E-DAC Sep 2020 Project Documentation

Project Name

- **Inventory and Billing System (IBS)**

Course name

- **eDAC**

Team members

1. **Maheshwari Modala (PRN - 200950181050)**
2. **Sagar Sahu (PRN - 200950181087)**
3. **Sandipan Bera (PRN - 200950181089)**
4. **Shubham Sharma (PRN - 200950181101)**
5. **Vaibhav Chauhan (PRN – 200950181109)**

Project Overview

- The application has been designed to be used by such stores or shops where there is need to keep the track of incoming and outgoing goods with minimal interference (or surveillance) of the store manager/shop owner.
- The application opens with the options for login and register. By default, there is no user registered.
- The user to register first will get the administrator rights and can start maintaining the inventory and billing records. The rights provided to an administrator **(apart from the rights provided to a limited user)** are:
    - Keeping track of the goods present in the stock.
    - Updating the price and other attributes of a particular item.
    - Setting the privileges of other employees, including login.
    - Keeping the track of daily sales
    - Fetching reports
        - Purchase report
        - Sales report
- Later on, employees can register themselves using their email ID and password. At the time of registration, the employee will be provided with an auto-generated employee ID which will be used for login.
- **A new employee cannot login until gets approved by the administrator.**
- The application comprises of three pages viz. inventory, billing and monitoring, wherein monitoring page can only be accessed by the administrator(s).
- **Inventory Page (mostly be used at the warehouse/backdoor) –** On this page, entry of the goods received at the warehouse (or shop) will be done with the help of the supplier's invoice number. The records registered here are:
    - Supplier's name, invoice number, registration number & invoice value
    - List of all the goods supplied on that particular invoice having attributes such as item code, name, quantity, unit price etc.
    - All of these records will be saved in the database at once.
- **Billing page (mostly be used at the billing counter) –** On this page, entry of the goods sold will be recorded. The records registered and updated in the database are:
    - Bill no., item name and quantity
    - Customer record – customer name, mobile number, email ID
    - **Note: The privilege of setting the selling price of an item is given only to the administrator, which cannot be altered by any other employee.**
- **Monitoring page (used only by the manager/owner) –** On this page, the manager/owner can analyze the sale/purchase trend and can manipulate the price of the products to increase the efficiency of the store/shop. The other rights enjoyed by an administrator are given above.

Scenario – Billing at the counter

What happens in presentation?

- As soon as the billing page loads, the next bill no. appears in an un-editable field.
- The user will now enter the item code of the item to be billed. There can be three scenarios:
    - The user may enter a wrong item code, which is not there in the stock. He will get an error ("Item not found")
    - The user may again mistakenly enter an item code which is there in the stock but its selling price has not been set by the administrator. He will again get an error ("Item not for sale. Contact administrator")
    - The use may enter the correct item code. The name of the item will be fetched along with its selling price and the quantity available in the store.
- He will then enter the quantity (not more than the quantity available in the stock) and add the item in the table below.
- The table will be populated with all the relevant data based on the number of items sold.
- Total quantity and amount will be calculated in the tabs below.
- The customer's name and email ID will be fetched from the database if the customer has ever purchased from the store earlier.
- The print button will do the following operations:
    - Redirect to invoice page to take the print of the generated invoice.
    - Create an entry of the invoice in the invoices table.
    - Update the quantity of the goods purchased in the inventory table.
    - Create an entry of all the goods purchased on that particular invoice in the item_sale table.

What happens in service?

- A service function is returning a string with the next bill no. based on the count of bills present in the invoices table. Function gets triggered on page load.
- A function takes the item code as the input argument and fetches the data from two tables for that particular item code. Function gets triggered on blur event of item code.
    - One from the inventory table containing all the data related to goods.
    - Another from retail_price_table containing the item code and the selling price.
    - The function then checks if the item code fetched is present in both the tables. If yes, whether it's selling price is set or not. If yes,
        - Only the data relevant for billing (such as item code, brand, item name, unit of measurement, quantity and selling price) is pushed into an object of BillingObject class.
        - This object is then pushed to another object of CResult having status code and reason and is returned from the function.
- A function takes customer's mobile number as input and fetches the customer's details from invoices table. Function gets triggered on blur event of mobile no. field.
- A function takes the invoice object and saves it in the invoices table.
- A function updates the inventory table with the subtracted quantity of items sold.
- A function pushes all the items sold in item_sale table.

What happens in DAO?

- Following are the repositories involved in billing process:

- **InventoryRepository** => having itemCode as the primary key in the table and all other data related to items such as name, quantity, unitPrice etc.
    - Name and quantity of an item is fetched from this table, and
    - Based on the sale, quantity of each item is updated.
- **RetailPriceDataRepository** => having itemCode as the primary key in the table, and only the sellingPrice of the items.
    - Selling price of an item is fetched from this table.
    - Nothing gets updated in this table from billing page.
- **InvoiceRepository** => having invoiceNumber as the primary key and all other information related to customer.
    - Customer's name and email ID is fetched from this table.
    - New invoice nos. and customer details are saved.
- **ItemSaleReporitory** => having sno as the primary key in the table and all list of all the items sold on a particular invoice.
    - No data is fetched from this table.
    - List of invoice nos., item code and quantity sold in saved in this table.

Challenge(s)

- Fetching only one row from the table using a non-primary key.
    - To achieve this, SQL's native query is used wherein, by passing an SQL query as the value inside the @Query annotation returns only what is relavant.
- Getting logged out of the application, every time the page gets reloaded.
    - To overcome this problem, use of localStorage was made, so as to save the state of the user as to whether he/she is logged in or logged out on a particular event.
- A user having limited privilege was also able to visit the monitoring page which is supposed to be visited only by an admin user, because of the <Route> component of react-router-dom.
    - To solve this problem, a customized component is created and <Route> is passed as a prop along with other properties, such as isPrivileged which will route the users to the pages based on their privilege.

Learnings during the project

- **Project planning**
  - Preparing product backlog
  - Preparing sprints
- **Project designing**
  - Service layer design
  - User Interface design using Canva
- **Working in team**
  - Implementation of scrum methodology using Trello's scrum board
  - Implementation of git for version controlling which helped all the team members to get updated with the latest version of the app via Github
  - **Front-end:** https://github.com/finalprojectcdac/project_front_end.git
  - **Back-end:** https://github.com/finalprojectcdac/inventoryandbillingsystem.git
- **Project deployment**
  - The final app has been deployed on Heroku using Github
  - https://ibs-dev-app.herokuapp.com
- **Technologies such as**
  - BootStrap
  - Functional components of ReactJS
  - Git
  - Github
  - Trello
  - Heroku