

# Book Portal Operations Management System

D. A. D. J. Sachithra Dangalla

Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka  
sachithradangalla@gmail.com

**Abstract**— This paper discusses a system that provides an admin system and a public website for a book store with multiple branches. The admin system allows multiple user levels of the staff members to log in and perform separate duties within the system and the public website allows customers to view availability of items and purchase items online. The admin system generates monthly and annual reports, manages user notifications, incorporates role based access control, saves user logs and provides an invoice for physical bill issue in addition to controlling the items showcased in the website and managing the online purchase requests. The system is implemented using JSP and MySQL on MVC architecture and guest checkouts are integrated using PayPal.

**Keywords**—*web application; book; JavaServer Pages; PayPal; online payment; report generation; Admin; Branch Admin; Sales Staff*

## I. INTRODUCTION

Book Portal is a local book store with multiple branches. Currently the items sold are books of many types, which are further divided into subjects. The store manages the physical customer monetary transactions and the report generation is handled manually. The two main problems attended by the Book Portal operations management system are the automatic report generation and reaching out to online customers. The motivation of Book Portal is to step into the world of technology by the use of a central management system to handle the inventory and the internal operations of the store while expanding the customer base by introducing online purchasing ability through a publicly accessible website.

By using a central management system, the integrity of the internal operations is increased and the inventory management allows more flexibility in decision making. Furthermore, the user logs increase the transparency of the operations performed by users within the system. By introducing a public website, the visibility of the store is increased and the customer base is expanded which will result in increasing the profit of the store.

The central system facilitates three user roles, namely Admin (One per system), Branch Admin (One per branch) and Sales Staff. Each user accesses the system by logging in and each user role has a defined set of responsibilities within the system. Report generation is automated by the system and the public website showcases the inventory items which are allowed to be purchased through PayPal transactions.

The paper is structured as follows. Section II discusses the literature relevant to the system. Section III discusses the system models covering the aspects regarding the system requirements and design. Section IV covers the system implementation with regard to the procedures, algorithms used and the user interfaces. Section V discusses the testing approach of the system including the test analysis, performance and failures and security measures. Section VI concludes the paper with conclusions and future work.

## II. LITERATURE REVIEW

Sarasavi online book store [1] is a Sri Lankan store with online book purchasing capabilities. The online store showcases book details and preferred payment methods are credit cards and eZ Cash transactions. Booktopia [2] is a similar Australian based online bookstore with similar features. The main method of payment is credit card transactions. By analyzing the two websites it was observed that the books can be categorized into types, which can further be divided into subjects. Furthermore, Sri Lankan customers expect books to be categorized according to the language, mainly because there is more than one main language used in the country.

Book Portal has integrated PayPal as the main payment method which has credit card paying options embedded in it. Although PayPal is used for this system, currently it is not supported in Sri Lanka. However, the knowledge on the API usage can be used to set up supported commercial payment gateways for actual transactions. Using PayPal has provided an independent layer upon which payments are handled. The book store handles transactions in Sri Lankan rupees and the popular currency format of PayPal is US dollars. Yahoo Finance [6] provides a platform which offers dynamically updated currency conversion.

During checkout, the site prompts for shipping details which the customer can specify manually. This feature allows customers from any corner of the world to buy books and send to any desired person and address within Sri Lanka.

By analyzing the similar systems, it was decided to use JavaServer Pages as the main language of implementation which is proven to be a robust and flexible server-side scripting method incorporated in dynamic web pages. Servlets have a wide set of advantages that includes efficiency, persistency, portability and security [3]. Efficiency is maintained by the use of dynamic service method calls and the session tracking feature in the servlet API is playing a major role in preserving the persistency of the implementation. Java is an independent and portable

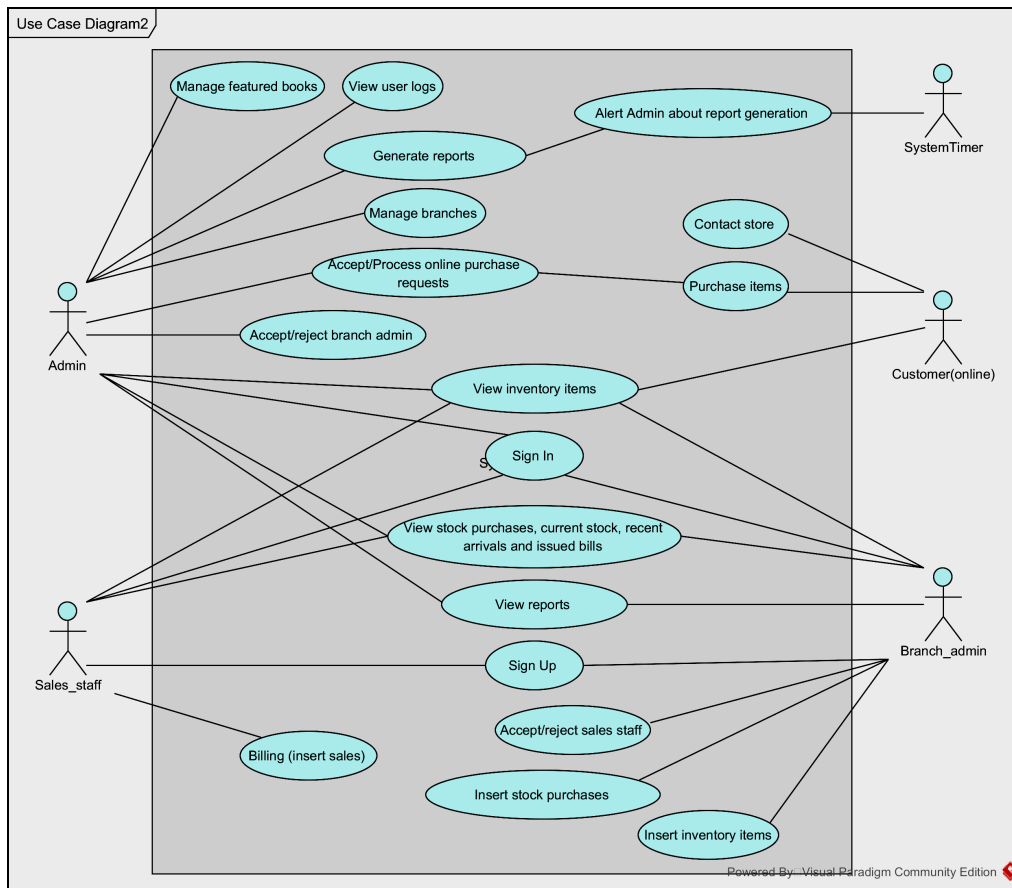


Fig. 1.-Use Case Diagram

platform upon which JSP is built. Java is object oriented, robust and multithreaded, which are favorable features with regard to the Book Portal Operations Management System. Due to the fact that the execution is performed on the server side, the use of servlets secures the system.

### III. SYSTEM MODELS

#### A. System Requirements

The functional requirements of the system include the ability for verified users to log in to the system, manage inventory items by allowing insertion of inventory items. When a user signs up, a notification is to be generated to the specific higher order user which he will have to decide to accept or reject. For example, if a Branch Admin signs up, the Admin will be notified and when a Sales Staff signs up from Branch X, the Branch Admin of Branch X is notified. Only Branch Admins are allowed to insert inventory items (books) and stock purchases. Current stock for each book in each branch is to be maintained by the system which is updated by stock purchases and issuing of bills. Admin is allowed to view user logs, insert branches, view branches, add or remove a set of 10 featured books and manage online purchase requests. Online purchase requests are to be handled by Admin and the system notifies the customer about the status of the purchase request through emails. Physical bills are issued by the Sales Staff and must

incorporate autocomplete mechanisms, auto calculations and user friendly interfaces. The monthly reports on a specific month are to be generated at the start of the successive month and the annual reports of a year are to be generated at the start of the successive year. The monthly branch reports must include the sold list of books and the purchased books and other reports must include an overview and comparison of the sales and purchases.

Figure 1 represents the use cases of the system. One major use case is the sign up, after which the user will have to be accepted by a higher user role before being allowed to log in to the system. All three user levels are allowed to view inventory items, stock purchases, current stock, issued bills and recent arrivals. Branch Admin is allowed to insert inventory items and stock purchases of the system and the Sales staff is allowed to issue bills. The issuing of bills and stock purchases update the current stock which will maintain stock values for each book for each branch. The Admin is responsible for managing featured books and branches and he is allowed to view user logs. Public is allowed to purchase items which are processed by the Admin. The system generates reports relevant to each branch and the entire store monthly and annually each.

## B. System Design

The primary design goal of the system was to create a robust system that interacts with a medium-sized database. Therefore design patterns that support low coupling and high cohesion were considered and MVC design pattern satisfies the main objectives of the system design. The use of object oriented programming language was preferred due to its flexibility and therefore Java and JavaServer Pages were decided as the main languages of implementation while using JDBC to interact with the MySQL database. By the use of an object oriented approach, components are treated as objects and the dependency of the objects are minimized which reduces the complexity of the system. By using the MVC design pattern, the backend implementation of the system is separated from the frontend and the interaction of the two layers can be controlled independently.

Figure 2 represents the models of the system. The hierarchy of the "Report" classes is as shown in the class diagram where Report act as the parent class which is inherited by the four types of report types. The three types of users inherit the "User" class and each user is associated with the classes "Notification" and "UserLog" separately. Each "BranchAdmin" and "SalesStaff" user has a related "Branch" and the "Stock" is associated with the "Branch" which manages the stock of a branch. The main type of object the system is frequently accessing and interacting with is "Book" which has a lengthy list of attributes. Each book has a title, an ISBN code, a "Publisher", a published date, a language and a "Subject" which belongs to a "Type" and a book may have multiple "Authors". When a customer physically purchases books from a branch a "Bill" will be issued which will be processed by the "SalesStaff". If the purchase is online, a "PurchaseRequest" is associated with the Bill which will be processed by the "Admin".

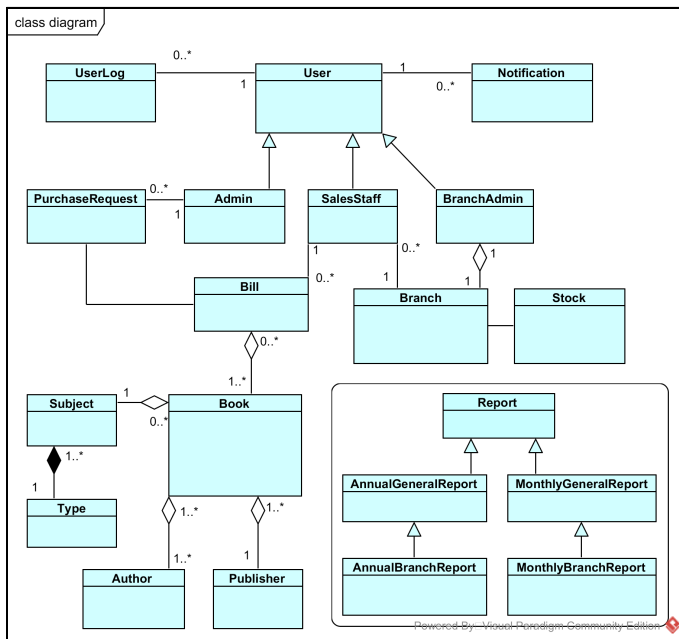


Fig. 2.-Class diagram

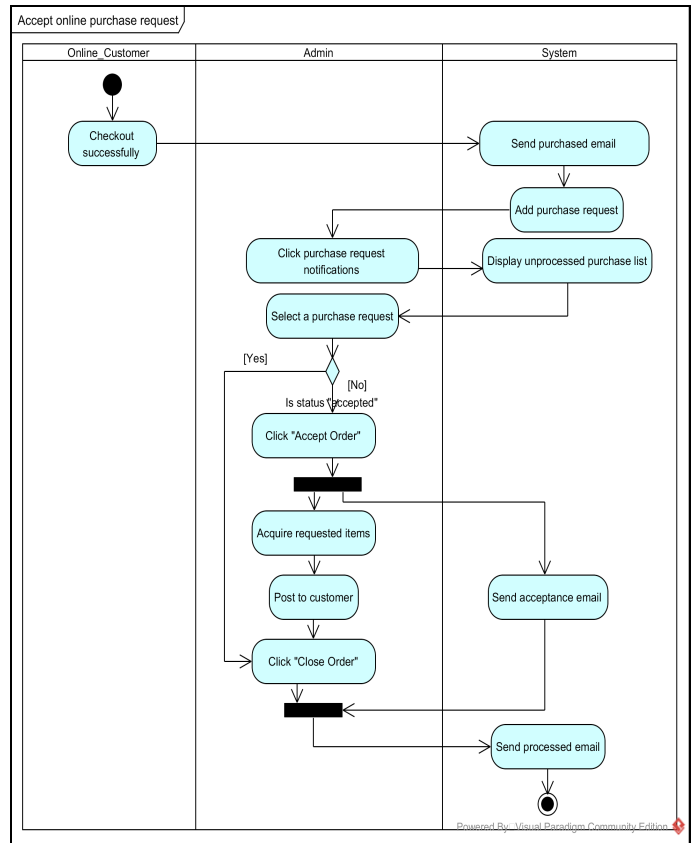


Fig. 3.-Activity Diagram of Online Purchase Process

Figure 3 represents the activities relevant to an online purchase of Book Portal. The system notifies the customer through email and notifies Admin through the internal system notifications. Admin "Accept"s each purchase request after which he manually posts the ordered items. Upon successful processing, Admin "Close"s the request which notifies the customer the delivery status of the order.

The database was designed with reference to the class diagram and the interactions of the tables are quite similar to those between the objects of the class diagram. Each object is identified by a unique ID number and object which have a potential expiry condition are incorporated with a status. User, Branch, Notification tables have an integer status while Purchase Request table has a String status to represent 3 status levels.

## IV. SYSTEM IMPLEMENTATION

### A. Implementation Procedure

The project was initiated as a Java web application in Netbeans IDE 8.1 and the MVC structure was arranged in the project. The ERD was forward engineered to create the database using MySQL Workbench 6.2. The Java classes as designed in the class diagram were created and the initial codes such as getters and setters were implemented. A git repository was created in order to achieve version control of the project.



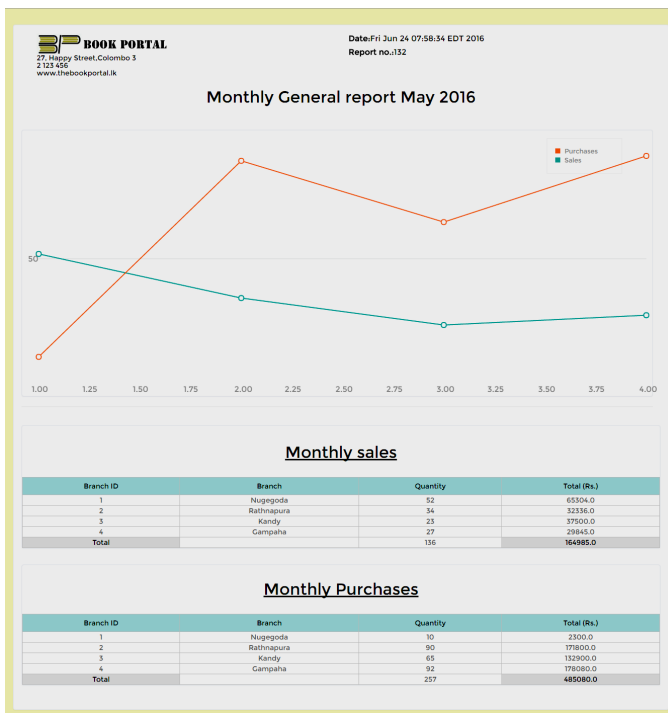


Fig. 5.-Monthly General Report

Figure 6 and Figure 7 include interfaces of the checkout process. The cart includes the functionalities of incrementing and decrementing quantity and removing items from the cart. The order confirmation modal window displays the currency conversion results.

## V. SYSTEM TESTING AND ANALYSIS

### A. Testing Approach

The main types of testing used in Book Portal Operations Management System are database integrity testing, functional testing and user interface testing. Book Portal uses a single MySQL database and during database integrity testing, the database processes were tested as an independent subsystem. That is, the effect of the user interfaces was eliminated by executing the processes separately.

The test targets of the functional testing were basically the use cases of the system. Each use case was tested against relevant and irrelevant inputs and the expected results. Manual testing included examining the source code to find out the bugs and executing the code to observe the results. The results were compared with the expected outcomes in order to verify the correctness of the test target. In the automated approach, the tests of the source code containing logical methods were performed and observed using JUnit [7] and the success or the failure status was observed through the JUnit console.

User interface testing was conducted by incorporating Selenium IDE [8] where the test user actions were recorded and assertions were exported as an executable code. The main objective of user interface testing was to minimize abnormal behaviors and increase the usability of the system.

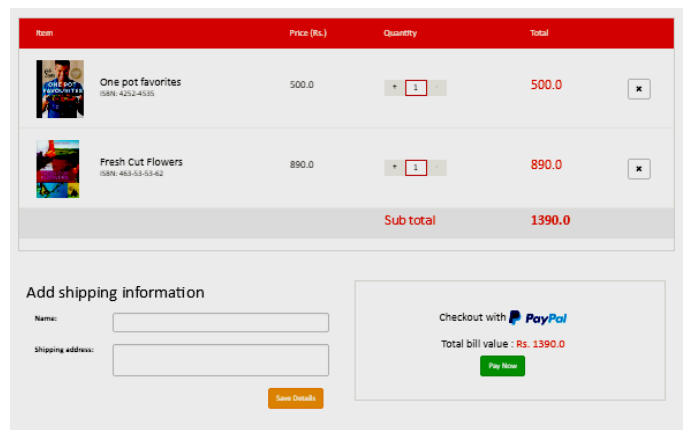


Fig. 6.-Cart items and Checkout

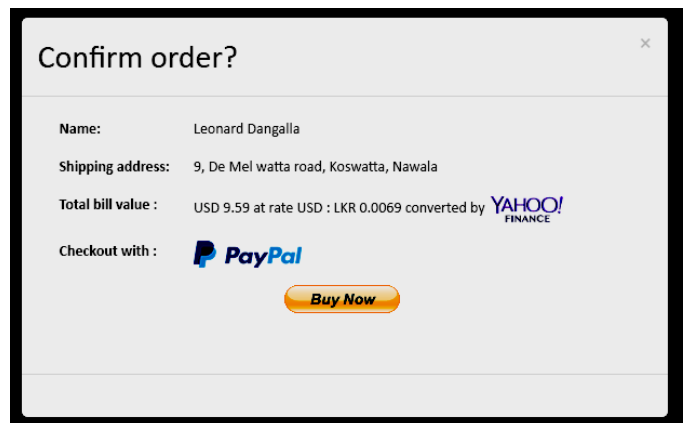


Fig. 7.-Order confirm and Currency Conversion

The end users of the system are assumed to have the basic IT literacy and interface testing tested the user inputs against multiple relevant and irrelevant inputs. Sign up, book insert, issue bill, report view, notification bar, cart, payment status were recognized as crucial user interfaces of the system.

### B. Testing Results

Database integrity testing was successfully conducted which has minimized the possible problems that will affect data integrity. Every field of all the tables in the database was verified either to use a default value or to reject data of inappropriate data type through error messages displayed to the customer without hindering the system processes. During functional testing, each method was tested using JUnit until they returned the exact expected result.

In manual GUI testing, the results were observed after the user manually carried out the testing process and the results were compared against the expected results to analyze each test. In the automated approach, Selenium IDE mimicked the user interaction and stated if the test case was successful or not. Each user functionality is directed to the relevant webpage with a specific parameter (e.g. msg=success) which are typically passed through get method. According to the passed message, success or an error messages are displayed prior to the content of the page.

### C. Performance and Failures

Among the performance issues that were raised during system implementation, one critical issue was the bill issue which consumed undesirable time to load the invoice template. Possible causes were the use of autocomplete mechanism and auto calculation mechanism. The loading time was reduced by implementing the autocomplete and auto calculating mechanisms at the final stage of loading. However, if the user tried to input the first bill item entry of the bill before completion of the page load, about which the user is unaware of, the system entry was frozen until the page load was completed. Therefore the earlier approach was re-implemented where the user is aware of the fact whether the page load is completed or not. Therefore an assumption was made that after issuing a bill, the Sales staff will be processing the order i.e. packing the items or handling monetary transactions while the next bill is being loaded which will provide sufficient time for it to be properly loaded.

### D. Security measures

The security measures of the system include role based access control, password encryption, audit trails and session control. The signed up users must be accepted by higher level users in order to be allowed to log in to the system. In case of a server failure or unauthorized access attempt, error pages 404 and 500 is set up to be directed to. This was implemented by editing the error-pages of the web.xml file of the JSP project. User logs are saved in along with the IP address and a set of keywords describing the action performed. Session control is achieved by the Servlet API usage of the project. MD5, a one-way encryption method is used to encrypt the passwords before saving them in the database. This prevents the violated use of the password in the face of an attacked database.

## VI. CONCLUSION AND FUTURE WORK

This paper presents an online book portal system. The system is successfully implemented and deployed using Openshift cloud [9]. The admin system is accessible through <http://bookportal-sachithra.rhcloud.com> and the public website is accessible through <http://bookportal-sachithra.rhcloud.com/site/home.jsp>. The system has achieved the defined functional and non-functional requirements and the following future contributions are to be deployed through updates.

One of the main possible updates of the system is to save the generated reports in PDF format in the server which will increase the integrity of the reports. The main objective is to capture a JSP webpage as displayed in the web browser, onto a PDF file. This requires the conversion of not only text, but all the components of the report including the graphs and tables to PDF format in JSP and save in the server which can then be retrieved to be viewed or to be downloaded. If there is a mechanism to capture a JSP webpage as displayed in the browser into PDF format, this goal can be easily achieved.

Another major future update is to expand the payment gateways of the system. Currently, the system only uses

PayPal as a payment gateway and the usage of local payment gateways such as eZCash or mCash would increase the visibility of the book store among Sri Lankan customers. Although PayPal is not supported in Sri Lanka as of now, the knowledge on the API usage can be used to set up commercial payment gateways, such as PayHere[10], for actual transactions.

Automated report generation is currently triggered by the admin login date and availability of reports in the database. It can be extended to further enhance the automation process by using a sophisticated utility offered by a 3<sup>rd</sup> party service provider. Currently, the customers purchase through guest checkouts, which has its own set of advantages. However, it would be another achievement if the customers are allowed to create accounts which support "wishlist"s and announcements.

New functionality can be added to receive customer ratings and reviews for books. A functionality to manage announcements such as seasonal offers and discounts through the admin system which can be displayed in the public website is another possible update to the system. These can be allowed to be controlled by the admin system similar to the current implementation of the featured books.

## REFERENCES

- [1] "Books Sri Lanka - Sarasavi online bookstore for buying books online", Sarasavi.lk, 2016. [Online]. Available: <http://www.sarasavi.lk/>. [Accessed: 15- Jun- 2016].
- [2] "Booktopia - Books, Online Books, #1 Australian online bookstore, Buy Discount Books, eBooks and DVDs from Australia and the world.", Booktopia.com.au, 2016. [Online]. Available: <http://www.booktopia.com.au/>. [Accessed: 15- Jun- 2016].
- [3] "Computer Science and Engineering", University of Nevada, Reno, 2016.
- [4] "DataTables | Table plug-in for jQuery", Datatables.net, 2016. [Online]. Available: <https://datatables.net/>. [Accessed: 17- Jun- 2016].
- [5] "JavaMail API Reference Implementation: Wiki: Home — Project Kenai", Java.net, 2016. [Online]. Available: <https://java.net/projects/javamail/pages/Home>. [Accessed: 17- Jun- 2016].
- [6] "Yahoo Finance - Business Finance, Stock Market, Quotes, News", Yahoo Finance, 2016. [Online]. Available: <http://finance.yahoo.com/>. [Accessed: 18- Jun- 2016].
- [7] "JUnit - About", Junit.org, 2016. [Online]. Available: <http://junit.org/junit4/>. [Accessed: 18- Jun- 2016].
- [8] "Selenium - Web Browser Automation", Docs.seleniumhq.org, 2016. [Online]. Available: <http://docs.seleniumhq.org/>. [Accessed: 18- Jun- 2016].
- [9] "OpenShift: PaaS by Red Hat, Built on Docker and Kubernetes", Openshift.com, 2016. [Online]. Available: <https://www.openshift.com/>. [Accessed: 18- Jun- 2016].
- [10] "PayHere - Digital Payments Platform of Sri Lanka", PayHere, 2016. [Online]. Available: <https://www.payhere.lk/>. [Accessed: 13- Oct- 2016].