EECS 4413
Final Project Report
Mom&Pop store

Dmitry Savinov 217303231
Yixi Zhao 214936298
Ahmed Abdelfattah 216225484

Github Repository
https://github.com/ahmedwab/Mom-Pop

Table of Contents

Overall Planning and Collaboration

Phase 1 – Initiation and Planning

- Project kick-off meeting was held over Zoom where the team has discussed implementation requirements, and agreed on the project's scope as well as developed communication plan and development timeline

  Major points were to

  - have weekly Zoom status meetings
  - use What'sApp messenger for day-to-date communication
  - use Github as a storage platform

- First status meeting was used to divide work and assign it to team members accordingly.

  Overall approach was taken:
  Individual work:

  - Development of an assigned requirement *(i.e. Servlet, jspx Page, SQL Relationship, ..ect.)*
  - Unit and component test of the assigned requirement
  - Deployment of an assignment requirement to GitHub branch and regression test

  Group Work:

  - Integration testing activities
  - Peer Code Reviews

Phase 2 – Execution and Monitoring and Controlling (Testing)

- Weekly Team meetings were held in accordance to the communication plan and covered 3 major components:
  - What each team member completed?
  - What each team member will be working on next?
  - Are there any impediments
  - Conduct and demo integration testing
  - Code reviews

Phase 3 – Implementation and Posit Implementation verification

- Team Zoom session to go conduct final round of end-to-end testing to ensure correct functionality after implementation

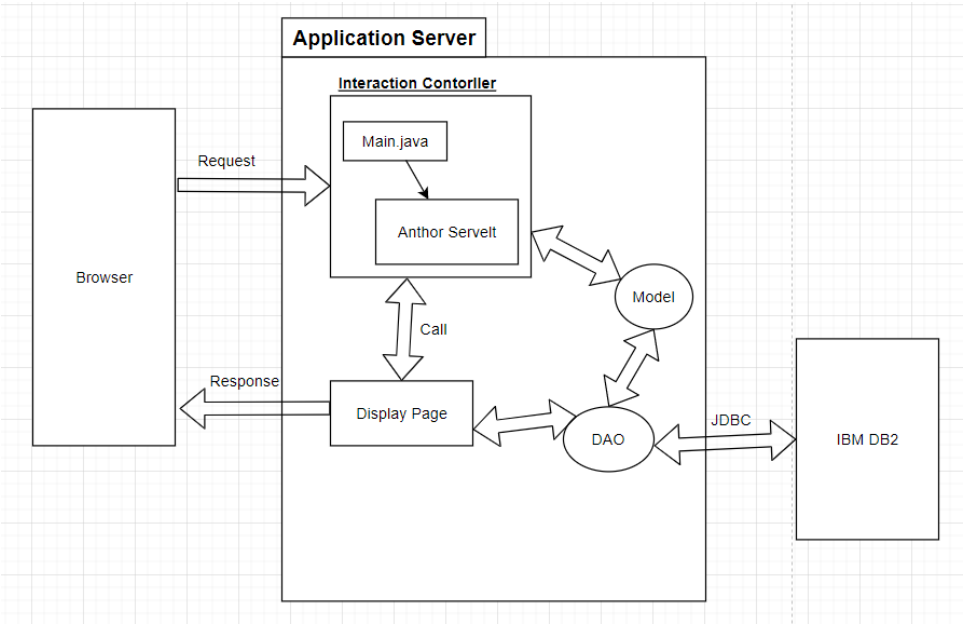
Architecture

To begin with, we decided to put Orders and Items into one table to optimize the database, but we found.there will be trouble in the analytics function. We separated the Orders and Items into two outer join tables. Besides that, to avoid bugs in the database, we also make the book table outer join with the item table and the review table. As for the password, we thought it is unsafe to save the password directly to the database, so we added a Hash function in the model. When the model tries to send a password to the database, it will send the hashed password. On the other hand, the model will hash the password that the users entered and compare it with the password returned from the database which matched the username the user entered. Considering some users don't want to register as Customers or review with their name on the review board, we add functions that allow users process the order and write reviews as anonymities.

Architecture of Application

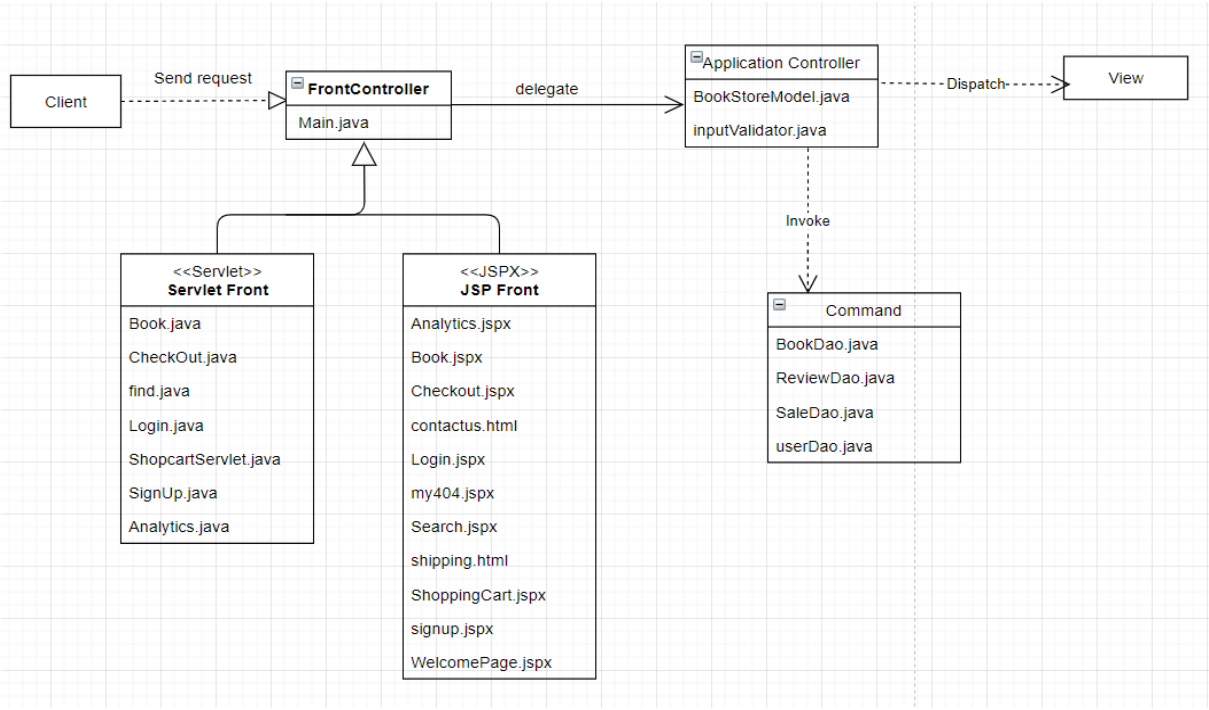
We used MVC with Servlets and JSPX. In our Project, we create a ctrl package including all interaction Controller, a model package including all methods interacting with the Dao part, a Dao package including all the method connecting to database, a bean package including all the encapsulated Objects will be used, and a rest package including all the class for rest service of Order and adding books.

As for design decisions, we decided to design our Projects using Front-Controller Design Pattern based on the MVC architecture, because we thought it is a convenient way to manage and maintain the Project. In this way, we just separate the functionality into three parts, and one of us took charge of one part, and made those parts can be implemented. After finishing, one of us will collect and re-composite three parts as one. We did not need to meet every day to discuss the process of the project. Besides that, using MVC architecture, we can add new functions or new Objects fast, and do not need to do many changes.

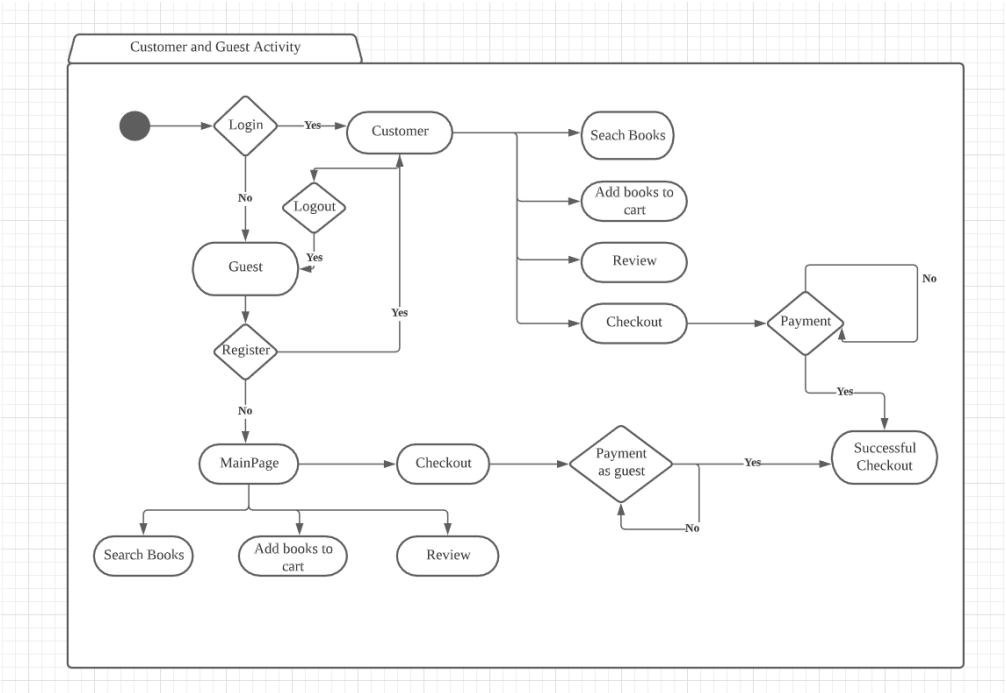


Design Pattern of Application

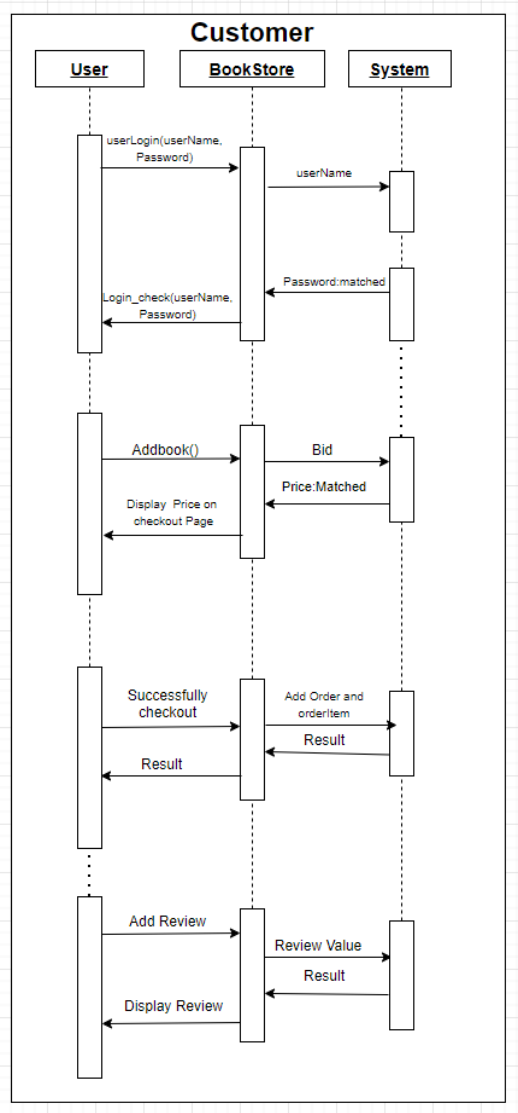
We used the Front-Controller Design Pattern because it is a convenient way to share services and code during several requests, and we do not have trouble controlling and logging users' access. We made a main servlet method as a central entry point for the users.

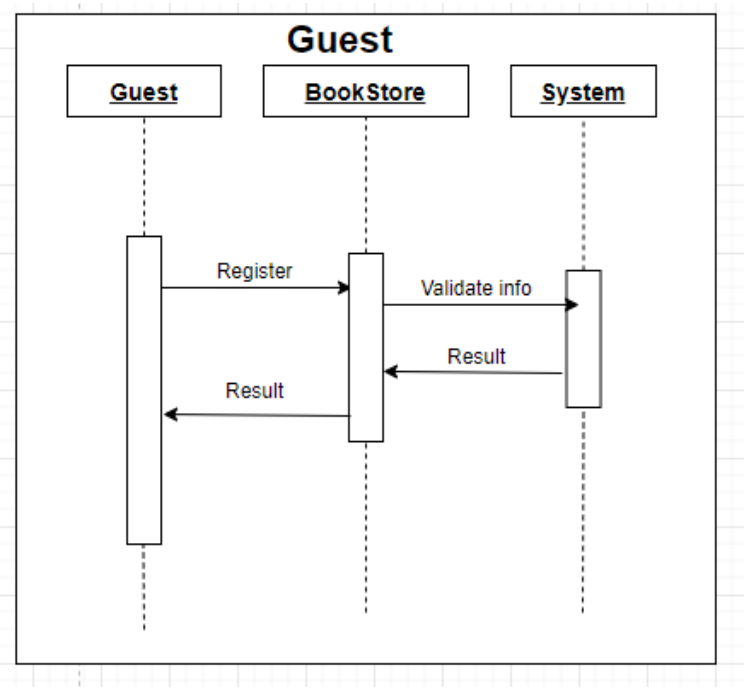


Activity UML diagram

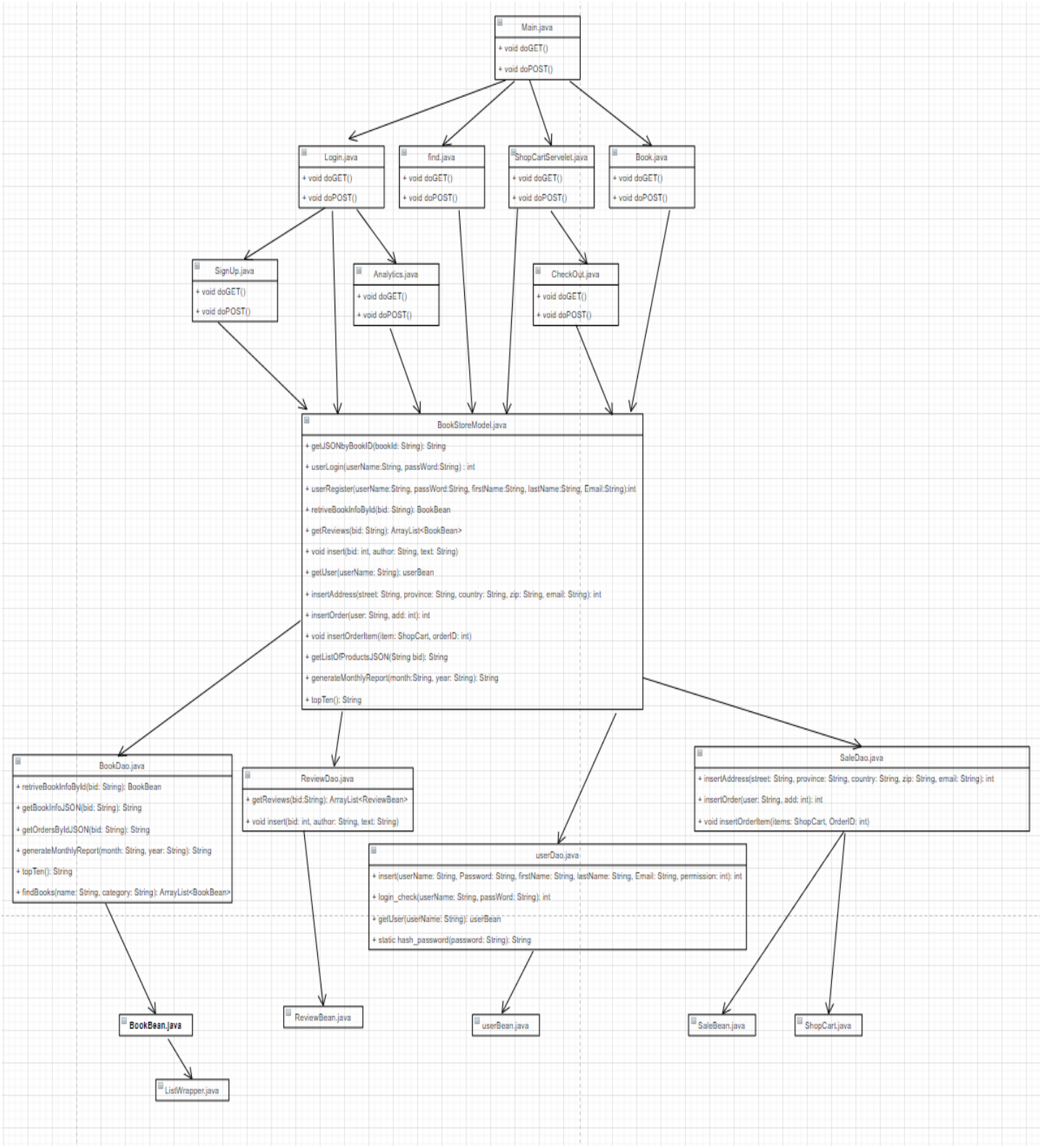


Class Diagram





Class Diagram



Design and Implementation

Authentication

For authentication, we developed a user table in Db2 with a column that indicated whether the user had permission for Analytics and other admin operations indicated by 0 or whether the user was a day to day user looking to shop with permission 0.

Upon logging in in the Login page, we retrieve the user's information that includes username and permission. If username admin logs in with username admin and password ABCabc then the admin will be redirected to the Analytics page where he can display total monthly sales and most popular books.

Otherwise, the logged in user will be redirected to the Main Bookstore page where that person can browse, add reviews and checkout.

Security

Injection

To disallow the possibility of SQL injection, statements were prepared and then executed as queries or updates depending on the wanted functionality of the query.
The class would call the main BookStoreModel and that model class will then call the DAO classes for the intended SQL script.

Moreover, frontend and backend validation was utilized in order to check the values being placed in the input fields in order to nullify invalid inputs. For example, retrieving analytics will go through an input sanitization class to endure that month and year are handled properly to then be communicated with the BookDAO.

XSS security

A filter class under the ctrl package was implemented in order to filter out unwanted scripts that may cause vulnerability in the project. The filtration includes:

- script tags

- src='...' type of expression
- lonesome </script> tag
- lonesome <script ...> tag
- eval(...) expressions
- expression(...) expressions
- javascript:... expressions
- vbscript:... expressions
- onload= expressions

Rest

http://localhost:8080/BookStore/api/products/read?bid=#bookidvalue

Retrieves the book by book id

http://localhost:8080/BookStore/api/orders/read?bid=#orderidvalue

Retrieves the order by order id

Individual Contributions

Team Member	Delivered Components	Learning About Other Elements
Dmitry Savinov	B. Product Catalog Component/Service (REST API) C. Order Process Component/Service (REST API) Shopping Cart functionality	Participating in as well as conducting integration testing activities enabled me to learn about the elements of the project that were developed by others. Peer code review sessions were a great way to get into needy-greedy

		details of the implementation of components developed by others

Ahmed Abdelfattah	E. BookStore Main Page Front End Design G. Payment Page Front End and Back End Integration Testing XSS filtering	SQL integration allowed us to learn integration of database elements and manipulate them with a jspx file.

Frank	Front End and Back End Register and login Set up the IBM DB2 database Write UML diagrams	Learn how to use the IBM Cloud Service and connect it to local projects. How to use the Hash function on Local and Database. How to collaborate with peers on a web project.

Project Team Signatures

TEAM MEMEBER	SIGNAUTRE	DATE
Dmitry Savinov	Dmitry Savinov	Dec 3, 2020
Yixi Zhao	Yixi Zhao	Dec 3, 2020
Ahmed Abdelfattah	Ahmed Abdelfattah	Dec 3, 2020