Project Deliverable 2

CS387

AAmazon - Another Amazon

[Recommender System using Graph DBs]

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

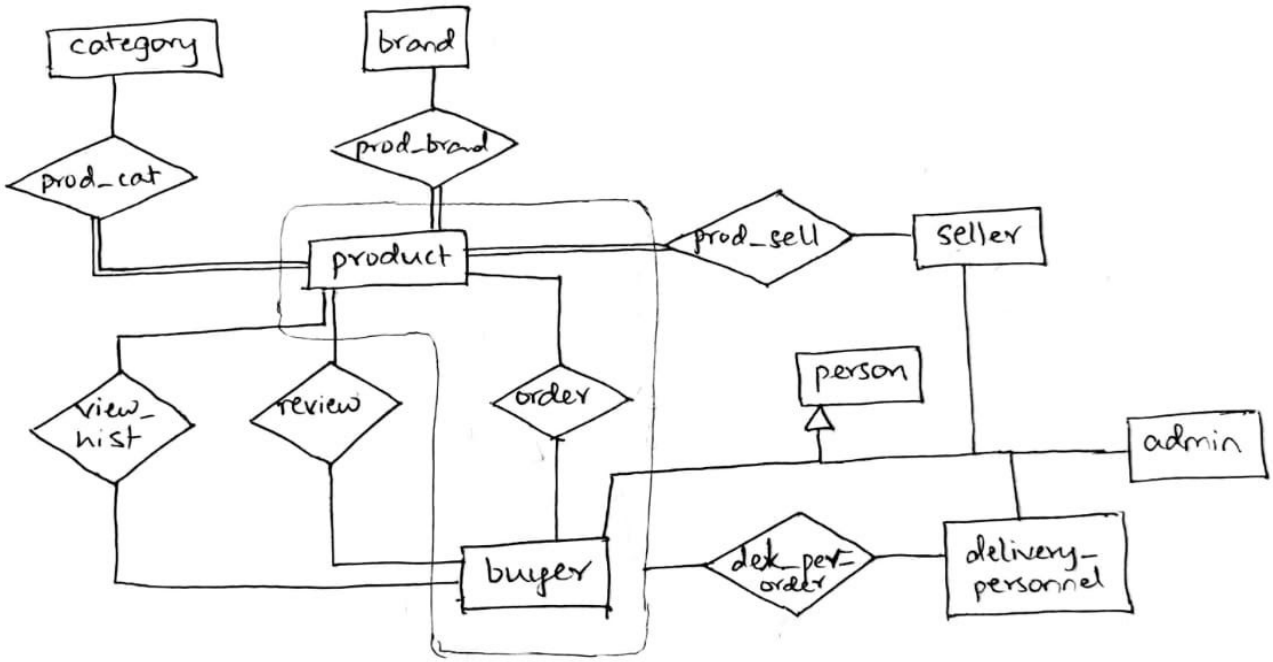
Detailed enumerated English requirements:

1. The application supports 5 roles/modes of access, namely admin, seller, buyer, delivery personnel, and users without login
2. The admin users can view and access all the entities in the dataset (because privacy is a myth). The admin can also add new categories and brands.
3. There will be a special sign-up portal for buyers, sellers and delivery personnel, which will generate a request.
4. Sellers can add objects for sale, where they must include the name, description, price, image and quantity of the product.
5. Any user who hasn't logged in can browse the AAmazon products, but these actions will not trigger any changes to the database, as sellers can misuse this feature to promote their products.
6. Users who are logged in are classified as buyers and may browse and place orders for any item put to sale. The buyer can specify the quantity while placing the order.
7. As soon as the order is placed, it will appear in the buyer's order history with the status "order requested", and this will be sent to the seller to await confirmation of shipping. This will also decrement the available quantity of the said item, preventing buyers from ordering "out of stock" items.
8. An order, once confirmed and shipped by a seller, will be assigned a delivery personnel. This will change the status of the order from "requested" to "shipped". This assignment will be done to the delivery personnel with the minimum number of active tasks. As a result, the order will appear in the UI of the delivery personnel with the pickup and drop addresses. We assume, for simplicity, that any order will be picked up from the "seller warehouse" and delivered to the

"buyer address" by a single delivery agent. Upon delivery, the delivery agent will update the status of the order as "delivered", which will reflect in the buyer and seller order histories.

9. Any buyer will get an option to "review" a product, on a scale [1-5] along with a review_text. This review will be stored, to be used later in the recommendation system. A product's reviews will be visible to other buyers while viewing the product.
10. All transactions will be done using our very own AAmazon pay balance [No we do not support cash, credit or net banking :)]. Any buyer can increase this balance by sending a request to the admin (equivalent to externally transferring money to our AAmazon account). This balance will reflect in the buyer's UI after the admin's approval.
11. We will use a hybrid recommender system with a combination of collaborative filtering and content based filtering. For the content based filtering, we will use the category and brand data of the products from the buyer's order history and browse history. As for the collaborative filtering, we will use the "also view by" and "also bought by" attributes of products, along with their reviews entered by other users.
12. We will use the python API of the neo4j graphical database.
13. The reviews and order history will be imported from a subset of the "Amazon Product Reviews" dataset.
14. A user who is not logged in will see recommendations solely based on the product ratings.
15. A buyer will see recommendations based on his browse and order history on his home page.
16. A search query from a buyer will result in recommendations biased by the search query, along with the buyer's history. The UI for the buyer will include an option to sort and filter the results by price, category, brand and popularity (rating).

ER Diagram



Major entities/relationships:

1. order - order_id, user_id, product_id, status, del_person_id, timestamp - refers to the history of products bought by user
2. review - reviewer_id, product_id, rating, helpful, timestamp, review_text
3. view_history - user_id, product_id, timestamp - refers to a product clicked on by a user
4. product - product_id, seller_id, product_name, availability_count, rating, brand_id, category_id, also_bought, also_viewed
5. delivery_personnel - id, name, username, password, contact
6. category - category_id, category_name
7. brand - brand_id, brand_name
8. person - id, name, username, password
9. buyer -[WEAK]- address, contact
10. seller - [WEAK]- warehouse_address, contact
11. admin -[WEAK]

Use cases:

1. **Search products**:
    a. Description - search the products available for sale. Accessible to all buyers, users without login.
    b. Trigger - None
    c. Primary actor - All buyers and users without login
    d. Inputs by primary actor & constraints on input - Search query for product. No hard constraints, can be any text.
    e. Preconditions for use case: searching can be done from any page(search bar is always available). This is relevant when there are products in our DB.
    f. Main success path: Query on the “product”, “view_history” & “orders” tables to make recommendations and output a list of products in certain order.
    g. Exceptions: None
    h. Post condition: DB - unchanged. UI - show a list of products

2. **View product**:
    a. Description - view a product the user has clicked on, that is seen on searching or on home page
    b. Trigger
        i. Users without login - none
        ii. Buyer - insert into “view_history” table
    c. Primary actor - All buyers and users without login.
    d. Inputs by primary actor & constraints on input - Click on the product.
    e. Preconditions for use case: This is relevant when there are products in our DB.
    f. Main success path: Query on “product” and “review” to obtain all its details and reviews. Show the list of products also bought with the product and a list of reviews of it.
    g. Exceptions: None
    h. Post condition: DB - changed as per trigger. UI - see product details

3. **Give review of product**:
    a. Description - give review on a product being viewed
    b. Trigger - Insert into the “reviews” table
    c. Primary actor - All buyers.

d. Inputs by primary actor & constraints on input - Fields of the review, that is text and rating
e. Preconditions for use case: Viewing product as a logged in buyer.
f. Main success path: Insert into the “reviews” table, go back to page for viewing the product.
g. Exceptions - If a user has already given a review for a product then we will show the option of edit review instead of give review.
h. Post condition: DB - changed as per trigger. UI - see product details

4. **Buy product:**
    a. Description - buy product
    b. Trigger -
        i. Insert into “orders” as requested order(attribute) and await confirmation from the seller.
        ii. Update available quantity in the “product” table.
        iii. Update balance of “buyer”
    c. Primary actor - buyers only
    d. Inputs by primary actor & constraints on input - Click on confirm buy (UI)
    e. Preconditions for use case - On the page of viewing product.
    f. Main success path: Insert into “orders” as requested order and change “buyer” balance.
    g. Exceptions: If quantity given by the user is greater than the available quantity OR order cost is greater than user balance then don’t place order.
    h. Post condition: DB - changed as per trigger. UI - show the view product page with message “ordered successfully” or “invalid order”

5. **Login:**
    a. Description - login / sign up page
    b. Trigger
        i. Sign up - update buyer/seller/delivery_personnel table
        ii. Login - None
    c. Primary actor - All buyers, users without login, seller, delivery personnel,admin
    d. Inputs by primary actor & constraints on input - Email id, password, user type - buyer/seller/delivery_personnel
    e. Preconditions for use case - Admin should exist
    f. Main success path - If sign up, then create a new user & update corresponding table. Then (for both sign up & login) open home page.

g. Exceptions: If invalid login credentials are given then display message & on the same page.
h. Post condition: DB - changed as per trigger. UI - go to home page

6. **Logout:**
    a. Description - log out user
    b. Trigger - NA
    c. Primary actor - Buyer, seller, delivery_personnel
    d. Inputs by primary actor & constraints on input - Click on log out
    e. Preconditions for use case - User is logged in
    f. Main success path - Logout the user and show home page for a not logged in user, i.e popular products
    g. Exceptions: If a non logged in user is accessing then we won't show the log out option.
    h. Post condition: DB - None. UI - go to the home page of a non logged in user.

7. **Update details:**
    a. Description - update details of a user
    b. Trigger - update corresponding user table
    c. Primary actor - buyers, sellers, delivery_personnel
    d. Inputs by primary actor & constraints on input - new email, name, contact, address, etc
    e. Preconditions for use case - User should be logged in
    f. Main success path - Update the buyer/seller/delivery_personnel tables and show the updated details.
    g. Exceptions: If email address id already existing, don't update details and display the message.
    h. Post condition: DB - changed as per trigger. UI - stay on same page showing new details.

8. **View delivered orders:**
    a. Description - view delivered orders by delivery personnel.
    b. Trigger - NA
    c. Primary actor - All delivery personnels
    d. Inputs by primary actor & constraints on input - click on view delivered orders.
    e. Preconditions for use case - can be viewed when logged in as delivery personnel

f. Main success path - Query on the “orders” table to get delivered orders.
g. Exceptions: none
h. Post condition: UI - shows list of delivered orders.

9. **View and update pending deliveries:**
    a. Description - view and update pending orders to delivered orders
    b. Trigger -
        i. View - none
        ii. Update - update in the “orders” table & set status to delivered
    c. Primary actor - All delivery personnels
    d. Inputs by primary actor & constraints on input -
        i. View - click on view pending orders
        ii. Update - click on update order from the list of orders
    e. Preconditions for use case -
        i. View - can be viewed when logged in as delivery personnel
        ii. Update - can be updated on the page with the list of pending orders
    f. Main success path - Query on the “orders” table to view and update pending orders list.
    g. Exceptions: none
    h. Post condition: DB - changed as per trigger. UI - shows list of pending orders.

10. **Recommendation for buyer:**
    a. Description - Home page for a buyer, show list of recommended products using the recommender system
    b. Trigger - NA
    c. Primary actor - All buyers
    d. Inputs by primary actor & constraints on input - click on home.
    e. Preconditions for use case - can be viewed when logged in as buyer or non logged in user, in which case show the most popular products as we don’t have user data
    f. Main success path - Query on “orders”, “review_data”, “meta_data” to generate recommendations and output the list of products.
    g. Exceptions: none
    h. Post condition: UI - shows list of recommended products.

11. **Add a product:**
    a. Description - add product by seller.
    b. Trigger - update “product” table.

c. Primary actor - All sellers
d. Inputs by primary actor & constraints on input - click on add product and provide product details.
e. Preconditions for use case - can be viewed when logged in as seller
f. Main success path - Insert into “product” table to add the product.
g. Exceptions: none
h. Post condition: DB - changed as per trigger. UI - display message that product is successfully added and return to home page.

12. **View products not ordered:**
    a. Description - view products of a seller which have not been ordered by any buyer
    b. Trigger - NA
    c. Primary actor - All sellers
    d. Inputs by primary actor & constraints on input - click on view products not ordered.
    e. Preconditions for use case - can be viewed when logged in as seller
    f. Main success path - Query on “products” table and list all products of the seller where quantity > 0.
    g. Exceptions: none
    h. Post condition: DB - none. UI - show list of products.

13. **View and update order requests:**
    a. Description - view pending order requests made to the seller for some product by various buyers and update from “requested” to “shipped“
    b. Trigger -
        i. View - none
        ii. Update - Update the corresponding “order” status to “shipped”. Query on delivery_personnel and orders to find delivery_personnel with least number of active orders and assign to the order.
    c. Primary actor - All sellers
    d. Inputs by primary actor & constraints on input -
        i. View - click on view pending orders
        ii. Update - click on update order from the list of order requests
    e. Preconditions for use case -
        i. View - can be viewed when logged in as seller
        ii. Update - can be updated on the page with the list of pending order requests. Some delivery_personnel should exist in DB

f. Main success path - Query on the “orders” table to view and also “delivery_personnel” table to update pending orders list.
g. Exceptions: No delivery_personnel in the DB, in that case don’t perform triggers.
h. Post condition: DB - changed as per trigger. UI - shows list of pending order requests.

14. **View products shipped:**
    a. Description - view products shipped by a seller.
    b. Trigger - NA
    c. Primary actor - All sellers
    d. Inputs by primary actor & constraints on input - click on view products shipped.
    e. Preconditions for use case - can be viewed when logged in as seller
    f. Main success path - Query on “orders” table and list all products of the seller where quantity > 0.
    g. Exceptions: none
    h. Post condition: DB - none. UI - show list of products.

15. **Update user balances:**
    a. Description - update balance for payments.
    b. Trigger - update buyer/seller balances.
    c. Primary actor - admin.
    d. Inputs by primary actor & constraints on input - click on user and update the balance.
    e. Preconditions for use case - can be viewed when logged in as admin.
    f. Main success path - Query on “buyer/seller” table and list all buyers and sellers.
    g. Exceptions: none.
    h. Post condition: DB - update as per trigger. UI - show list of users.

16. **Add brands and categories:**
    a. Description - add brand or category.
    b. Trigger - insert into “brand/category” tables
    c. Primary actor - admin.
    d. Inputs by primary actor & constraints on input - fields of brand/category.
    e. Preconditions for use case - can be done when logged in as admin.
    f. Main success path - insert into corresponding table and get back to the page showing brand/category.

g. Exceptions: can only add new brands and categories, no duplicates. Easy code check.
h. Post condition: DB - update as per trigger. UI - show list of brands/categories.