CS387 Project Deliverable 4 - Test plan
E-Commerce Recommender System

Bhaskar Gupta	Niraj Mahajan	Shivam Goel	Tathagat Verma
180050022	180050069	180050098	180050111
180050022@iitb.ac.in	nirajm@iitb.ac.in	180050098@iitb.ac.in	180050111@iitb.ac.in

1. Requirement - Login
   Use case: login for all users (i.e buyer / seller / admin / delivery personnel)
   Inputs - username, password
   Outputs - home page, i.e list of products
   Backend changes - none. We have used the user id in tests below as input. They are always valid (hence no negative tests there) due to the check at this step.
   Test procedure - click on the login tab, enter details, click “login”. If details are correct then open the user homepage showing list of products and also the username on the top navigation bar. We will test this with a valid username and password, also with incorrect username and password.

2. Requirement - Sign up for buyer
   Use case: same as requirement
   Inputs - name, username, password, address, email, contact number
   Outputs - if valid details are given then a pop up “account created” otherwise popup “invalid entry”
   Backend changes - create a buyer node with the given details
   Test procedure - click “login”, click “new user”, enter details, click “register”. If the username / number / email are not unique or the email / number are invalid (eg email not having @ etc) then the DB should remain unchanged, otherwise a new buyer node will be created. We will check these DB changes manually for a positive test and for each type of the negative test mentioned.

3. Requirement - View list of products
   Use case: viewing list of product for all users
   Inputs - user id
   Outputs - show list of products based on the recommendations
   Backend changes - none
   Test procedure - list of products are shown in the home page of every user. We will check manually that the products being shown are sorted by the recommendation value generated as per the DB query that we run in a terminal.

4. Requirement - Viewing buyers / sellers / delivery personnel for admin
   Use case:
   a. List all users
   b. Click on a user to view details

   Inputs - admin id
   Outputs - list of buyers / sellers / delivery personnel (the user type that admin selected)
   Backend changes - none
   Test procedure - click on “buyers” / “sellers” / “delivery agents” tab and check that all the corresponding users have been listed.

5. Requirement - Add seller / delivery personnel by admin
   Use case: same as requirement
   Inputs - admin id, name, username, password, contact number, email id
   Outputs - go to the list of seller / delivery agents page (same as the type of user added)
   Backend changes - create a seller / delivery personnel node with the given details
   Test procedure - click on the add seller / delivery personnel tab, enter details, click on “Add”. If the email id / contact number / username are not unique or are invalid (i.e email without @ or number having alphabets) then give a pop up message “invalid entry”. If all details are valid then give the pop up “Added user successfully”. We will use invalid tests of all the types mentioned above which make the entire test and check manually in the DB that valid input creates nodes and invalid inputs make no changes.

6. Requirement - Product details for seller
   Use case:
   a. View details
   b. Increase available quantity of the product

   Inputs - seller id, product id
   Outputs - product details i.e name, description, quantity, cost. If seller clicks on “re-stock” then a pop up message will show saying “quantity updated”
   Quantity input on restocking has to be a non-negative integer. Other forms make negative tests. For invalid input, give a message “invald quantity”
   Backend changes - update the quantity property in the product node
   Test procedure - click on product icon from the list of products, enter new quantity:
   a. 100
   b. -12
   c. 12a

   On test a. we should get the success popup and the invalid popup for cases b. & c. We will check manually that the quantity property of the product node has been updated for test a. and no change occurs in the DB for test b. & c.

7. Requirement - Add product for seller
   Use case: same as requirement
   Inputs - seller id, product name, cost, brand(from dropdown), category(from dropdown), product description, quantity, image link

Cost has to be a positive real number and quantity has to be positive integer. Other forms make negative tests. All other inputs do not have negative tests.
Outputs - will go to the page of listing products of the seller and give a popup of success / failure depending on whether the inputs are valid.
Backend changes - on success : create a new product node. Add edge between the product node and seller node. Add properties of products as given in the input. On failure : no backend changes.
Test procedure - Click on the “add product” tab, get a form, enter details, click on add product. Manually check in DB for creation of a product node and that the product shows in the list of products page for the seller If the cost or quantity are invalid then the DB should remain unchanged. We check with a positive test and negative tests for cost and quantity.

8. Requirement - View requested orders for seller
   Use case:
   a. List the orders which are yet to be approved by the seller
   b. Mark order as “ship”, notifying that a delivery personnel must be allotted

   Inputs - seller id
   Outputs - list of all orders that are yet to be shipped.
   Backend Changes - Update the status of the order as shipped
   Test procedure - Click on the requested orders section from the nav bar. To mark an order as “shipped”, click on the “ship” button. This can be confirmed from the buyer UI where the order will now show shipped. This can also be confirmed from the delivery personnel UI, where the order will show up as an active delivery.

9. Requirement - Product details for buyer
   Use case:
   a. View product details
   b. Add product to cart with specified quantity
   c. List “also bought” and “also viewed” products

   Inputs - product id, buyer id
   Outputs - product name, seller name, cost, brand, category, product description, quantity, image link, also bought products, also viewed products
   Backend changes - On adding an item to cart, “cart” edges must be introduced between the buyer and respective items. “view_history” edge is created between buyer and the product with the current timestamp.
   Test procedure - click on a product to view the details. Click on the “add to cart” button (verify by visiting the cart page for the user). Check manually that a “view_history” edge has been created between the user and product in the DB.

10. Requirement - View cart for buyer
    Use case:
    a. List products in cart
    b. Place order

Inputs - buyer id
Outputs - lists the details of all the products in the cart in brief
Backend changes - On placing an order(clicking “place order”), an availability check is done first. That is, recheck if all the items in the cart are still available. (Someone else might have bought all the available products). If any of the items are not available, pop a message that says “xx items are no longer available”. Once the feasibility of the order is checked, the cart edges from the buyer must be removed and a corresponding order entry must be added to the database.
Test procedure - The cart list can be viewed from the cart section of the buyer UI. On placing the order, the status can be checked in the buyer UI, where the order will be marked “requested”. This order will be visible to the seller for confirmation of shipping.

11. Requirement - View product in the cart for buyer
    Use case:
    a. View product details
    b. Remove item from cart

    Inputs - buyer id
    Outputs - show the product details, i.e name, description, cost, quantity selected. When buyer clicks “remove from cart”, go to the page showing updated list of products
    Backend changes - when user clicks on “remove from cart”, remove the “cart” edge joining the buyer node and product
    Test procedure - click on the product icon from the list of products shown in cart, click on “remove from cart”. Manually check that the cart edge no longer exists in the DB and does not show in the list of products in cart

12. Requirement - List past orders for all users
    Use case: same as requirement
    Inputs - user id
    Outputs -
    a. For admin - list all the orders that were placed at any time
    b. For buyer - list orders placed by the buyer
    c. For seller - list orders that have been accepted by the seller
    d. For delivery personnel - list orders that have been delivered by him/her

    Backend changes - none
    Test procedure - click on “order history” tab and manually check for each case above that the correct filtered data has been shown

13. Requirement - View past order details for all users
    Use case:
    a. Viewing the order details. However different users get to see only the information that is needed for them. Hence different for each user.
    b. Giving rating for the order by buyer

    Inputs - user id
    Outputs :

a. For admin - show all order details, i.e order id, items in the order, order status, delivery personnel, buyer, cost
b. For buyer - order id, status, cost, items. If the status is “confirmed”, then show the delivery personnel name and contact. Otherwise do not show delivery personnel details
c. For delivery personnel - order id, customer address
d. For seller - order id, product, buyer name, cost

Backend changes - none
Test procedure - click on the “order history” tab, click on the order icon for the particular order (for each of the users) and manually check that only the required data is shown for each of the possible input sets

14. Requirement - recharge balance by buyer
Use case: same as requirement
Inputs - buyer id, new balance
valid input : positive real number. Inputs other than this form negative tests.
Outputs - give a pop up saying “balance update successful” and go to the homepage
Backend changes - update the balance property for the buyer node
Test procedure - click on the “recharge” tab, enter the new amount, click on the “recharge” button. We will manually check from the DB that the value has been appropriately updated. For negative tests, the DB should be unchanged. Hence the 2 test values that we will use are 100 and -100.

15. Requirement - List ongoing orders for delivery personnel
Use case: same as requirement
Inputs - delivery personnel id (will be valid as authentication was done in login)
Outputs - list of all orders that are yet to be delivered by the particular delivery personnel
Backend changes - none
Test procedure - Click on “ongoing orders” tab from the navigation bar. Manually check that only the orders have been listed whose status is “confirmed”

16. Requirement - View ongoing order details for delivery personnel
Use case:
a. View order details
b. Mark order as delivered

Inputs - delivery personnel id
Outputs - on clicking “mark as delivered”, go to the list of delivered orders. This completed order will show up along with other completed orders.
Backend changes - Update the status property of order as “completed”
Test procedure - Click on the order icon from the list of orders and then click on “mark as delivered”. Manually check that the order property has been updated and the order is showing in the list of completed orders at the redirected page.