**GTU Department of Computer Engineering**

**CSE 222/505 – Spring 2021**

**Project Report 2**

**Group-9**

**141024008**

**AHMET DÖNMEZ**

**1901042260**

**BURAK ÇİÇEK**

**1801042628**

**FURKAN ÇELEN**

**141044082**

**HATİCE ARGUN**

**1801042667**

**MEHDİ KURTCEBE**

**200104004095**

**MEHMET HÜSEYİN YILDIZ**

**1901042680**

**SENA ERDOĞAN**

Warehouse Automation

**Fixes From Report 1**

* **Use case diagrams combined in one piece.**
* **Users of the system detailed.**
* **Database was mentioned as a fault in first report feedback, instead of using single database class we separated each database in related classes by using data structures as static variables, for the runtime stage only.**

**Problem Definition**

This warehouse automation application provides good features for companies that needs fast and simple UI for their product supplying. There are many supermarket chains that keep track of their stocks manually and this causes conflicts between the warehouse and branches. Some of these problems are inconsistencies in registration of the products, products being delivered later than expected, quantities of the products in the branches not matching the warehouse’s stock, since the stock is not trackable, products cannot be delivered to the branches at the predicted time.

It is aimed to eliminate these problems for companies with the automation system we will prepare.

**Users of the System**

* Warehouse Administrator can add and remove employees, branches, and products.
* Warehouse employee can see product requests from branches and accept or reject them, send products to branches, inform administrator for out of stocks in the warehouse.
* Branch employee can make sales and request product from the warehouse.
* Transportation employee can update shipment status.

**Functional Requirements**

|  |  |
| --- | --- |
| Requirement ID | Description of the Requirement |
| FR1 | The system shall assign roles to users. |
| FR2 | The system shall provide one admin user initially. |
| FR3 | The system shall let users to login to the system with their information. |
| FR4 | The system shall let visitors who do not have an account to sign up to system with the approval of the administrator. |
| FR5 | The system shall store users’ passwords encrypted. |
| FR6 | The system shall let employees to change their own password and name. |
| FR7 | The system shall let administrator to change all personnel’s passwords and names. |
| FR8 | The system shall let administrator to addition and deletion and searching of employees. |
| FR9 | The system shall let administrator to addition, deletion and searching of branches. |
| FR10 | The system shall let administrator to addition, deletion and searching of products. |
| FR11 | The system shall provide product categorization. |
| FR12 | The system shall provide the uniqueness of each product in the warehouse. |
| FR13 | The system shall inform branch employees in case of stock shortage of products in the branch. |
| FR14 | The system shall inform warehouse employee in case of stock shortage of products in the warehouse. |
| FR15 | The system shall let branch employees to request products from warehouse. |
| FR16 | The system shall let warehouse employees to accept or reject branch requests. |
| FR17 | The system shall provide to inform branch employee in case of rejection and acceptance of branch requests. |
| FR18 | The system shall let warehouse employee to send shipments to a branch. |
| FR19 | The system shall provide warehouse employees to automatic selection of couriers for shipment operations. |
| FR20 | The system shall let administrator and warehouse employees to see shipments. |
| FR21 | The system shall let branch employee to see products transferred to his/her branch. |
| FR22 | The system shall let transportation employees to see his/her shipment. |
| FR23 | The system shall let transportation employees to update his/her shipment status. |
| FR24 | The system shall let warehouse employees to see all products in the warehouse. |
| FR25 | The system shall let warehouse employees to inform administrator for out of stocks. |
| FR26 | The system shall let branch employees to see all products in the branch. |
| FR27 | The system shall let branch employees to make sales. |

**Non-functional Requirements**

**Utility**

* The system shall be user friendly.
* The system shall be constructed by a trained administrator firstly then can be used by non-expert personal.
* The system UI must be not complex for users.

**Reliability**

* The system should give error messages against the illogical process attempts.
* The system should provide error handling mechanisms to prevent system crashes.

**Performance**

* The system would have very little amount of data at the beginning, but as the users use the system, the amount of data in the system increases and here, the system needs to increase efficiency by using fast algorithms on top of these data. The system targets this in terms of efficiency.

**Space requirements**

* The system will store old users' data on the server side. These are the login information of the users, products, requests that were created by the users. Since, all these data tend to be enlarged as the new users register the system, and the users make use of the system. Therefore, the system should be scalable.

**Supportability**

* The system should be portable, so moving from one OS to another OS shall not create any problem.

**User Interface**

* All the user operations will be done through any standard PC and user will not need another hardware.

**Domain Requirements**

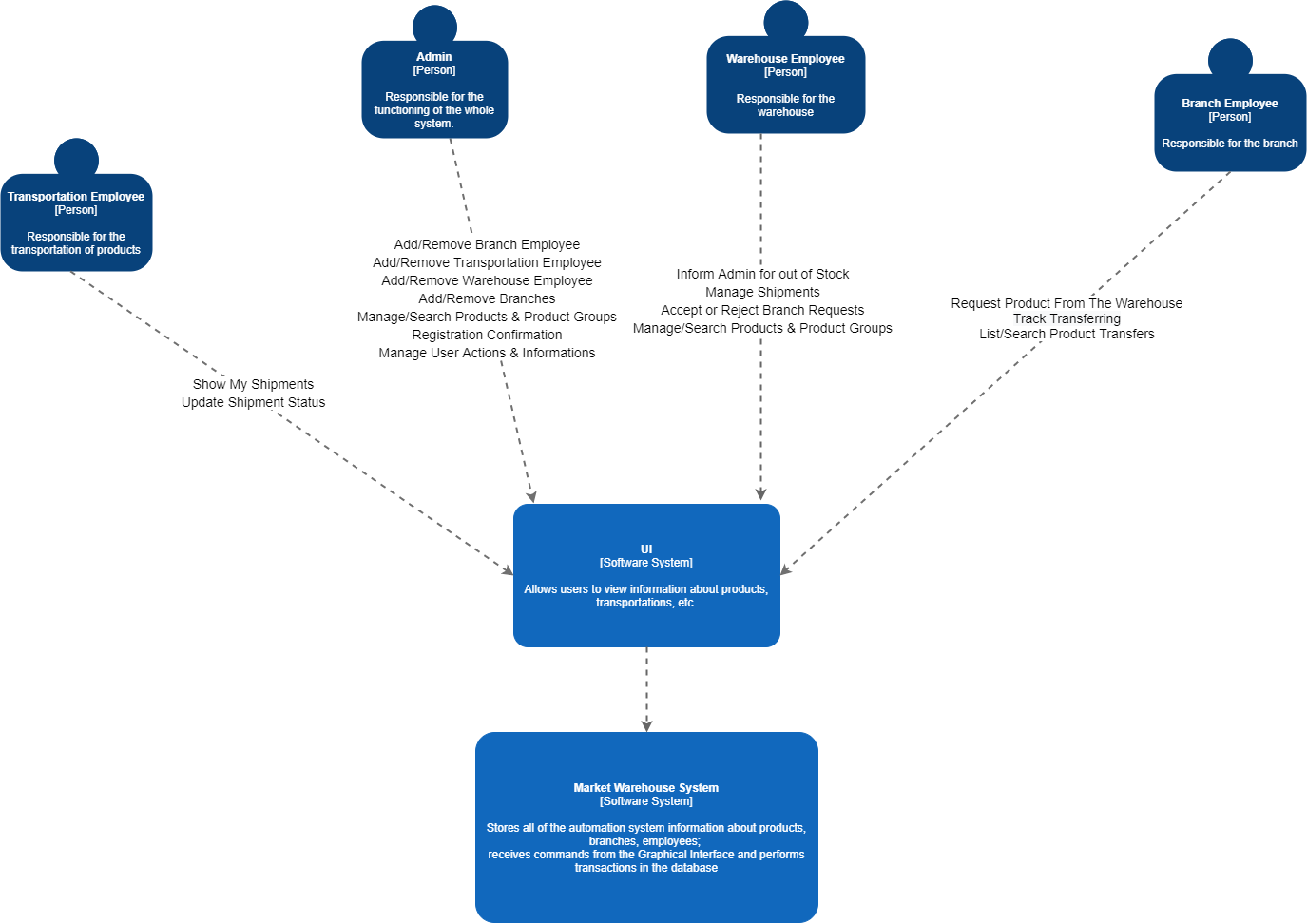
* The system server shall work on Windows, Linux –Ubuntu and Mac Os.

**Use Case Diagram**

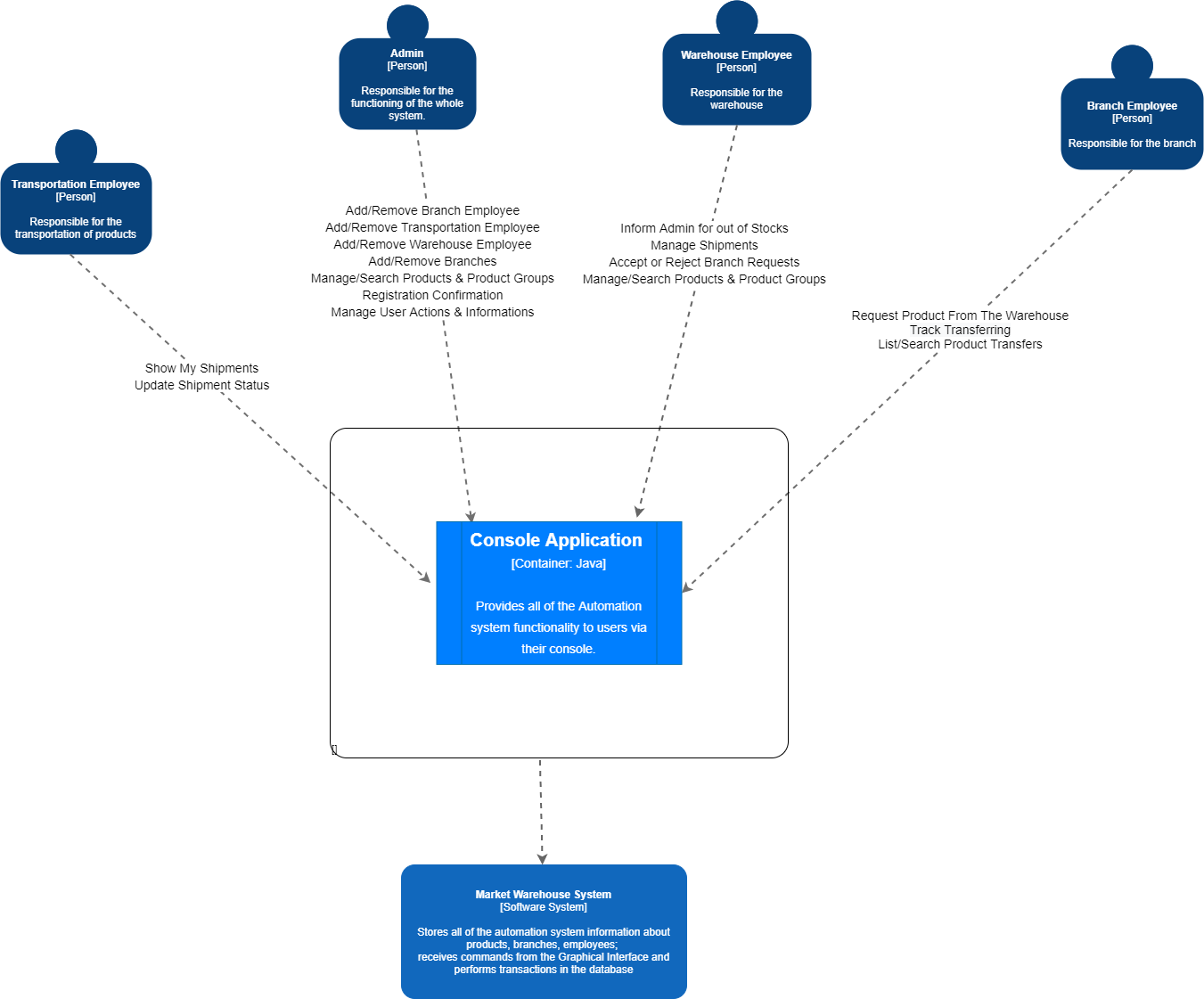


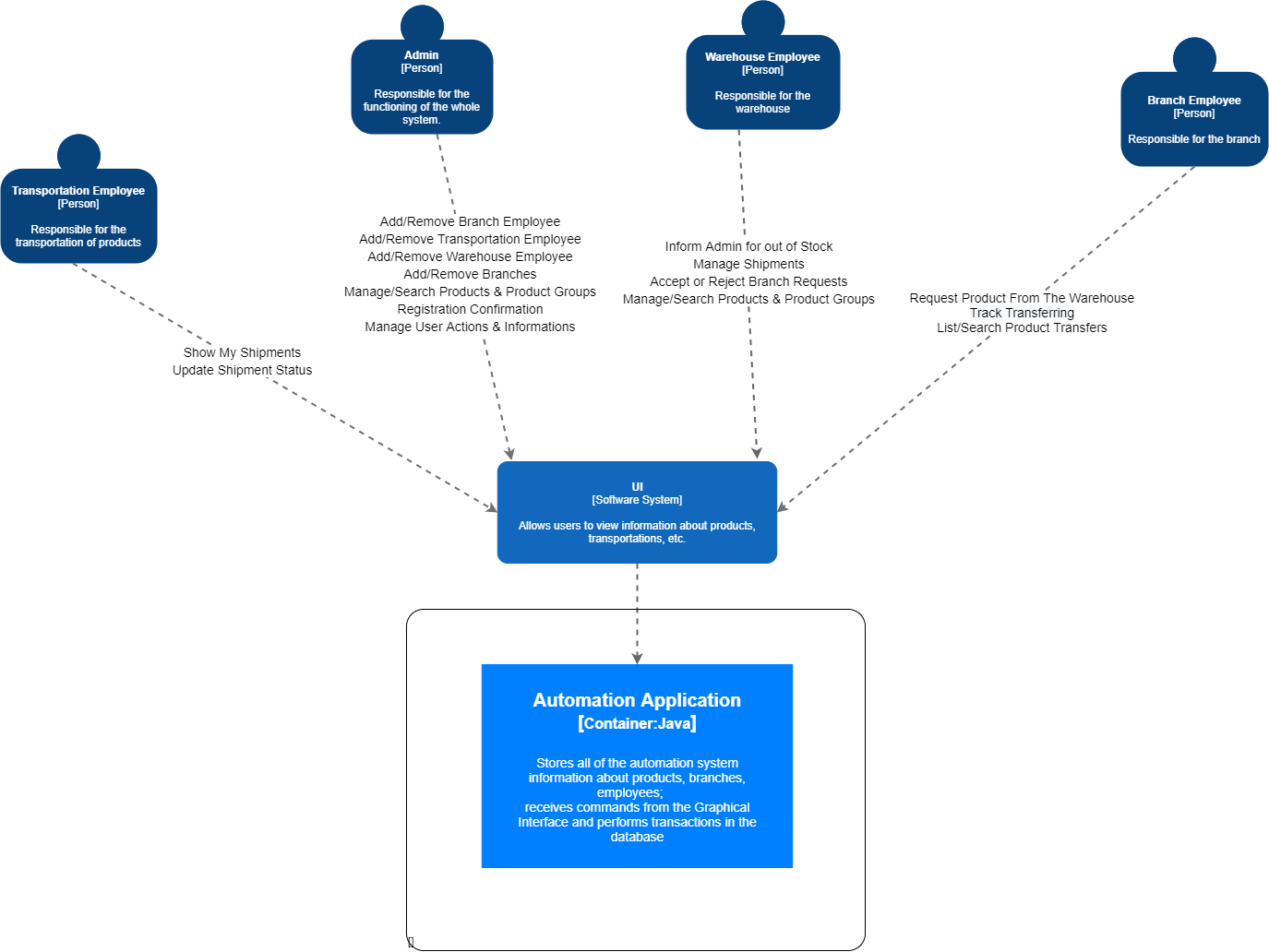
**C4 Diagrams**

**Context Diagram**

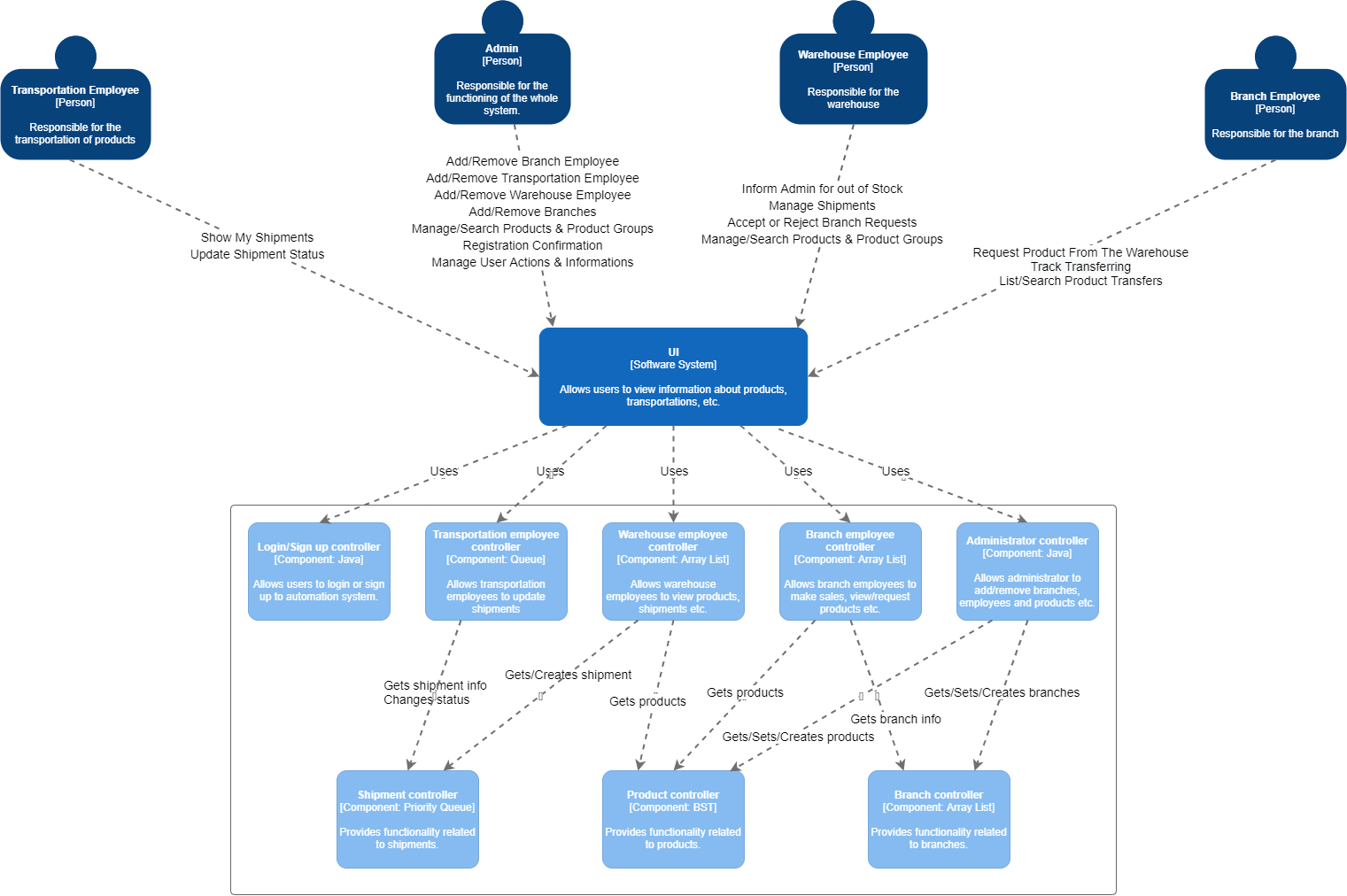
****

**Container Diagram**

****

****

**Component Diagram**

****

**Data Structures Used for Components**

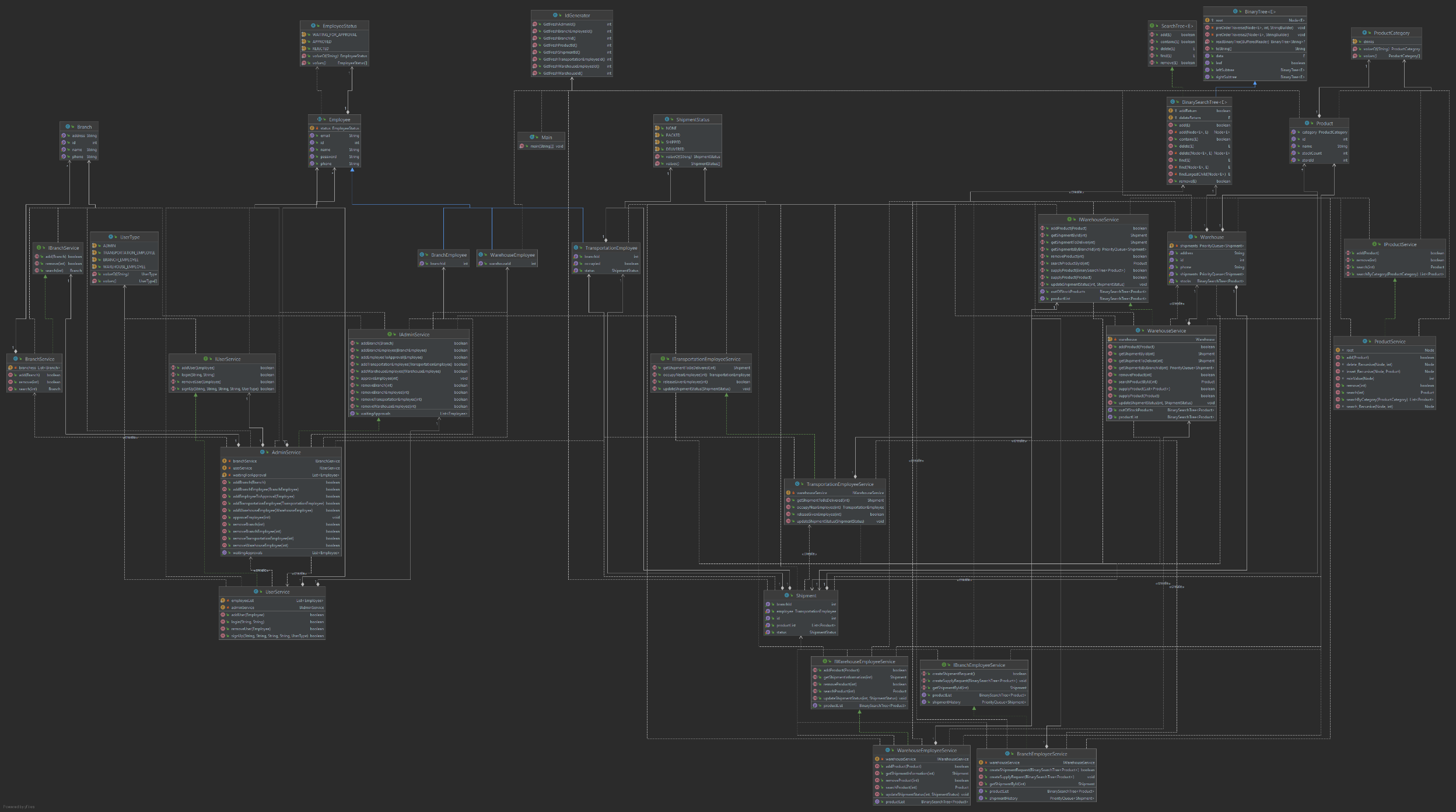
To assign shipments to transportation employees automatically and fairly, we decided to store transportation employees in queue data structure.

We predicted that users would make many searches on products, so we decided to store products in binary search tree data structure to make searches fast.

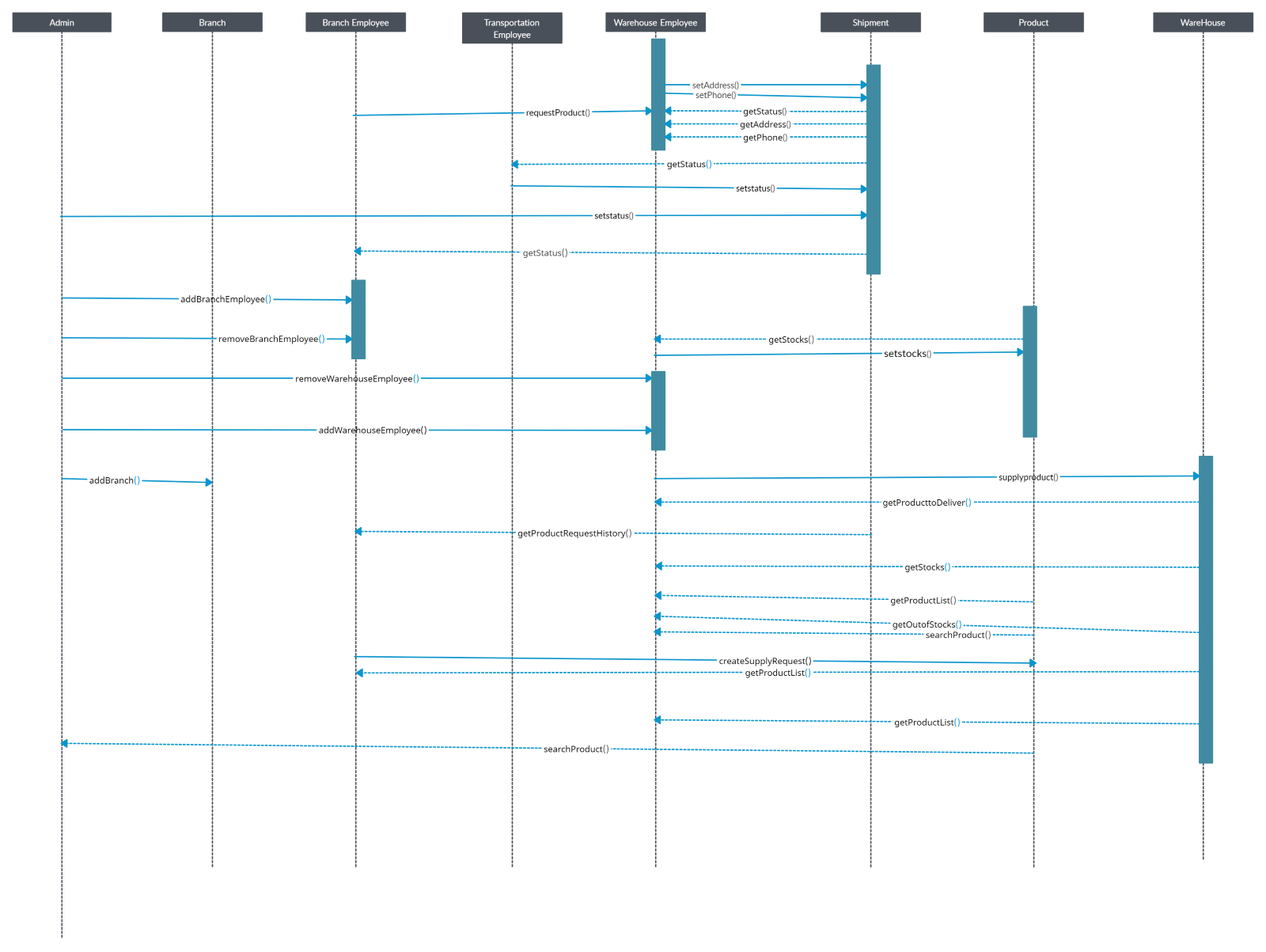
Some shipments may be urgent. They may need to be delivered immediately. To provide this, we decided to store shipments in priority queue data structure.

Other employees, and branches are stored in array list data structure to get them fast and store efficiently.

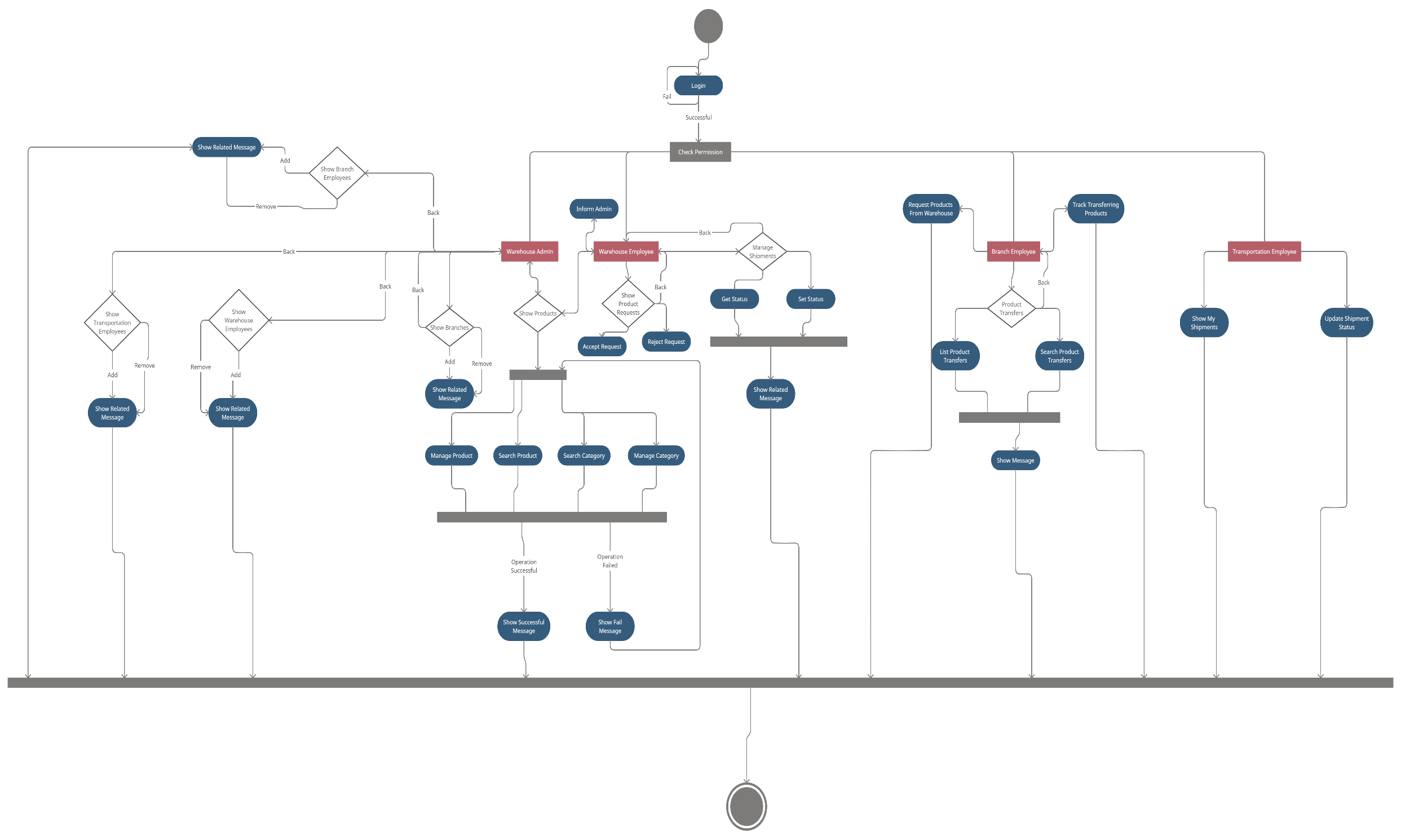
**Class Diagram**



**Sequence Diagram**

****

**Activity Diagram**



**Implementation Details**

**BranchEmployeeService**

* boolean createShipmentRequest method takes a list of Product objects as input and requests these products from the warehouse using WarehouseService class's supplyProduct method which also takes a list of Product objects as input and creates a shipment for this request.
* BinarySearchTree<Product> getProductList method goes through all the products using WarehouseService's getProductList method and checks their store id information. If a product's store id matches current store's, it is added to a BinarySearchTree and it is returned at the end of this method. Throws an IllegalStateException if there are no products in the current branch (if the size of the BinarySearchTree is 0).
* Shipment getShipmentById method takes a shipment id as input and uses WarehouseService’s getShipmentsByBranchId method to go through the list of Shipment objects to find the Shipment matching with the same shipment id and returns that Shipment. Throws NoSuchElementException if the shipment does not exist.
* PriorityQueue<Shipment> getShipmentHistory methodcalls WarehouseService's getShipmentsByBranchId method and returns the PriorityQueue that method returns. Throws IllegalStateException if PriorityQueue's size is 0.
* void createSupplyRequest method takes a BinarySearchTree of products and passes it to WarehouseService's supplyProduct method.

**ProductService**

* This class also contains a BST implementation that has nodes. Nodes stores Products and a key (keys equal to Product Id’s.) but I will not detail this implementation because it is standart BST implementation.
* boolean add methods takes a Product as a parameter to insert a Product recursively to Binary tree depends on productId. Product class stores productId variable and also stores stockId, name, category and stockCount. Methods returns true if addition completes successfully.
* boolean remove method also remove production method from ProductService class. It takes product Id as input and remove the node that stores related product Id. If node is empty, nothing happens also. After deletion operation, BST nodes always reposition itself. Returns true if operation completes successfully.
* public Product search method takes a product Id as input after that search this Id and trying to find Id related product in Binary Tree. Returns product if operation completes successfully. Otherwise, returns null with an error message.

**UserService**

* login method takes Email and password as String and returns whether mail and password match or not.
* signUp method takes name, phone number, e-mail and password as String and takes userType. Registers it to the database.
* addUser method takes an employee instance and Registers to the database.
* removeUser method takes an employee instance and deletes it from the database.

**WarehouseEmployeeService**

This class contains the implementation of basic job descriptions for warehouse employee. It uses the Warehouse Service for the operations it will provide. As can be seen throughout the project, the operations to be performed by all employees have been abstracted and each one has been defined separately.

* updateShipmentStatus: This method gets the shipment id and gets the Shipment status. Updates the corresponding shipment status.
* getShipmentInformation: This method checks whether it is on the system by getting the shipment id, and returns it if the corresponding shipment is in the system.
* getProductList: This method fetches all shipments in warehouse. Thus, employee can access the product list.
* searchProduct: This method takes the product id and checks if this product is available at warehouse. If there is, the related product will be returned.
* addProduct: Adds this method to the given productu warehouse
* removeProduct: This method takes the product id and deletes it if it is in warehouse.

**AdminService**

* addBranch method takes a Branch object and adds it to the system using add method of the BranchService. In a similar way, removeBranch method takes the id of the branch and removes that branch using remove method of the BranchService.
* addBranchEmployee, addTransportationEmployee, addWarehouseEmployee and removeBranchEmployee, removeTransportationEmployee, removeWarehouseEmployee methods work in a similar way. They take relevant objects and add or remove it to the system using addUser and removeUser method of the UserService.
* This service stores waiting sign up approvals. getWaitingApprovals method gets all approvals.
* approveEmployee method takes a user id and approves the user with given id.

**BranchService**

* add method takes a Branch object as a parameter and adds it to the system. Returns true if operation is successful, otherwise false.
* remove method takes the id of the branch as a parameter and removes that branch from the system. Returns true if operation is successful, otherwise false.
* search method searches for the branch with given branch id in branches. Returns branch object if operation is succesfull, otherwise null.

**WarehouseService**

* warehouse class does general operations for the shipment and warehouse associated employees. It uses priorityqueue to store the shipments.
* supplyProduct method gets a product and adds it to the shipments if given product is not null.
* getoutofstockproducts is gives products that are not available in the warehouse currently.
* getproductlist method gives list of products that are in the warehouse.
* getshipment by branch id is a method for finding shipment with given branch id, it is like a helper, it returns shipments that associated with the given branch id. Throws exception if given branch id is not valid or shipments are not available on the warehouse.
* Getspihmenttodeliver method is returns shipment with finding by given branch id, this method is for the transportation employee.

**ShipmentService**

* getid method is for identifying shipment, and returns shipment’s own id, for recognizing and determining shipments easily.
* getEmployee method returns transportation employee that will transport current shipment.
* getProductlist returns list of the products that will ship on this shipment instance.
* setProductlist gets list of the products that will ship on this shipment instance and arranges the shipment’s products.
* getStatus method gives information about the shipment’s current conditions.
* setStatus method is for the setting the current situation of the shipment. That is useful for the communication between personnel.
* setEmployee methods gets transportation employee and assigns it to the class’s own transportation employee variable.

**Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Requirement ID | Test Scenario | Test Steps | Test Data | Expected Results | Actual Results | Pass/Fail |
| T01 | **FR3** | Check administrator login with valid data. | 1.Go to software  2.Enter username  3.Enter password  4.Click login |  | User should login into system as an administrator. |  |  |
| T02 | **FR3** | Check administrator login with invalid username. | 1.Go to software  2.Enter invalid username  3.Enter password  4.Click login |  | System should give an error message and should not login. |  |  |
| T03 | **FR3** | Check administrator login with invalid password. | 1.Go to software  2.Enter username  3.Enter invalid password  4.Click login |  | System should give an error message and should not login. |  |  |
| T04 | **FR3** | Check warehouse employee login with valid data. | 1.Go to software  2.Enter username  3.Enter password  4.Click login |  | User should login into system as a warehouse employee. |  |  |
| T05 | **FR3** | Check warehouse employee login with invalid username. | 1.Go to software  2.Enter invalid username  3.Enter password  4.Click login |  | System should give an error message and should not login. |  |  |
| T06 | **FR3** | Check warehouse employee login with invalid password. | 1.Go to software  2.Enter username  3.Enter invalid password  4.Click login |  | System should give an error message and should not login. |  |  |
| T07 | **FR3** | Check branch employee login with valid data. | 1.Go to software  2.Enter username  3.Enter password  4.Click login |  | User should login into system as a branch employee. |  |  |
| T08 | **FR3** | Check branch employee login with invalid username. | 1.Go to software  2.Enter invalid username  3.Enter password  4.Click login |  | System should give an error message and should not login. |  |  |
| T09 | **FR3** | Check branch employee login with invalid password. | 1.Go to software  2.Enter username  3.Enter invalid password  4.Click login |  | System should give an error message and should not login. |  |  |
| T10 | **FR3** | Check transportation employee login with valid data. | 1.Go to software  2.Enter username  3.Enter password  4.Click login |  | User should login into system as a transportation employee. |  |  |
| T11 | **FR3** | Check transportation employee login with invalid username. | 1.Go to software  2.Enter invalid username  3.Enter password  4.Click login |  | System should give an error message and should not login. |  |  |
| T12 | **FR3** | Check transportation employee login with invalid password. | 1.Go to software  2.Enter username  3.Enter invalid password  4.Click login |  | System should give an error message and should not login. |  |  |
| T13 | **FR4** | A user signs up the system. | 1.Go to software  2.Click sign up  3.Enter name  4.Enter mail  5.Enter phone  6.Enter password  7.Select user type  8.Click sign up |  | System should create an unapproved user and an approval request. |  |  |
| T14 | **FR4** | Administrator approves/disapproves an approval request. | 1.Go to software  2.Login as an administrator  3.Go to notifications page  4.Select the request  5.Click approve/disapprove |  | System should approve the user or disapprove and delete the user. |  |  |
| T15 | **FR6** | Personnel changes his/her password. | 1.Go to software  2.Login  3.Go to my account page  4.Click change password  5.Enter current password  6.Enter new password  7.Click change |  | System should successfully change the password. |  |  |
| T16 | **FR6** | Personnel changes his/her password with an incorrect current password input. | 1.Go to software  2.Login  3.Go to my account page  4.Click change password  5.Enter incorrect current password  6.Enter new password  7.Click change |  | System should give an error message that says wrong password and should not change the password. |  |  |
| T17 | **FR6** | Personnel changes his/her name. | 1.Go to software  2.Login  3.Go to my account page  4.Click change name  5.Enter new name  6.Click change |  | System should successfully change the name. |  |  |
| T18 | **FR7** | Administrator changes other personnel’s password. | 1.Go to software  2.Login as an administrator  3.Go to personnel page  4.Choose personnel  5.Click change password  6.Enter administrator’s password  7.Enter personnel’s new password  8.Click change |  | System should successfully change the personnel’s password. |  |  |
| T19 | **FR7** | Administrator changes other personnel’s password with an incorrect password input. | 1.Go to software  2.Login as an administrator  3.Go to personnel page  4.Choose personnel  5.Click change password  6.Enter incorrect administrator’s password  7.Enter personnel’s new password  8.Click change |  | System should give an error message that says wrong password and should not change the password. |  |  |
| T20 | **FR7** | Administrator changes other personnel’s name. | 1.Go to software  2.Login as an administrator  3.Go to personnel page  4.Choose personnel  5.Click change name  6.Enter personnel’s new name  7.Click change |  | System should successfully change the personnel’s name. |  |  |
| T21 | **FR8** | Administrator adds a new branch employee to the system. | 1.Go to software  2.Login as an administrator  3.Go to add branch employee page  4.Enter name  5.Enter username  6.Choose branch  7.Enter password  8.Click add |  | Administrator should successfully add branch employee to the system. |  |  |
| T22 | **FR8** | Administrator adds a new branch employee with an existing username. | 1.Go to software  2.Login as an administrator  3.Go to add branch employee page  4.Enter name  5.Enter existing username  6.Choose branch  7.Enter password  8.Click add |  | System should give an error message that says the username is already existing and should not add the employee. |  |  |
| T23 | **FR8** | Administrator removes a branch employee from the system. | 1.Go to software  2.Login as an administrator  3.Go to remove branch employee page  4.Choose personnel |  | Administrator should remove branch employee from the system. |  |  |
| T24 | **FR8** | Administrator adds a new transportation employee to the system. | 1.Go to software  2.Login as an administrator  3.Go to add transportation employee page  4.Enter name  5.Enter username  6.Enter password  7.Click add |  | Administrator should successfully add transportation employee to the system. |  |  |
| T25 | **FR8** | Administrator adds a new transportation employee with an existing username. | 1.Go to software  2.Login as an administrator  3.Go to add transportation employee page  4.Enter name  5.Enter existing username  6.Enter password  7.Click add |  | System should give an error message that says the username is already existing and should not add the employee. |  |  |
| T26 | **FR8** | Administrator removes a transportation employee from the system. | 1.Go to software  2.Login as an administrator  3.Go to remove transportation employee page  4.Choose personnel |  | Administrator should remove transportation employee from the system. |  |  |
| T27 | **FR8** | Administrator adds a new warehouse employee to the system. | 1.Go to software  2.Login as an administrator  3.Go to add warehouse employee page  4.Enter name  5.Enter username  6.Enter password  7.Click add |  | Administrator should successfully add warehouse employee to the system. |  |  |
| T28 | **FR8** | Administrator adds a new warehouse employee with an existing username. | 1.Go to software  2.Login as an administrator  3.Go to add warehouse employee page  4.Enter name  5.Enter existing username  6.Enter password  7.Click add |  | System should give an error message that says the username is already existing and should not add the employee. |  |  |
| T29 | **FR8** | Administrator removes a warehouse employee from the system. | 1.Go to software  2.Login as an administrator  3.Go to remove warehouse employee page  4.Choose personnel |  | Administrator should remove warehouse employee from the system. |  |  |
| T30 | **FR8** | Administrator displays personnel. | 1.Go to software  2.Login as an administrator  3.Go to personnel page |  | Administrator should see all personnel. |  |  |
| T31 | **FR9** | Administrator adds a branch to the system. | 1.Go to software  2.Login as an administrator  3.Go to add branch page  4.Enter name  5.Click add |  | Administrator should successfully add branch to the system. |  |  |
| T32 | **FR9** | Administrator adds a branch with an existing name. | 1.Go to software  2.Login as an administrator  3.Go to add branch page  4.Enter existing name  5.Click add |  | System should give an error message that says the name is already existing and should not add the Branch. |  |  |
| T33 | **FR9** | Administrator removes a branch from the system. | 1.Go to software  2.Login as an administrator  3.Go to remove branch page  4.Choose branch |  | Administrator should remove branch from the system. |  |  |
| T34 | **FR9** | Administrator displays a branch. | 1.Go to software  2.Login as an administrator  3.Go to branches page  4.Choose branch |  | Administrator should see branch information. |  |  |
| T35 | **FR10, FR11** | Administrator adds a product to the system. | 1.Go to software  2.Login as an administrator  3.Go to add product page  4.Enter model name  5.Choose category  6.Enter price  7.Click add |  | Administrator should successfully add product to the system. |  |  |
| T36 | **FR12** | Administrator adds a product with an existing model name | 1.Go to software  2.Login as an administrator  3.Go to add product page  4.Enter existing model name  5.Choose category  6.Enter price  7.Click add |  | System should give an error message that says the model name is already existing and should not add the product. |  |  |
| T37 | **FR10** | Administrator removes a product from the system. | 1.Go to software  2.Login as an administrator  3.Go to remove product page  4.Choose product |  | Administrator should remove product from the system. |  |  |
| T38 | **FR10** | Administrator displays a product. | 1.Go to software  2.Login as an administrator  3.Go to products page  4.Choose product |  | Administrator should see product information. |  |  |
| T39 | **FR25** | Administrator displays notifications. | 1.Go to software  2.Login as an administrator  3.Go to notifications page |  | Administrator should see notifications from warehouse employees. |  |  |
| T40 | **FR14** | Warehouse employee displays notifications. | 1.Go to software  2.Login as a warehouse employee  3.Go to notifications page |  | Employee should see stock shortages in the warehouse and requests from branches |  |  |
| T41 | **FR13, FR17** | Branch employee displays notifications. | 1.Go to software  2.Login as a branch employee  3.Go to notifications page |  | Employee should see stock shortages in the branch and status of requests. |  |  |
| T42 | **FR15** | Branch employee requests a product from warehouse. | 1.Go to software  2.Login as a branch employee  3.Go to request product screen  4.Enter model name  5.Enter amount  6.Click request |  | System should successfully create a request and inform warehouse. |  |  |
| T43 | **FR16** | Warehouse employee accepts/rejects a request from branch. | 1.Go to software  2.Login as a warehouse employee  3.Go to notifications screen  4.Choose the request  5.Click accept/reject |  | System should update request status as accepted/rejected. |  |  |
| T44 | **FR18, FR19** | Warehouse employee sends a shipment to a Branch. | 1.Go to software  2.Login as a warehouse employee  3.Go to send a shipment page  4.Choose product  5.Enter amount  6.Choose branch  7.Click send |  | System should create a shipment and automatically assign to a transportation employee. |  |  |
| T45 | **FR18** | Warehouse employee sends a shipment with larger amount than in stock. | 1.Go to software  2.Login as a warehouse employee  3.Go to send a shipment page  4.Choose product  5.Enter larger amount than in stock  6.Choose branch  7.Click send |  | System should give an error message that says there is insufficient stock and should not create the shipment. |  |  |
| T46 | **FR20** | Administrator displays shipments. | 1.Go to software  2.Login as an administrator  3.Go to shipments screen |  | Administrator should see all shipments |  |  |
| T47 | **FR20** | Warehouse employee displays shipments. | 1.Go to software  2.Login as a warehouse employee  3.Go to shipments screen |  | Warehouse employee should see all shipments |  |  |
| T48 | **FR21** | Branch employee displays shipments. | 1.Go to software  2.Login as a branch employee  3.Go to shipments screen |  | Branch employee should see shipments going to the Branch. |  |  |
| T49 | **FR22** | Transportation employee displays his/her shipments. | 1.Go to software  2.Login as a transportation employee  3.Go to my shipments screen |  | Transportation employee should see his/her current and old shipments. |  |  |
| T50 | **FR23** | Transportation employee updates his/her shipment status. | 1.Go to software  2.Login as a transportation employee  3.Go to my shipments screen  4.Choose shipment  5.Click update  6.Choose status  7.Click update |  | System should update shipment status. |  |  |
| T51 | **FR24** | Warehouse employee displays products in the warehouse. | 1.Go to software  2.Login as a warehouse employee  3.Go to products page |  | Warehouse employee should see all products in the warehouse |  |  |
| T52 | **FR24** | Warehouse employee displays a product. | 1.Go to software  2.Login as a warehouse employee  3.Go to products page  4.Choose product |  | Warehouse employee should see product information. |  |  |
| T53 | **FR25** | Warehouse employee informs administrator for out of stocks. | 1.Go to software  2.Login as a warehouse employee  3.Go to products page  4.Click inform for out of stocks  5.Choose product  6.Click apply |  | System should create a request and inform administrator. |  |  |
| T54 | **FR26** | Branch employee displays all products in the Branch. | 1.Go to software  2.Login as a branch employee  3.Go to products page |  | Branch employee should see all products in the branch. |  |  |
| T55 | **FR26** | Branch employee displays a product in the branch. | 1.Go to software  2.Login as a branch employee  3.Go to products page  4.Choose product |  | Branch employee should see product information. |  |  |
| T56 | **FR27** | Branch employee makes a sale. | 1.Go to software  2.Login as branch employee  3.Choose to make a sell  4.Choose product  5.Enter amount  6.Click sell |  | System should decrease the amount of the product with given amount. |  |  |
| T57 | **FR27** | Branch employee makes a larger amount of sale than in stock. | 1.Go to software  2.Login as branch employee  3.Choose to make a sell  4.Choose product  5.Enter larger amount than in stock  6.Click sell |  | System should give an error message that says there is insufficient stock and should not make the sell. |  |  |