 - Clothes | mavi | T_shirt | 130,00 | 15 |
| 7 - Clothes | mavi | Jeans | 260,00 | 14 |

Enter The Product Name To Remove: chocolate

● T32

```

----- BRANCH EMPLOYEE MENU -----
- 1 - Add Product
- 2 - Remove Product
- 3 - Add Requested Products
- 4 - Display Products
- 5 - Display Requested Products
- 0 - Exit

Choice: 4
[H[2]
----- Products -----
Product Type    Brand Name    Product Name    Price    Stock
1 - Personal Car    blendax      shampoo        52,00     3
2 - Furniture       ikea         table          260,00    4
3 - Drink           coca cola   coke           14,30    45
4 - Food            algida      ice-cream      11,05    16
5 - Clothes          mavi        T_shirt        130,00   15
6 - Clothes          mavi        Jeans          260,00   14

```

● T33

```

----- BRANCH MANAGER MENU -----
- 1 - Add Branch Employee
- 2 - Remove Branch Employee
- 3 - Add Customer
- 4 - Remove Customer
- 5 - Display Branch Employees
- 6 - Display Products
- 7 - Display Branch Sales Price
- 0 - Exit

Choice: 5
----- Employees of branch1 -----
Employee Name      Employee Username
sevim             Sevim7

```

● T34

```

Choice: 6
----- Products -----
Product Type    Brand Name    Product Name    Price    Stock
1 - Clothes      brand2       clothes2       145,34    4
2 - Personal Car    blendax      shampoo        98,93    3
3 - Drink         brand3       tea            16,38    45
4 - Furniture     ikea         furniture1    171,47    4
5 - Electronic    samsung      phone          1300,00   42
6 - Food          ruffle       crisps        172,38    4

```

- T35

```
Choice: 3

Enter Customer Name To Add:
Enter Name: Leyla

Enter Age: 40

Male: M-m Female: F-f Other: O-o
Enter Gender: f

Enter Username: Leyla8

Enter Password(Min 8 Charachters): 23451234421
CUSTOMER IS ADDED SUCCESSFULLY.
```

- T36

```
Choice: 3

Enter Customer Name To Add:
Enter Name: ali

Enter Age: 34

Male: M-m Female: F-f Other: O-o
Enter Gender: m

Enter Username: malis3

THIS USERNAME IS TAKEN ALREADY!
Enter Username: |
```

- T37

```
-----
Choice: 4

----- Customers -----
Name          Gender      Age
-----
Umut          MALE        28
ali           MALE        29
alper         MALE        21
avni          MALE        25
emir          MALE        25
kerem         MALE        31
necip         MALE        19
oğuzhan      MALE        30
tümay         FEMALE      25
-----


Enter Customer Name To Remove: necip
Selected customer is removed successfully:

----- Customers -----

```

| Name | Gender | Age |
|---------|--------|-----|
| Umut | MALE | 28 |
| ali | MALE | 29 |
| alper | MALE | 21 |
| avni | MALE | 25 |
| emir | MALE | 25 |
| kerem | MALE | 31 |
| oğuzhan | MALE | 30 |
| tümay | FEMALE | 25 |

● T38

```
Choice: 1

Enter Branch Employee Name To Add:
Enter Name: ahmet

Enter Age: 65

Male: M-m    Female: F-f    Other: O-o
Enter Gender: m

Enter Username: ahmet65

Enter Password(Min 8 Charachters): ah123456
BRANCH EMPLOYEE IS ADDED SUCCESSFULLY.
```

● T39

```
Choice: 1

Enter Branch Employee Name To Add:
Enter Name: osman

Enter Age: 45

Male: M-m    Female: F-f    Other: O-o
Enter Gender: m

Enter Username: osmanpzs

Enter Username: THIS USERNAME IS TAKEN ALREADY!
```

● T40

```
Choice: 2

----- Employees of branch1 -----
Employee Name      Employee Username
-----
osman            Osmanpzs
ahmet           ahmet65
Osman           osmannnnnn
sevim            Sevim7
-----

Enter Branch Employee Name To Remove: sevim
Selected branch employee is removed successfully

----- Employees of branch1 -----
Employee Name      Employee Username
-----
osman            Osmanpzs
ahmet           ahmet65
Osman           osmannnnnn
-----
```

● T41

```
Choice: 7

-----
Total Amount Of Sales Of Branch : 0,00
-----
```

- T42

```
Enter Product Name: Dress
Enter Brand Name: mavi
1 - Book
2 - Clothes
3 - Drink
4 - Electronic
5 - Food
6 - Furniture
7 - Personal Care
8 - Toy
Enter Type: 2
PRODUCT IS REQUESTED FROM BRANCH!
```

- T43

```
Choice: 2
Category: Clothes
Name: T_shirt
Brand Name: mavi
Type: Clothes
Sale Price: 130,00
Size: XL
Material Type: matType1
Color: blue
Summery State: true
Gender: FEMALE

Category: Clothes
Name: Short
Brand Name: mavi
Type: Clothes
Sale Price: 145,34
Size: L
Material Type: matType1
Color: green
Summery State: true
Gender: MALE
```

- T44

```
-----  
Last Removed Product from Basket: Short  
Last Added Product to Basket: tea  
-----
```

```
----- CUSTOMER MENU -----
```

- 1 - Display Products
- 2 - Display Basket
- 3 - Display Order History
- 0 - Exit

```
-----  
Choice: |
```

- T45

| Basket of Customer avni | | | | |
|-------------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Clothes | mavi | T shirt | 130,00 | 1 |
| 2 - Drink | çaykur | tea | 16,38 | 45 |
| 3 - Drink | çaykur | tea | 16,38 | 45 |

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

Choice: 2

Enter Product Number: 1

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

Choice: 1

| Basket of Customer avni | | | | |
|-------------------------|------------|--------------|-------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| Drink | çaykur | tea | 16,38 | 45 |
| Drink | çaykur | tea | 16,38 | 44 |

Total Amount : 32,76

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

Choice: 1

- T46

1 - Add Product To Basket
2 - Request Product
3 - Branch Suggestion For Product That Not Available
0 - Back To Menu

Choice: 3

Enter Product Name: Dress

Enter Brand Name: mavi

1 - Book
2 - Clothes
3 - Drink
4 - Electronic
5 - Food
6 - Furniture
7 - Personal Care
8 - Toy

Enter Type: 2

Enter Entry Price: 250
The Closest Branch is branch2.

- T47

```

1 - Add Product To Basket
2 - Request Product
3 - Branch Suggestion For Product That Not Available
0 - Back To Menu
Choice: 1

```

Enter Product Name: tea

| ----- Basket of Customer avni ----- | | | | |
|-------------------------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | stock |
| 1 - Clothes | mavi | T_shirt | 130,00 | 1 |
| 2 - Drink | çaykur | tea | 16,38 | 45 |
| 3 - Drink | çaykur | tea | 16,38 | 45 |

```

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

```

Choice:

- T48

| ----- Basket of Customer elif ----- | | | | |
|-------------------------------------|------------|--------------|-------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Drink | coca cola | coke | 14,30 | 45 |

```

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

```

Choice: 2

Enter Product Number: 1

| ----- Basket of Customer elif ----- | | | | |
|-------------------------------------|------------|--------------|-------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| | | | | |

```

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

```

Choice:

- T49

```

1 - Add Product To Basket
2 - Request Product
3 - Branch Suggestion For Product That Not Available
4 - Display Sorted
0 - Back To Menu
Choice: 4

```

```

1 - Compare by Brand
2 - Compare by Name
3 - Compare by Price
4 - Compare by Type
5 - Compare by Gender
6 - Compare by Kind
7 - Compare by Material Type
Choice: 3

```

```

Category: Clothes
Name: T_shirt
Brand Name: mavi
Type: Clothes
Sale Price: 130,00
Size: L
Material Type: matType1
Color: blue
Summery State: true
Gender: FEMALE

Category: Clothes
Name: Jeans
Brand Name: levi's
Type: Clothes
Sale Price: 169,00
Size: M
Material Type: matType3
Color: green
Summery State: false
Gender: FEMALE

Category: Clothes
Name: Jeans
Brand Name: levi's
Type: Clothes
Sale Price: 182,00
Size: L
Material Type: matType3
Color: green
Summery State: false
Gender: FEMALE

Category: Clothes
Name: Jeans
Brand Name: mavi
Type: Clothes
Sale Price: 260,00
Size: S
Material Type: matType1
Color: blue
Summery State: false
Gender: MALE

```

```

Category: Clothes
Name: Jacket
Brand Name: bershka
Type: Clothes
Sale Price: 299,00
Size: S
Material Type: matType2
Color: blue
Summery State: false
Gender: MALE

```

```

Category: Clothes
Name: Dress
Brand Name: bershka
Type: Clothes
Sale Price: 390,00
Size: M
Material Type: matType3
Color: blue
Summery State: false
Gender: FEMALE

```

```
1 - Add Product To Basket
2 - Request Product
3 - Branch Suggestion For Product That Not Available
4 - Display Sorted
0 - Back To Menu
Choice: 4

1 - Compare by Brand
2 - Compare by Name
3 - Compare by Price
4 - Compare by Type
5 - Compare by Gender
6 - Compare by Kind
7 - Compare by Material Type
Choice: 2
```

```
Category: Clothes
Name: Dress
Brand Name: bershka
Type: Clothes
Sale Price: 390,00
Size: M
Material Type: matType3
Color: blue
Summery State: false
Gender: FEMALE

Category: Clothes
Name: Jacket
Brand Name: bershka
Type: Clothes
Sale Price: 299,00
Size: S
Material Type: matType2
Color: blue
Summery State: false
Gender: MALE

Category: Clothes
Name: Jeans
Brand Name: levi's
Type: Clothes
Sale Price: 169,00
Size: M
Material Type: matType3
Color: green
Summery State: false
Gender: FEMALE
```

```
Category: Clothes
Name: T_shirt
Brand Name: mavi
Type: Clothes
Sale Price: 130,00
Size: L
Material Type: matType1
Color: blue
Summery State: true
Gender: FEMALE
```

- T50

```

1 - Add Product To Basket
2 - Request Product
3 - Branch Suggestion For Product That Not Available
0 - Back To Menu
Choice: 1

```

Enter Product Name: coke
INVALID PRODUCT!

- T51

| ----- Basket of Customer avni ----- | | | | |
|-------------------------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Clothes | mavi | Short | 145,34 | 4 |

```

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

```

Choice: 2

Enter Product Number: 0
PRODUCT MUST BE POSITIVE!

- T52

| ----- Basket of Customer avni ----- | | | | |
|-------------------------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Clothes | mavi | T_shirt | 130,00 | 1 |
| 2 - Drink | çaykur | tea | 16,38 | 45 |
| 3 - Drink | çaykur | tea | 16,38 | 45 |

```

1 - Buy Products
2 - Remove Product From Basket
0 - Back To Menu

```

Choice: 1

Correctness Test Results

Cases:

Test1 -> Tests public methods of structures, and user classes. Display methods.

Test2 -> Tests Graph implementations correctness.

Test3 -> Test Customer's buy method's correctness.

Results:

Test1

```
Creating company
Company Name: company1
Administrator: None
Branches: None

-----
Creating an administrator

User Type: Administrator
Company: company1
Name: admin1
Username: ADMIN
Age: 30
Gender: FEMALE

-----
Company after the administrator creation.

Company Name: company1
Administrator: admin1
Branches: None

-----
Creating a branch

-----
Branch Name: branch1
BranchManager: None
----- Employees -----
  Name          Gender      Age
-----


----- Customers -----
  Name          Gender      Age
-----


----- Products -----
  Product Type   Brand Name   Product Name   Price   Stock
-----
```

Creating a branch manager

User Type: Branch Manager
Branch: branch1
Name: manager1
Username: mngr_1
Age: 30
Gender: MALE

Branch after the manager creation.

Branch Name: branch1
BranchManager: manager1

----- Employees -----
Name Gender Age

----- Customers -----
Name Gender Age

----- Products -----
Product Type Brand Name Product Name Price Stock

Creating an employee and customer.

User Type: Branch Employee
Branch: branch1
Name: elif
Username: Elifabla
Age: 30
Gender: FEMALE

```
-----
Admin adds branch to company. Company after adding branch.

Company Name: company1
Administrator: admin1
Branches: 1 : branch1

-----
Creating a new product.

Category: Clothes
Name: clothes1
Brand Name: brand1
Type: Clothes
Sale Price: 172.38
Size: XL
Material Type: matType1
Color: blue
Summery State: false
Gender: FEMALE

-----
Employee adds product to the branch. Branch after adding product.

-----
Branch Name: branch1
BranchManager: manager1
----- Employees -----
Name          Gender      Age
elif          FEMALE     30

----- Customers -----
Name          Gender      Age
ayse          FEMALE     25

----- Products -----
Product Type   Brand Name   Product Name   Price      Stock
1 - Clothes    brand1      clothes1       172.38     0

-----


Employees add new products.
Customer display products, then adds and removes product to(from) basket and displays basket.

-----
----- Products -----
Product Type   Brand Name   Product Name   Price      Stock
1 - Drink      brand3      tea            16.38     0
2 - Furniture   ikea        furniture1    171.47     0
3 - Clothes     brand2      clothes2       145.34     0
4 - Clothes     brand1      clothes1       172.38     0
5 - Electronic  samsung     phone          1300.00    0
6 - Food        ruffle      crisps         172.38     0
7 - Personal Car blendax    shampoo        98.93     0

----- Basket of Customer ayse -----
Product Type   Brand Name   Product Name   Price      Stock
1 - Clothes    brand1      clothes1       172.38     0
2 - Clothes    brand2      clothes2       145.34     0
3 - Drink       brand3      tea            16.38     0

----- Customer prints order history.

Last Removed Product from Basket: None
Last Added Product to Basket: tea

-----

Branch Manager adds new customers.

-----
Branch Name: branch1
BranchManager: manager1
----- Employees -----
Name          Gender      Age
elif          FEMALE     30
ayse          FEMALE     37
ahmet         MALE       24
burak         MALE       42
emre          MALE       44
irem          FEMALE     31
```

| -- Customers -- | | | | |
|------------------|------------|--------------|---------|-------|
| Name | Gender | Age | | |
| asli | FEMALE | 25 | | |
| ayse | FEMALE | 25 | | |
| -- Products -- | | | | |
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Drink | brand3 | tea | 16.38 | 0 |
| 2 - Furniture | ikea | furniture1 | 171.47 | 0 |
| 3 - Clothes | brand2 | clothes2 | 145.34 | 0 |
| 4 - Clothes | brand1 | clothes1 | 172.38 | 0 |
| 5 - Electronic | samsung | phone | 1300.00 | 0 |
| 6 - Food | ruffle | crisps | 172.38 | 0 |
| 7 - Personal Car | blendax | shampoo | 98.93 | 0 |

| |
|---------------------------|
| Employee removes product. |
|---------------------------|

| Branch Name: branch1 | | |
|-------------------------|--------|-----|
| BranchManager: manager1 | | |
| -- Employees -- | | |
| Name | Gender | Age |
| elif | FEMALE | 30 |
| ayse | FEMALE | 37 |
| ahmet | MALE | 24 |
| burak | MALE | 42 |
| emre | MALE | 44 |
| irem | FEMALE | 31 |

| -- Customers -- | | |
|-----------------|--------|-----|
| Name | Gender | Age |
| asli | FEMALE | 25 |
| ayse | FEMALE | 25 |

| --- Products --- | | | | | |
|------------------|------------|--------------|---------|-------|--|
| Product Type | Brand Name | Product Name | Price | Stock | |
| 1 - Drink | brand3 | tea | 16.38 | 0 | |
| 2 - Furniture | ikea | furniture1 | 171.47 | 0 | |
| 3 - Clothes | brand2 | clothes2 | 145.34 | 0 | |
| 4 - Clothes | brand1 | clothes1 | 172.38 | 0 | |
| 5 - Electronic | samsung | phone | 1300.00 | 0 | |
| 6 - Food | ruffle | crisps | 172.38 | 0 | |
| 7 - Personal Car | blendax | shampoo | 98.93 | 0 | |

Employee removes product.

Branch Name: branch1

BranchManager: manager1

| --- Employees --- | | |
|-------------------|--------|-----|
| Name | Gender | Age |
| elif | FEMALE | 30 |
| ayse | FEMALE | 37 |
| ahmet | MALE | 24 |
| burak | MALE | 42 |
| emre | MALE | 44 |
| irem | FEMALE | 31 |

--- Customers ---

| Name | Gender | Age |
|------|--------|-----|
| asli | FEMALE | 25 |
| ayse | FEMALE | 25 |

| Products | | | | |
|------------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Drink | brand3 | tea | 16.38 | 0 |
| 2 - Furniture | ikea | furniture1 | 171.47 | 0 |
| 3 - Clothes | brand1 | clothes1 | 172.38 | 0 |
| 4 - Electronic | samsung | phone | 1300.00 | 0 |
| 5 - Food | ruffle | crisps | 172.38 | 0 |
| 6 - Personal Car | blendax | shampoo | 98.93 | 0 |

Some customers requested products. One of the employees display requested products.

| Requested Products | |
|--------------------|--------|
| Product Name | Price |
| noodles | 9.88 |
| tshirt | 148.07 |
| cream | 17.94 |

Employee adds requested product according to the priority.

Manager displays products.

| Products | | | | |
|------------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Drink | brand3 | tea | 16.38 | 0 |
| 2 - Furniture | ikea | furniture1 | 171.47 | 0 |
| 3 - Clothes | brand1 | clothes1 | 172.38 | 0 |
| 4 - Electronic | samsung | phone | 1300.00 | 0 |
| 5 - Food | brand5 | noodles | 13.00 | 0 |
| 6 - Food | ruffle | crisps | 172.38 | 0 |
| 7 - Personal Car | blendax | shampoo | 98.93 | 0 |

Manager removes an employee, and displays employees.

| ----- Employees of branch1 ----- | |
|----------------------------------|-------------------|
| Employee Name | Employee Username |
| elif | Elifabla |
| ayse | aysee |
| ahmet | Ahmetinho |
| emre | Mr.shutterspeed |
| irem | Irem_y |

Admin adds new branches, and displays branches.

| ----- Branches ----- | |
|----------------------|----------------|
| Branch Name | Branch Manager |
| branch1 | manager1 |
| branch2 | manager2 |
| branch3 | manager3 |

Admin displays employees, and customers.

| ----- Employees of company1 ----- | | |
|-----------------------------------|---------------|-------------------|
| Branch Name | Employee Name | Employee Username |
| branch1 | elif | Elifabla |
| branch1 | ayse | aysee |
| branch1 | ahmet | Ahmetinho |
| branch1 | emre | Mr.shutterspeed |
| branch1 | irem | Irem_y |
| branch2 | osman | Osmanpzs |
| branch3 | sevim | Sevim7 |

| ----- Customers of company1 ----- | | |
|-----------------------------------|---------------|-------------------|
| Branch Name | Customer Name | Customer Username |
| branch1 | asli | aslili |
| branch1 | ayse | a/ayse58 |

| ----- Customers of company1 ----- | | |
|-----------------------------------|---------------|-------------------|
| Branch Name | Customer Name | Customer Username |
| branch1 | asli | aslippi |
| branch1 | ayse | a.ayse58 |

Admin removes one of the branches, and displays.

| ----- Branches ----- | |
|----------------------|----------------|
| Branch Name | Branch Manager |
| branch1 | manager1 |
| branch3 | manager3 |

----- DONE -----

Test2

| ----- TEST 2 ----- | |
|----------------------|----------------|
| ----- Branches ----- | |
| Branch Name | Branch Manager |
| branch1 | manager1 |
| branch2 | manager2 |
| branch3 | manager3 |
| branch4 | manager4 |
| branch5 | manager5 |
| branch6 | manager6 |
| branch7 | manager7 |
| branch8 | manager8 |
| branch9 | manager9 |
| branch10 | manager10 |
| branch11 | manager11 |
| branch12 | manager12 |
| branch13 | manager13 |
| branch14 | manager14 |
| branch15 | manager15 |

Before inserting edges:

```

Adjacency List
vertex (id = 1, branch name = branch1)-->
vertex (id = 2, branch name = branch2)-->
vertex (id = 3, branch name = branch3)-->
vertex (id = 4, branch name = branch4)-->
vertex (id = 5, branch name = branch5)-->
vertex (id = 6, branch name = branch6)-->
vertex (id = 7, branch name = branch7)-->
vertex (id = 8, branch name = branch8)-->
vertex (id = 9, branch name = branch9)-->
vertex (id = 10, branch name = branch10)-->
vertex (id = 11, branch name = branch11)-->
vertex (id = 12, branch name = branch12)-->
vertex (id = 13, branch name = branch13)-->
vertex (id = 14, branch name = branch14)-->
vertex (id = 15, branch name = branch15)-->
```

After inserting edges:

```

Adjacency List
vertex (id = 1, branch name = branch1)-->
  [(1, 2): 70.0]
  [(1, 3): 90.0]
  [(1, 4): 110.0]
  [(1, 5): 130.0]
  [(1, 6): 150.0]
  [(1, 7): 170.0]
  [(1, 8): 190.0]
  [(1, 9): 210.0]
  [(1, 10): 230.0]
  [(1, 11): 250.0]
  [(1, 12): 270.0]
  [(1, 13): 290.0]
  [(1, 14): 310.0]
  [(1, 15): 330.0]
vertex (id = 2, branch name = branch2)-->
  [(2, 1): 70.0]
  [(2, 3): 5.0]
  [(2, 5): 10.0]
  [(2, 15): 80.0]
vertex (id = 3, branch name = branch3)-->
  [(3, 1): 90.0]
  [(3, 2): 5.0]
  [(3, 6): 13.0]
  [(3, 7): 12.0]
vertex (id = 4, branch name = branch4)-->
  [(4, 1): 110.0]
  [(4, 11): 28.0]
vertex (id = 5, branch name = branch5)-->
  [(5, 1): 130.0]
  [(5, 2): 10.0]
  [(5, 6): 5.0]
vertex (id = 6, branch name = branch6)-->
  [(6, 1): 150.0]
  [(6, 3): 13.0]
  [(6, 5): 5.0]
  [(6, 9): 25.0]
vertex (id = 7, branch name = branch7)-->
  [(7, 1): 170.0]
  [(7, 3): 12.0]
vertex (id = 8, branch name = branch8)-->
  [(8, 1): 190.0]
  [(8, 13): 20.0]
vertex (id = 9, branch name = branch9)-->
  [(9, 1): 210.0]
  [(9, 6): 25.0]
  [(9, 13): 14.0]
vertex (id = 10, branch name = branch10)-->
  [(10, 1): 230.0]
vertex (id = 11, branch name = branch11)-->
  [(11, 1): 250.0]
  [(11, 4): 28.0]
  [(11, 12): 6.0]
  [(11, 14): 9.0]
vertex (id = 12, branch name = branch12)-->
  [(12, 1): 270.0]
  [(12, 11): 6.0]
  [(12, 15): 16.0]
vertex (id = 13, branch name = branch13)-->
  [(13, 1): 290.0]
  [(13, 8): 20.0]
  [(13, 9): 14.0]
vertex (id = 14, branch name = branch14)-->
  [(14, 1): 310.0]
  [(14, 11): 9.0]
  [(14, 15): 7.0]
vertex (id = 15, branch name = branch15)-->
  [(15, 1): 330.0]
  [(15, 2): 80.0]
  [(15, 14): 7.0]
  [(15, 12): 16.0]
```

Admin displays customers.

| ----- Customers of company1 ----- | | |
|-----------------------------------|---------------|-------------------|
| Branch Name | Customer Name | Customer Username |
| branch1 | ayse | a/ayse58 |

Customer displays products.

| ----- Products ----- | | | | |
|----------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Furniture | ikea | furniture1 | 171.47 | 0 |
| 2 - Clothes | brand1 | clothes1 | 172.38 | 0 |
| 3 - Personal Car | nivea | cream | 17.94 | 0 |

Every employee of each branch displays products.

Branch: branch1

| ----- Products ----- | | | | |
|----------------------|------------|--------------|--------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Furniture | ikea | furniture1 | 171.47 | 0 |
| 2 - Clothes | brand1 | clothes1 | 172.38 | 0 |
| 3 - Personal Car | nivea | cream | 17.94 | 0 |

Branch: branch2

| Products | | | | |
|--------------|------------|--------------|-------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| | | | | |
| | | | | |

Branch: branch3

| Products | | | | |
|--------------|------------|--------------|-------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| | | | | |
| | | | | |

Branch: branch4

| Products | | | | |
|--------------|------------|--------------|-------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| | | | | |
| | | | | |

Branch: branch5

| Products | | | | |
|--------------|------------|--------------|-------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| | | | | |
| | | | | |

Branch: branch6

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch7

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch8

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch9

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch10

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch11

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch12

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch13

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch14

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Branch: branch15

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |

Test3

| TEST 3 | | | | |
|--|--|--|--|--|
| Branch after new products, employee and customer added. | | | | |
| Branch Name: branch1 BranchManager: manager1 | | | | |
| Employees | | | | |
| Name Gender Age | | | | |
| elif FEMALE 30 | | | | |
| Customers | | | | |
| Name Gender Age | | | | |
| ayse FEMALE 25 | | | | |
| Products | | | | |
| Product Type Brand Name Product Name Price Stock | | | | |
| 1 - Drink dogus tea 16.38 10 | | | | |
| 2 - Clothes mavi blouse 96.20 6 | | | | |
| 3 - Clothes zara tshirt 145.34 7 | | | | |
| 4 - Electronic samsung phone 1300.00 0 | | | | |

Customer display products, then adds product to basket and displays basket.

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Drink | dogus | tea | 16.38 | 10 |
| 2 - Clothes | mavi | blouse | 96.20 | 6 |
| 3 - Clothes | zara | tshirt | 145.34 | 7 |
| 4 - Electronic | samsung | phone | 1300.00 | 0 |

| Basket of Customer ayse | | | | |
|-------------------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Electronic | samsung | phone | 1300.00 | 0 |
| 2 - Clothes | mavi | blouse | 96.20 | 6 |
| 3 - Clothes | zara | tshirt | 145.34 | 7 |

Customer purchases products in the basket.

| Basket of Customer ayse | | | | |
|-------------------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| Electronic | samsung | phone | 1300.00 | 0 |
| OUT OF STOCK! | | | | |
| Clothes | mavi | blouse | 96.20 | 6 |
| Clothes | zara | tshirt | 145.34 | 7 |

Total Amount : 241.54

Product stocks after purchase.

| Products | | | | |
|----------------|------------|--------------|---------|-------|
| Product Type | Brand Name | Product Name | Price | Stock |
| 1 - Drink | dogus | tea | 16.38 | 10 |
| 2 - Clothes | mavi | blouse | 96.20 | 5 |
| 3 - Clothes | zara | tshirt | 145.34 | 6 |
| 4 - Electronic | samsung | phone | 1300.00 | 0 |

DONE

Theoretical Analysis of the Implementation

Company:

- **Company(String):** $\Theta(1)$
- **getCompanyName():** Since it is getter $\Theta(1)$
- **getBranches():** Since it is getter $\Theta(1)$
- **getAdministrator():** Since it is getter $\Theta(1)$
- **setAdministrator/Administrator):** Since it is assignment $\Theta(1)$

Branch:

- **Branch(String):** $\Theta(1)$
- **getBranchName():** Since it is getter $\Theta(1)$
- **setBranchManager(BranchManager):** Since it is setter $\Theta(1)$
- **getBranchManager():** Since it is getter $\Theta(1)$
- **getBranchEmployees():** Since it is getter $\Theta(1)$
- **getBranchCustomers():** Since it is getter $\Theta(1)$
- **getProducts():** Since it is getter $\Theta(1)$
- **hasProduct(Product):** Products are held in HashMap, accessing it is averagely constant time, values are holded in TreeSet since TreeSet is a balanced tree, getting value is $O(\log n)$.
- **getStringEmployees():** $\Theta(n)$, n is the number of employees.
- **getStringCustomers():** $\Theta(n)$, n is the number of customers.
- **getStringProducts():** $\Theta(n)$, n is number of customers.

Product:

- **Product(String, String, String, double):** $\Theta(1)$
- **getSalePrice():** Since it is getter $\Theta(1)$
- **setEntryPrice(double):** Since it is setter $\Theta(1)$
- **getEntryPrice():** Since it is getter $\Theta(1)$
- **setName(String):** Since it is setter $\Theta(1)$
- **getName():** Since it is getter $\Theta(1)$
- **setBrand(String):** Since it is setter $\Theta(1)$
- **getBrand():** Since it is getter $\Theta(1)$
- **setType(String):** Since it is setter $\Theta(1)$
- **getStock():** Since it is getter $\Theta(1)$
- **setStock(int):** Since it is setter $\Theta(1)$
- **decreaseStock():** $\Theta(1)$
- **getType():** Since it is getter $\Theta(1)$

User:

- **User(String, int, Gender):** $\Theta(1)$
- **setName(String):** Since it is setter $\Theta(1)$
- **getName():** Since it is getter $\Theta(1)$
- **setAge(int):** Since it is setter $\Theta(1)$
- **getAge():** Since it is getter $\Theta(1)$
- **setGender(Gender):** Since it is setter $\Theta(1)$
- **getGender():** Since it is getter $\Theta(1)$

Administrator:

- **Administrator(String, int, Gender, Company):** $\Theta(1)$
- **setCompany(Company):** $\Theta(1)$
- **getCompany():** $\Theta(1)$
- **addBranch(Branch):** $O(1)$, Branches are held in a dynamic graph. Graph implementation have edges, and vertices in HashMap. Inserting to a Hash Map is averagely constant time.
- **removeBranch(Branch):** $O(|E|)$, removing the branch's time complexity is dependent on the number of connections. If there is no connection, or number of connection(edge) is close to number of vertices, time complexity becomes $O(|V|)$, v is the number of branches.
- **setBranchManager(Branch, BranchManager):** $O(n)$, because it traverses the branches with branch iterator, which are vertices. Vertices are branches. Iterator is $O(n)$, because HashMap is LinkedHashMap, therefore it is $O(n)$.
- **connectBranches(Branch, Branch, double):** $O(|V|)$, because in DynamicGraph's implementation, addEdge method takes id's, so that firstly it must find each branch's id's. To find that vertices must be traversed and it is linear with the number of branches.
- **connectBranches(int, int, double):** $\Theta(1)$, it is averagely constant. Because directly inserts the edge to HashMap.
- **removeConnection(int, int):** It takes $O(m+n)$ times, because the graph is an adjacency list and branches are held in an undirected graph. m is the number of edges that first id have, n is the other.
- **displayBranchGraph():** It takes $\Theta(|E|)$ times, because whole edges are traversed.
- **displayBranches():** $\Theta(n)$, it is linear because of traversing with iterator.
- **displayBranchManagers():** $\Theta(n)$, it is linear because, whole dataset traversed with iterator
- **displayBranchEmployees():** $\Theta(n)$, it is linear because, whole dataset traversed with iterator
- **displayCustomers():** $\Theta(n)$, it is linear because, whole dataset traversed with iterator.

Branch Employee:

- **BranchEmployee(String, int, Gender, Branch):** $\Theta(1)$
- **addProduct(Product):** $O(\log n)$, n is the number of products which are the same with a given product's type. Getting products is a constant time because, products are held in HashMap, after getting products according to given type, insertion to the red black tree is $O(\log n)$.

- **addProduct(Product, int):** $O(\log n)$, n is the number of Product which are the same with a given product's type. It makes removal, contains and add operations consecutively to a balanced tree, all of them is $O(\log n)$ because of the structure of the red black tree. It is the same with other addProduct according to the notations, but this takes a longer time because of the constants that we ignore in the notation.
- **removeProduct(Product):** $O(\log n)$, n is the number of Product which are the same with given product's type. Removing from the red black tree is $O(\log n)$.
- **removeProduct(Product, int):** $O(\log n)$, n is the number of Product which are the same with given product's type. It makes contains, removal, add operations consecutively to a balanced tree, all of them is $O(\log n)$ because of the structure of the red black tree. It is the same with other removeProduct according to the notations, but this takes a longer time because of the constants that we ignore in the notation.
- **findProductByName(String):** $O(n)$, n is the total number of products. Searches the whole map with an iterator.
- **addRequestedProducts():** $O(\log n)$, n is the number of Product which are the same with given product's type. Polling from Priority Queue is a constant time operation. It delegates the addProduct, and it is $O(\log n)$.
- **displayProducts():** $\Theta(n)$, it is linear because, whole dataset traversed with iterator.
- **displayRequestedProducts():** $\Theta(n)$, it is linear because, whole dataset traversed with iterator.
- **getBranch():** $\Theta(1)$

Branch Manager:

- **BranchManager(String name, int age, Gender gender, Branch branch):** $\Theta(1)$
- **setBranch(Branch):** $\Theta(1)$
- **getBranch():** $\Theta(1)$
- **displayBranches(Company):** It is $\Theta(n)$, m is the total number of the branches in given company.
- **displayBranchEmployees():** $\Theta(n)$, it is linear because, whole dataset traversed with iterator.
- **addCustomer(Customer):** $O(\log n)$, inserting to a Skip List takes logarithmic time averagely.
- **removeCustomer(Customer):** $O(\log n)$, deleting from Skip List takes logarithmic time averagely.
- **addBranchEmployee(BranchEmployee):** $O(\log n)$. Insertion to a balanced tree takes logarithmic time.
- **removeBranchEmployee(BranchEmployee):** $O(\log n)$. Removal from a balanced tree takes logarithmic time.

Customer:

- **Customer(String, int, Gender, Branch):** $\Theta(1)$
- **getBranchSuggestion(Company, Product):** $O(|V|^2 + |V|\log n)$, $|V|$ is the number of branches, n is the number of products for a given product type. V^2 comes from the dijkstra algorithm, n is there because of the hasProduct method. In each iteration of the loop, this hasProduct is called.

- **addProductToBasket(Product):** $O(n)$, ArrayList access is constant, insertion to linked list is $O(n)$.
- **removeProductFromBasket(Product):** $O(n)$, ArrayList access is constant, removal from linked list is $O(n)$.
- **requestProduct(Product):** $O(\log n)$, inserting binary heap is $O(\log n)$.
- **displayProducts():** $\Theta(n)$, it is linear because, whole dataset traversed with iterator.
- **displayBasket():** $O(n)$, n is the total number of products in the basket.
- **purchaseBasket():** $O(n)$, n is the total number of products in the basket.
- **printOrderHistory():** $\Theta(1)$, because it uses only peek operation.
- **getBranch():** $\Theta(1)$

Performance Analysis of the Implementation

Performance test for 100 dataset.

```

average of inserting 100 employee = 61236ns
average of inserting 100 customer = 92267ns
average of inserting 100 product = 10938ns
average of requesting 100 product = 2887ns
addRequestedProducts() = 374ns
remove middle employee = 355ns
remove middle customer = 1509ns
remove first employee = 45ns
remove first customer = 204ns
remove last employee = 112ns
remove last customer = 181ns
inserting one employee to full = 358ns
inserting one employee to full = 646ns
inserting one product to full = 179ns
requesting one product to full = 77ns
-----
```

Performance test for 1000 dataset.

```

average of inserting 1000 employee = 1777ns
average of inserting 1000 customer = 7127ns
average of inserting 1000 product = 2386ns
average of requesting 1000 product = 1027ns
addRequestedProducts() = 81ns
remove middle employee = 7ns
remove middle customer = 87ns
remove first employee = 27ns
remove first customer = 32ns
remove last employee = 15ns
remove last customer = 42ns
inserting one employee to full = 127ns
inserting one employee to full = 144ns
inserting one product to full = 51ns
requesting one product to full = 50ns
-----
```

Performance test for 10000 dataset.

```

average of inserting 10000 employee = 1358ns
average of inserting 10000 customer = 2388ns
average of inserting 10000 product = 1390ns
average of requesting 10000 product = 587ns
addRequestedProducts() = 2ns
remove middle employee = 0ns
remove middle customer = 4ns
remove first employee = 0ns
remove first customer = 2ns
remove last employee = 1ns
remove last customer = 1ns
inserting one employee to full = 11ns
inserting one employee to full = 7ns
inserting one product to full = 1ns
requesting one product to full = 0ns
-----
```

Thank You For Your Time...