NYU CS-GY 6083 A INET Fall 2024

Project Part 3

Name: Leile Zhang

1 Languages and frameworks

Java, JavaScript, MySQL, html, css, Springboot, React

2 The main queries

2.1 Login User Session Handling:

```
-- Login & User Session Handling:
-- find if userName has is used
SELECT COUNT(*) FROM Person WHERE userName = ?

-- insertPersonSql
INSERT INTO Person (userName, password, fname, lname, email) VALUES
    (?, ?, ?, ?, ?)

-- insertPhoneSql
INSERT INTO PersonPhone (userName, phone) VALUES (?, ?)

-- insertActSql
INSERT INTO Act (userName, roleID) VALUES (?, ?)

-- getRoleByUserName
SELECT r.roleID
    FROM Act a
    JOIN Role r ON a.roleID = r.roleID
    WHERE a.userName = ?

-- findPersonByUsername
SELECT * FROM Person WHERE userName = ?

-- getAllRoles
SELECT * FROM role
```

2.2 Find Single Item:

2.3 Find Single Item:

```
    -- findItemWithLocationByItemId
SELECT i.itemID, i.iDescription, i.photo, i.color, i.isNew, i.
    hasPieces, i.material, i.mainCategory, i.subCategory,
    p.pieceNum, p.pDescription, p.length, p.width, p.height,l.
       roomNum, l.shelfNum, l.shelf, l.shelfDescription, p.pNotes
    FROM Item i
    JOIN Piece p ON i.itemID = p.itemID
    JOIN Location l ON p.roomNum = l.roomNum AND p.shelfNum = l.
       shelfNum
    WHERE i.ItemID = ?
```

2.4 Find Order Items:

```
    -- findItemsByOrderId
SELECT iin.ItemID, p.pieceNum, p.roomNum, p.shelfNum, l.shelf, l.
    shelfDescription
    FROM ItemIn iin
    JOIN Piece p ON iin.ItemID = p.ItemID
    JOIN Location l ON p.roomNum = l.roomNum AND p.shelfNum = l.
       shelfNum
    WHERE iin.orderID = ?
```

2.5 Accept Donation:

```
    -- isValidDonor
SELECT COUNT(*)s
    FROM Person p
    JOIN Act a ON p.userName = a.userName
    JOIN Role r ON a.roleID = r.roleID
    WHERE p.userName = ? AND r.roleID = '4'

-- saveDonation_itemSql
INSERT INTO Item (iDescription, photo, color, isNew, hasPieces,
    material, mainCategory, subCategory) VALUES (?, ?, ?, ?, ?, ?, ?,
    ?)

-- saveDonation_getItemIdSql
SELECT LAST_INSERT_ID()

-- saveDonation_pieceSql
INSERT INTO Piece (ItemID, pieceNum, pDescription, length, width,
    height, roomNum, shelfNum, pNotes) VALUES (?, ?, ?, ?, ?, ?, ?,
    ?, ?)
```

```
16
17 -- saveDonation_donatedBySql
18 INSERT INTO DonatedBy (ItemID, userName, donateDate) VALUES (?, ?,
     ?)
19
20 -- ensureCategoryExists
21 SELECT COUNT(*) FROM Category WHERE mainCategory = ? AND subCategory
      = ?
22
23 -- insertCategorySql
24 INSERT INTO Category (mainCategory, subCategory, catNotes) VALUES
     (?, ?, ?)
```

2.6 Start an order:

```
1     -- isUserInRole
2 SELECT COUNT(*)
3     FROM Act a
4     JOIN Role r ON a.roleID = r.roleID
5     WHERE a.userName = ? AND r.roleID = ?
6
7 -- createOrder
8 INSERT INTO Ordered (orderDate, orderNotes, supervisor, client)
    VALUES (CURDATE(), NULL, ?, ?)
9
10 -- 6 Add to current order (shopping)
11 -- addItemToOrder
12 INSERT INTO ItemIn (ItemID, orderID, found) VALUES (?, ?, true)
13
14 -- getAllOrders
15 SELECT * FROM Ordered
```

3 Difficulties Encountered and Lessons Learned

- **Learning React from Scratch**:
  Starting from scratch to build a React front-end project was particularly challenging because I had no prior experience with React or front-end development. I had to familiarize myself with concepts such as components, state management, and routing. Despite the steep learning curve, I successfully built a functional interface, which gave me a solid foundational understanding of how React projects are structured. This process also deepened my appreciation for modern front-end frameworks and their role in creating dynamic web applications.

- **Improving Proficiency in Spring Boot**:
  The project required the creation of various RESTful APIs to support backend functionality, which significantly enhanced my skills in writing efficient and reusable Spring Boot code. I

learned how to properly structure service layers, handle database interactions with prepared statements, and manage error handling. Additionally, I gained a better understanding of debugging and testing APIs to ensure robust functionality.

- **Challenges in Use Case Analysis**:
  Translating use cases into code was one of the most intellectually demanding aspects of the project. Each use case involved considering multiple backend layers, parameter handling, and potential corner cases. For instance, when implementing the "Accept Donation" feature, I had to carefully design workflows to validate user roles, ensure data consistency across related tables, and handle database constraints effectively. Although this process was complex, it significantly improved my ability to break down abstract requirements into concrete implementation steps.