Final Project
Online Book Store

INFSCI 2710
December 5, 2017

Peng Jin (pej23)
Mingyan Chu (mic130)
Xinyi Chen (xic98)

Table of Contents

Part I: Overview

Our system is a large online book store with multiple users. This book store has a nationwide store system with many branch stores located in different regions. Each region has a region manager and each store has a store manager. The administrators contain a super administrator, region managers, store managers and ordinary salespersons. They all have different authorities in our management system. Our customers include both home customers and business customers. They can login through the same login system, but will be directed to different account pages.

This system has many practical functions. On the one hand, the customers can register, and search, browse, buy books in this system. On the other hand, the administrators can manage and manipulate sales data easily using this system. For example, the super administrator can create, update and delete new stores, region managers can see the regional turnover, store managers can update books in stores, and salespersons can process orders and transactions. When they access the management system, the system will differentiate their identities and then assign different authorities. These functions can be defined as:

(1) login/register system
(2) user account management system
(3) search engine for books
(4) information displaying
(5) automatically assigning salesperson
(6) shopping cart
(7) payment system
(8) dynamic and graphical analysis of data
(9) order management system
(10) customer messaging system

Part II: Assumptions

We have made some assumptions when defining ER diagram as follows:

(1) There can be some new stores which temporarily have no books.
(2) There can be some books which are no longer available in all stores.
(3) Every author recorded should be related to at least one book in Book table.
(4) Every manager should be assigned to only one store or region.
(5) Region managers does not to any store.
(6) Every region defined should has at least one store.

Part III: E-R Diagram

Description:

There are 13 entities in this E-R diagram. The author entity has a “wrote” relationship with book entity. Multiple stores which have id, name and address attributes can have multiple books. But sometimes, when a store is just created, it may not have any book. Many stores should belong to one region. One region has only one region manager which is also an employee. One store has one store manager too. Many salespersons which are also employees can work in one store. One salesperson should process many transactions and many customer messages. There is only one kind of book in one transaction, but the same book can belong to many transactions. One book may have many customer messages while one message only belongs to one book. A customer can send many messages and place many order. The entity customer can be divided into two groups, including home customer and business customer. They share same id, email, address attributes with each other and both have their own specific attributes. One order can only have one card information and one receiver information.

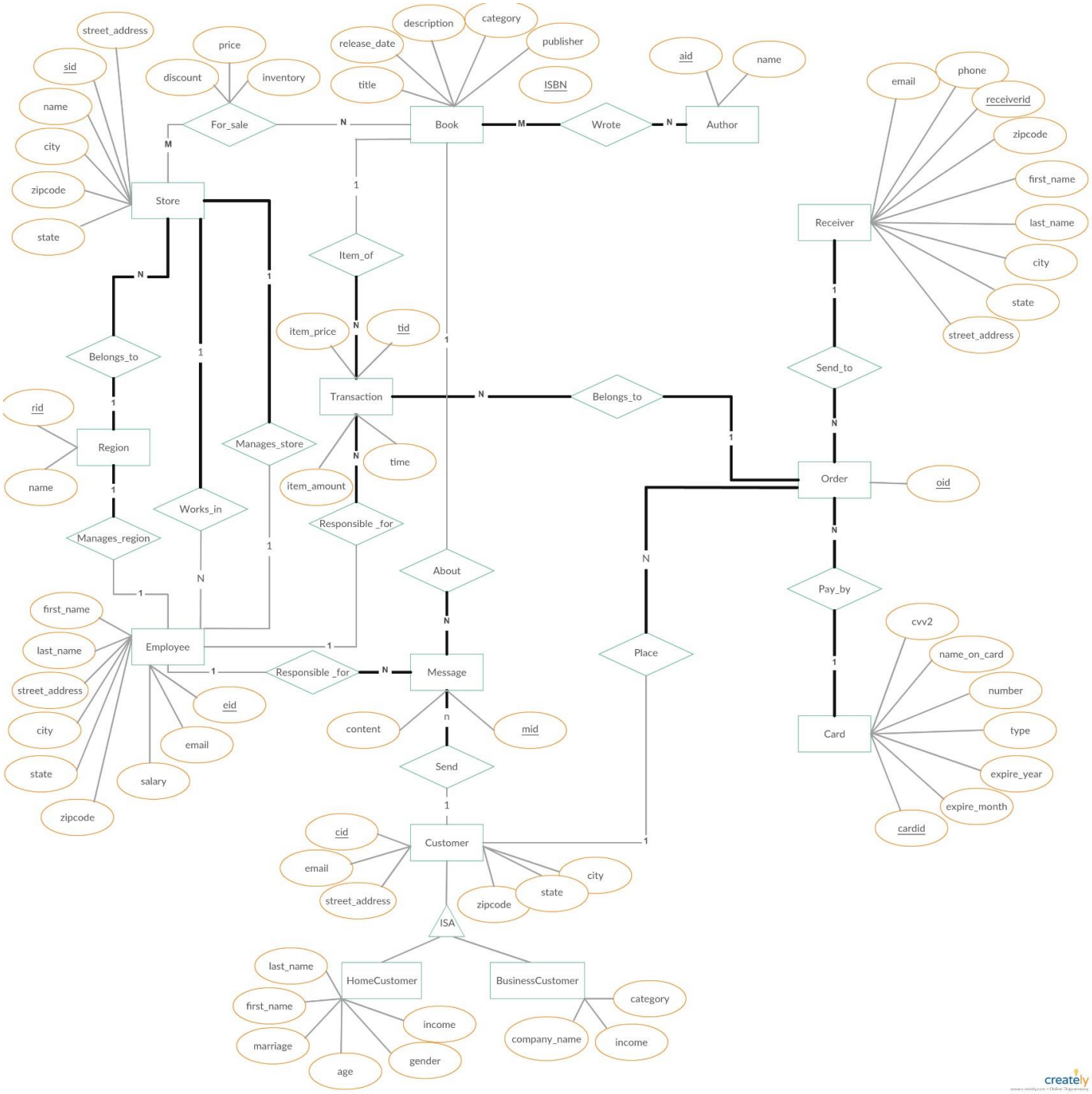


figure1. ER Diagram

Part IV: Relational schema

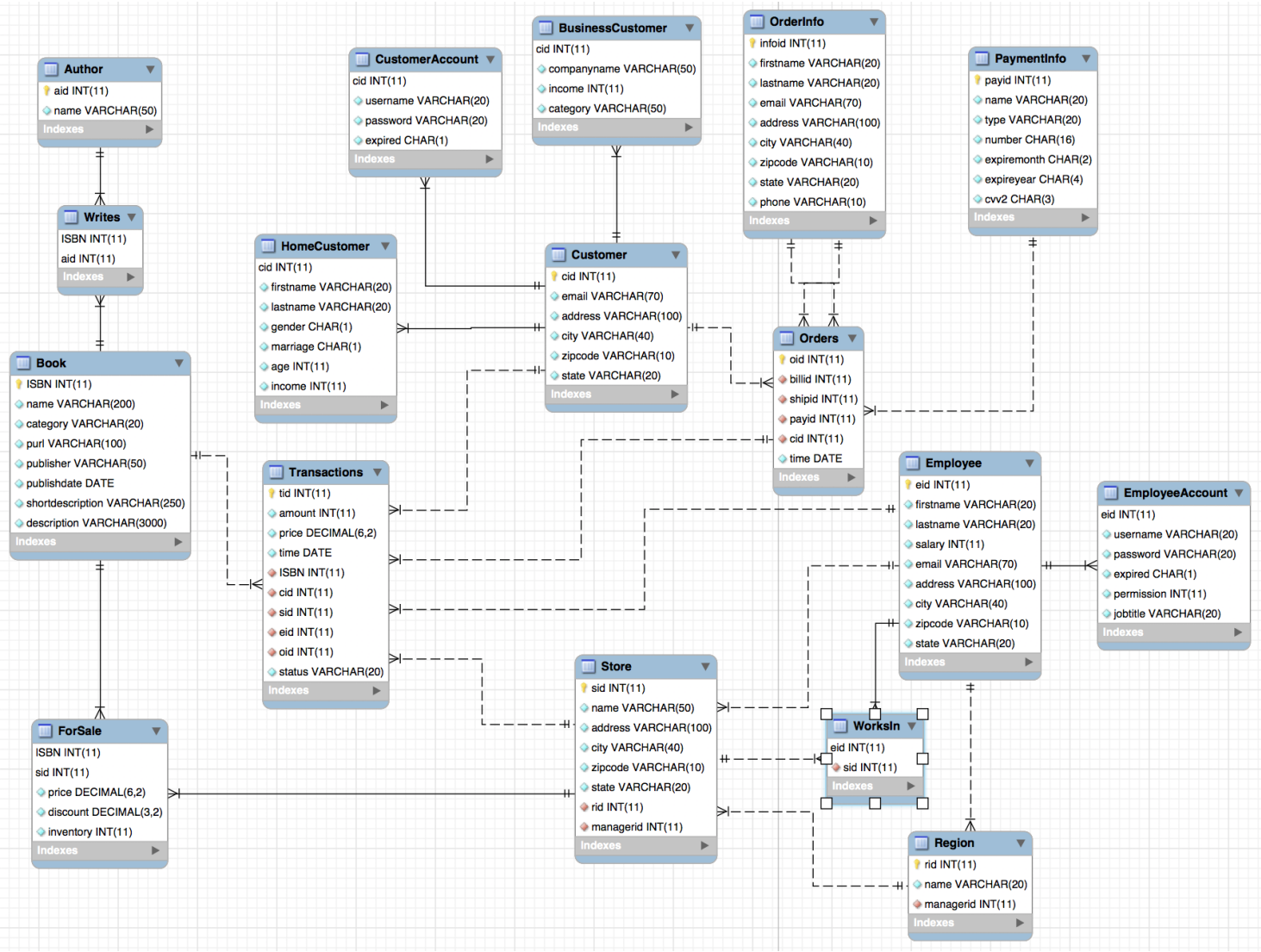


figure2. Relational Schema

Part V: DDL Statements

/*********Customer Section********/

```
CREATE TABLE Customer
(
cid INTEGER NOT NULL AUTO_INCREMENT,
```

```
email VARCHAR(70) NOT NULL,
address VARCHAR(100) NOT NULL,
city VARCHAR(40) NOT NULL,
zipcode VARCHAR(10) NOT NULL,
state VARCHAR(20) NOT NULL,
PRIMARY KEY (cid)
);

CREATE TABLE HomeCustomer
(
cid INTEGER NOT NULL,
firstname VARCHAR(20) NOT NULL,
lastname VARCHAR(20) NOT NULL,
gender CHAR(1) NOT NULL,
marriage CHAR(1) NOT NULL,
age INTEGER NOT NULL,
income INTEGER NOT NULL,
PRIMARY KEY (cid),
FOREIGN KEY (cid)
REFERENCES Customer(cid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE BusinessCustomer
(
cid INTEGER NOT NULL,
companyname VARCHAR(50) NOT NULL,
income INTEGER NOT NULL,
category VARCHAR(50) NOT NULL,
PRIMARY KEY (cid),
```

```
FOREIGN KEY (cid)
REFERENCES Customer(cid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE CustomerAccount
(
cid INTEGER NOT NULL,
username VARCHAR(20) NOT NULL,
password VARCHAR(20) NOT NULL,
expired CHAR(1) NOT NULL,
PRIMARY KEY (cid),
UNIQUE (username),
FOREIGN KEY (cid)
REFERENCES Customer(cid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);
```

/*********Company Section********/

```
CREATE TABLE Employee
(
eid INTEGER NOT NULL AUTO_INCREMENT,
firstname VARCHAR(20) NOT NULL,
lastname VARCHAR(20) NOT NULL,
salary INTEGER NOT NULL,
email VARCHAR(70) NOT NULL,
```

```
address VARCHAR(100) NOT NULL,
city VARCHAR(40) NOT NULL,
zipcode VARCHAR(10) NOT NULL,
state VARCHAR(20) NOT NULL,
PRIMARY KEY (eid)
);

CREATE TABLE Region
(
rid INTEGER NOT NULL AUTO_INCREMENT,
name VARCHAR(20) NOT NULL,
managerid INTEGER NOT NULL,
PRIMARY KEY (rid),
FOREIGN KEY (managerid)
REFERENCES Employee(eid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE Store
(
sid INTEGER NOT NULL AUTO_INCREMENT,
name VARCHAR(50) NOT NULL,
address VARCHAR(100) NOT NULL,
city VARCHAR(40) NOT NULL,
zipcode VARCHAR(10) NOT NULL,
state VARCHAR(20) NOT NULL,
rid INTEGER NOT NULL,
managerid INTEGER NOT NULL,
PRIMARY KEY (sid),
FOREIGN KEY (rid)
```

```
REFERENCES Region(rid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (managerid)
REFERENCES Employee(eid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE WorksIn
(
eid INTEGER NOT NULL,
sid INTEGER NOT NULL,
PRIMARY KEY (eid),
FOREIGN KEY (eid)
REFERENCES Employee(eid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (sid)
REFERENCES Store(sid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE EmployeeAccount
(
eid INTEGER NOT NULL,
username VARCHAR(20) NOT NULL,
password VARCHAR(20) NOT NULL,
expired CHAR(1) NOT NULL,
permission INTEGER NOT NULL,
```

```
jobtitle VARCHAR(20) NOT NULL,
PRIMARY KEY (eid),
UNIQUE (username),
FOREIGN KEY (eid)
REFERENCES Employee(eid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);
```

/*********Product Section********/

```
CREATE TABLE Book
(
ISBN INTEGER NOT NULL,
name VARCHAR(200) NOT NULL,
category VARCHAR(20) NOT NULL,
purl VARCHAR(100) NOT NULL,
publisher VARCHAR(50) NOT NULL,
publishdate DATE NOT NULL,
shortdescription VARCHAR(250) NOT NULL,
description VARCHAR(3000) NOT NULL,
PRIMARY KEY (ISBN)
);

CREATE TABLE ForSale
(
ISBN INTEGER NOT NULL,
sid INTEGER NOT NULL,
price DECIMAL(6,2) NOT NULL,
```

```
discount DECIMAL(3,2) NOT NULL,
inventory INTEGER NOT NULL,
PRIMARY KEY (ISBN, sid),
FOREIGN KEY (ISBN)
REFERENCES Book(ISBN)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (sid)
REFERENCES Store(sid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE Author
(
aid INTEGER NOT NULL AUTO_INCREMENT,
name VARCHAR(50) NOT NULL,
PRIMARY KEY (aid)
);

CREATE TABLE Writes
(
ISBN INTEGER NOT NULL,
aid INTEGER NOT NULL,
PRIMARY KEY (ISBN, aid),
FOREIGN KEY (ISBN)
REFERENCES Book(ISBN)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (aid)
REFERENCES Author(aid)
```

```
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

/*********Transaction Section********/

CREATE TABLE OrderInfo
(
infoid INTEGER NOT NULL AUTO_INCREMENT,
firstname VARCHAR(20) NOT NULL,
lastname VARCHAR(20) NOT NULL,
email VARCHAR(70) NOT NULL,
address VARCHAR(100) NOT NULL,
city VARCHAR(40) NOT NULL,
zipcode VARCHAR(10) NOT NULL,
state VARCHAR(20) NOT NULL,
phone VARCHAR(10) NOT NULL,
PRIMARY KEY (infoid)
);

CREATE TABLE PaymentInfo
(
payid INTEGER NOT NULL AUTO_INCREMENT,
name VARCHAR(20) NOT NULL,
type VARCHAR(20) NOT NULL,
number CHAR(16) NOT NULL,
expiremonth CHAR(2) NOT NULL,
expireyear CHAR(4) NOT NULL,
cvv2 CHAR(3) NOT NULL,
PRIMARY KEY (payid)
```

```
);

CREATE TABLE Orders
(
oid INTEGER NOT NULL AUTO_INCREMENT,
billid INTEGER NOT NULL,
shipid INTEGER NOT NULL,
payid INTEGER NOT NULL,
cid INTEGER NOT NULL,
PRIMARY KEY (oid),
FOREIGN KEY (billid)
REFERENCES OrderInfo(infoid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (shipid)
REFERENCES OrderInfo(infoid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (payid)
REFERENCES PaymentInfo(payid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (cid)
REFERENCES Customer(cid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE Transactions
(
tid INTEGER NOT NULL AUTO_INCREMENT,
```

```
amount INTEGER NOT NULL,
price DECIMAL(6,2) NOT NULL,
time DATE NOT NULL,
ISBN INTEGER NOT NULL,
cid INTEGER NOT NULL,
sid INTEGER NOT NULL,
eid INTEGER NOT NULL,
oid INTEGER NOT NULL,
status VARCHAR(20) NOT NULL,
PRIMARY KEY (tid),
FOREIGN KEY (ISBN)
REFERENCES Book(ISBN)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (cid)
REFERENCES Customer(cid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (sid)
REFERENCES Store(sid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (eid)
REFERENCES Employee(eid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (oid)
REFERENCES Orders(oid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);
```

/*********Other Section********/

```
CREATE TABLE Message
(
mid INTEGER NOT NULL AUTO_INCREMENT,
cid INTEGER NOT NULL,
ISBN INTEGER NOT NULL,
eid INTEGER NOT NULL,
content VARCHAR(1000) NOT NULL,
time DATE NOT NULL,
PRIMARY KEY (mid),
FOREIGN KEY (cid)
REFERENCES Customer(cid)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (ISBN)
REFERENCES Book(ISBN)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY (eid)
REFERENCES Employee(eid)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);
```

Part VI: Front-end design & front-end to back-end connection

Our front-end is composed of many javaserver pages, including the login page, home page, searching result page, product displaying page, shopping cart page, shipping and credit card information page, customer account page, administrator account page, data analysis pages and so on. We use AJAX technology to connect the front-end and back-end. The front-end pages are all in jsp format and the back-end is composed of java servlet documents.

Part VII: System implementation

The flow-process diagram to buy some books is shown below.

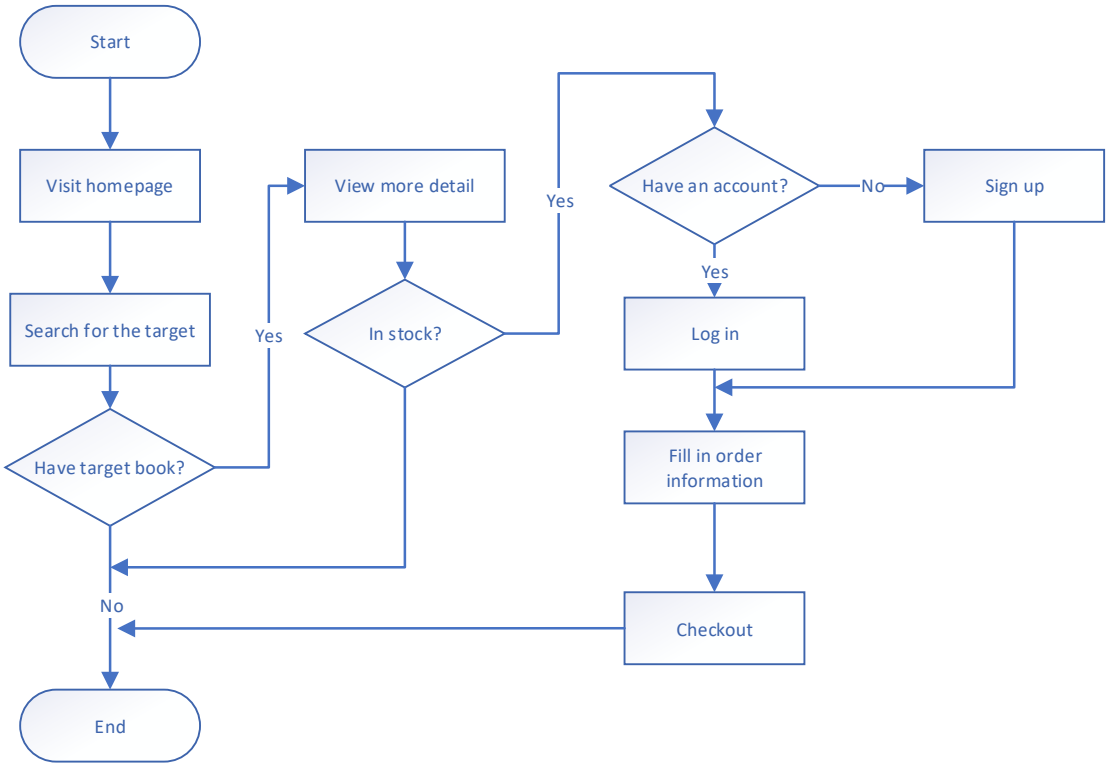


figure3. Float-progress Diagram

To implement the front-end functionality, we have developed some pages, and their descriptions are as follows.

Customer-side Process:

1. URL: /login.jsp

   Parameter: Null

   Functionality:

   Designed to implement the customer-side authentication; use checkbox to judge the current account type, and send the value to server.

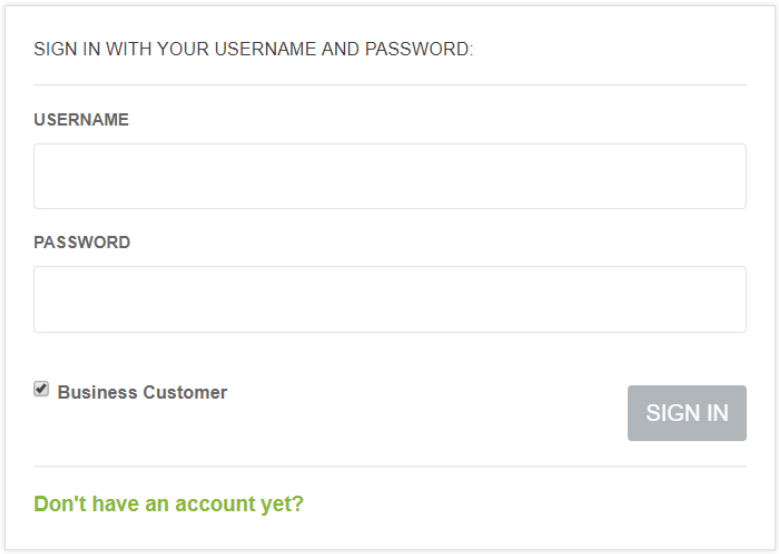


figure4. Login Section

2. URL: /register.jsp

   Parameter: Null

   Functionality:

   Similar to login page, use <select> element to judge the account type, and dynamically generate rest of the form. Implemented the validation check of user inputs, including username, password confirmation, zip code, etc.

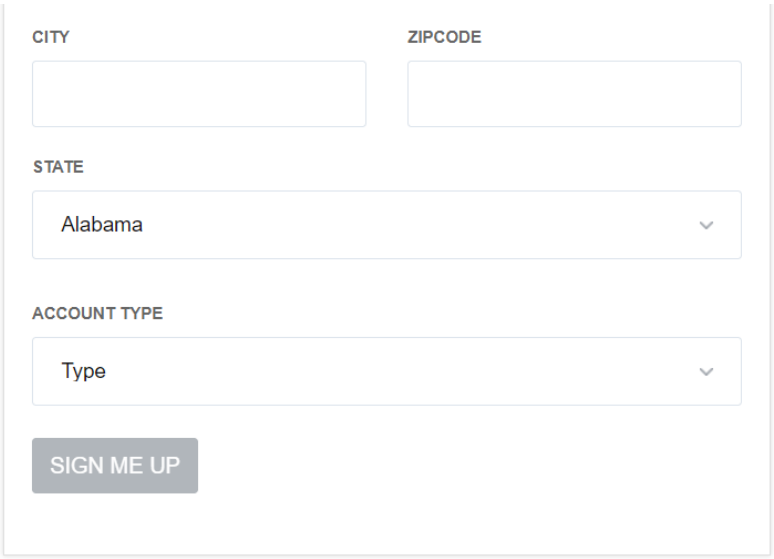


figure5. Part of Register Section

3. URL: /index.jsp

   Parameter: Null

   Functionality:

   Homepage. Show featured products and top 12 books.

figure6. Part of Homepage

4. URL: /search.jsp

   Parameter: [category] | [name] | [ISBN], at least 1 parameter

   Functionality:

   Get the parameter from other pages, and send them to server through ajax. After getting the response, show search results. Also implemented the category and price filter interface, and rerank function.

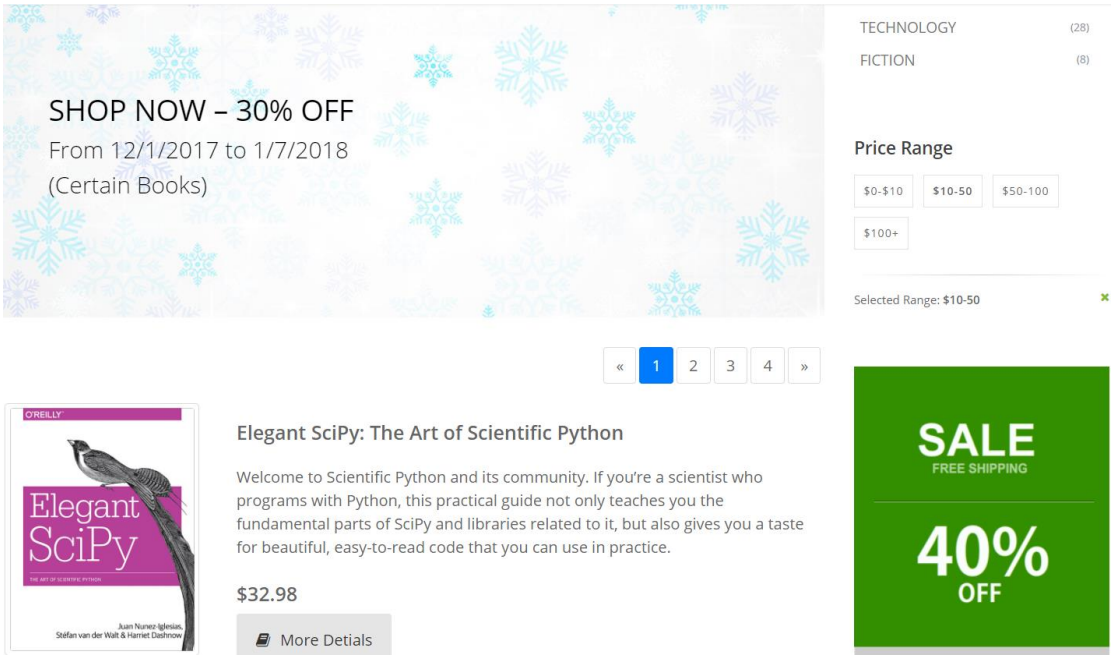


figure7. Part of Search Result Page

5. URL: /product.jsp

   Parameter: ISBN && sid

   Functionality:

   Get certain product's information such as book title, author and description. Customers can also send messages about this item to the salesperson.

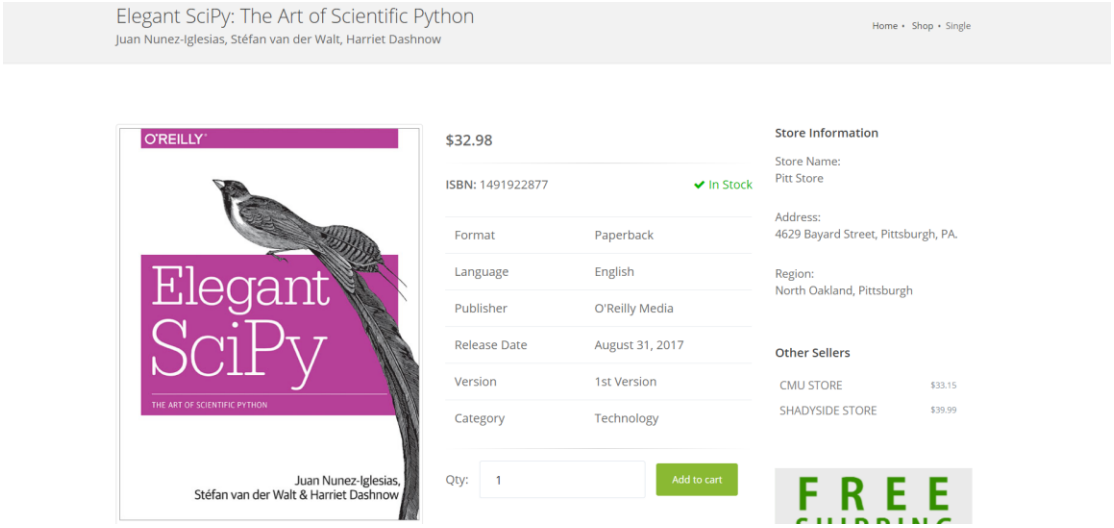


figure8. Part of Product Detail Page

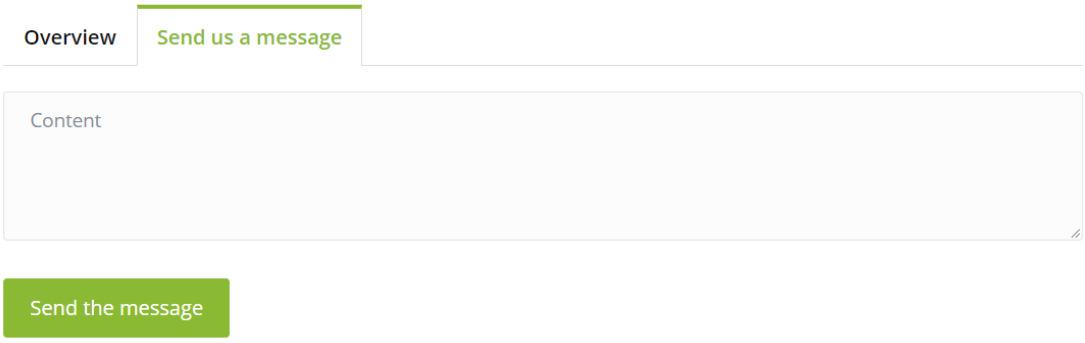


figure9. Send Message Section

6. URL: /cart.jsp

   Parameter: Null

   Functionality:

   Show items in cart. Customers are able to update the cart.

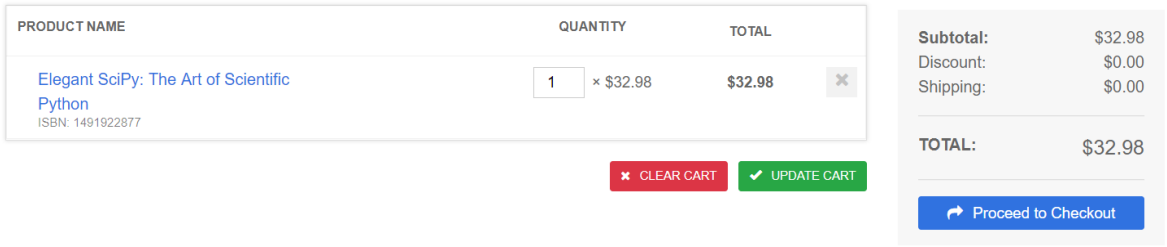


figure10. Shopping Cart

7. URL: /cart.jsp

   Parameter: Null

   Functionality:

   Implemented the interface for customers to fill in the shipping, billing and payment information.

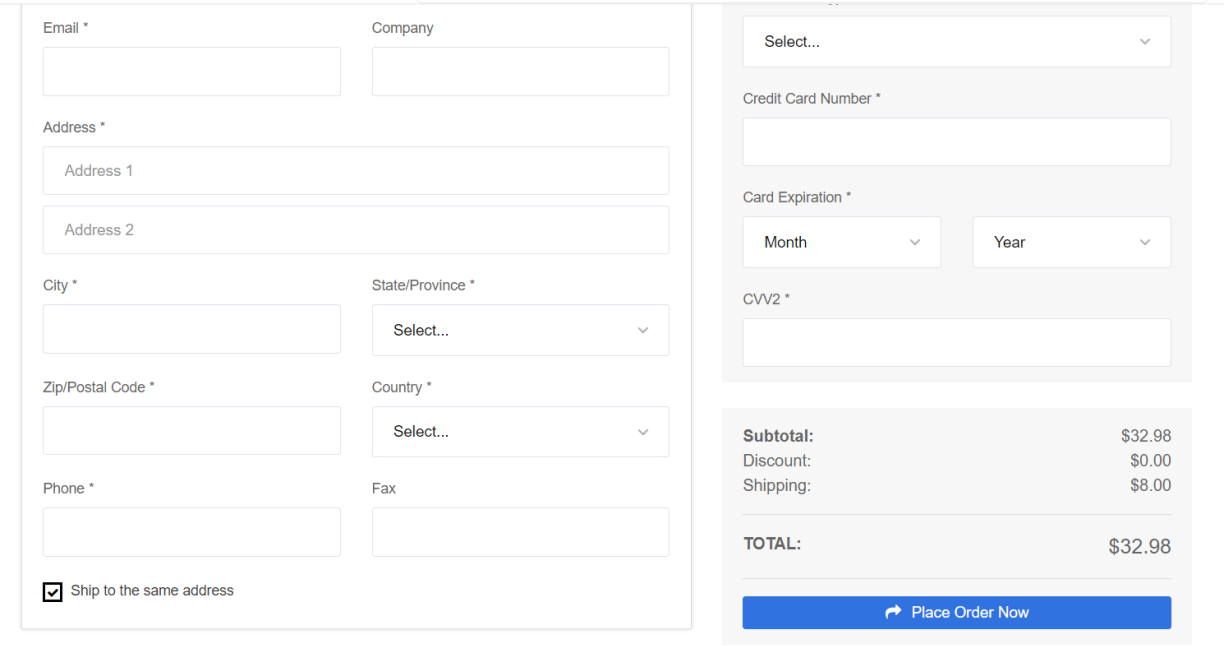


figure11. Payment Page

8. URL: /cart-result.jsp

   Parameter: Null

   Functionality:

   Show the result of payment.

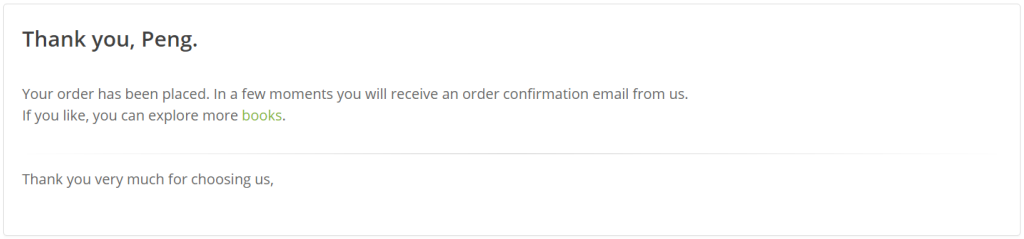


figure12. Payment Success

9. URL: /cart-result.jsp

   Parameter: Null

   Functionality:

   Allow customers to change some of the settings and view order history.

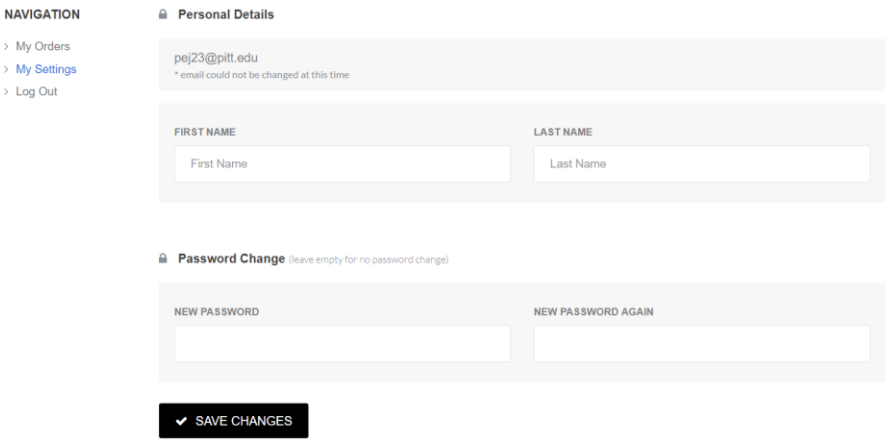


figure13. Customer Settings

Business Analysis Process:

1. Log in with a manager account.

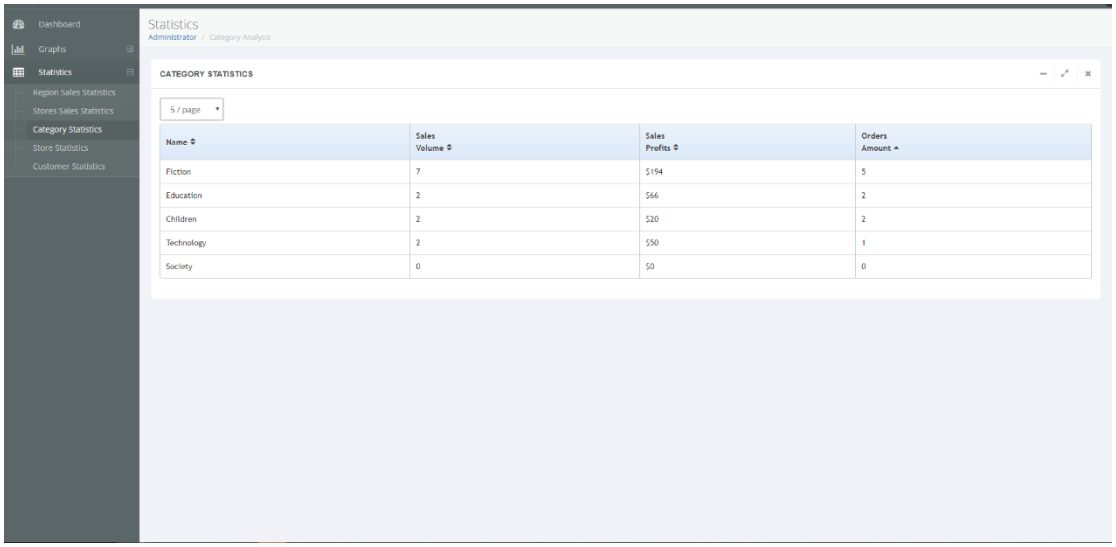


figure14. Manager Login

2. Use menu to direct

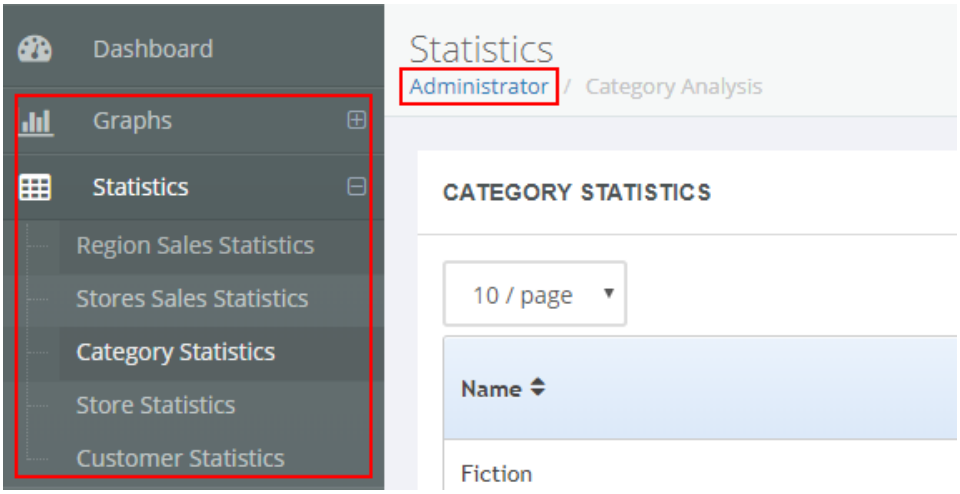
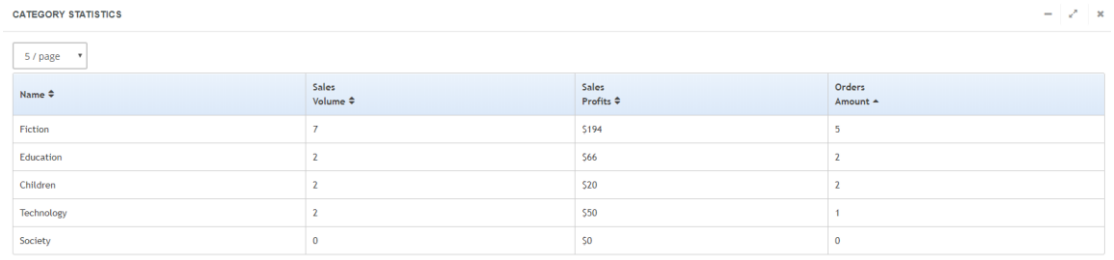


figure15. Dashboard

Managers in different levels will see different menu and data.

3. Category statistics

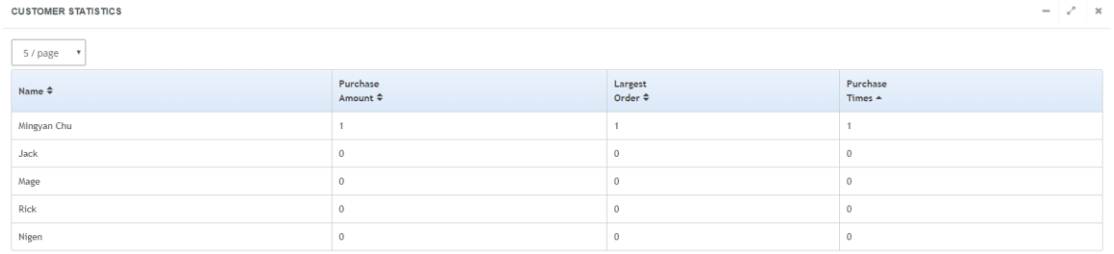

CATEGORY STATISTICS

5 / page

Name	Sales Volume	Sales Profits	Orders Amount
Fiction	7	$194	5
Education	2	$66	2
Children	2	$20	2
Technology	2	$50	1
Society	0	$0	0

figure16. Category Statistics

4. Customer statistics

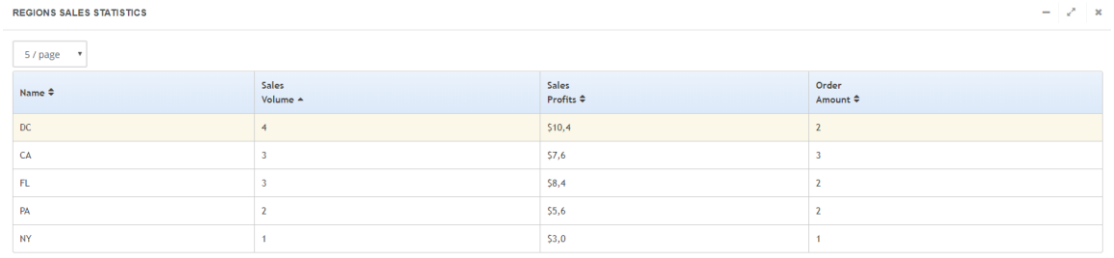

CUSTOMER STATISTICS

5 / page

Name	Purchase Amount	Largest Order	Purchase Times
Mingyan Chu	1	1	1
Jack	0	0	0
Mage	0	0	0
Rick	0	0	0
Nigen	0	0	0

figure17. Customer Statistics

5. Regions sales statistics

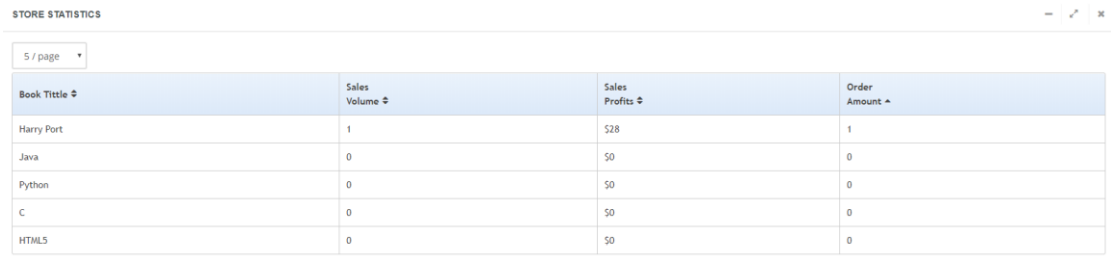

REGIONS SALES STATISTICS

5 / page

Name	Sales Volume	Sales Profits	Order Amount
DC	4	$10,4	2
CA	3	$7,6	3
FL	3	$8,4	2
PA	2	$5,6	2
NY	1	$3,0	1

figure18. Regions Sales Statistics

6. Stores statistics


STORE STATISTICS

5 / page

Book Tittle	Sales Volume	Sales Profits	Order Amount
Harry Port	1	$28	1
Java	0	$0	0
Python	0	$0	0
C	0	$0	0
HTML5	0	$0	0

figure19. Stores Statistics

7. Stores sales statistics


STORES SALES STATISTICS

5 / page

Name	Sales Volume	Sales Profits	Order Amount
Pitt	1	$28	1
UPenn	1	$28	1
Penn State	0	$0	0
Phl	0	$0	0
Rose Park	0	$0	0

figure20. Stores Sales Statistics

8. Some graphs demonstrate more information

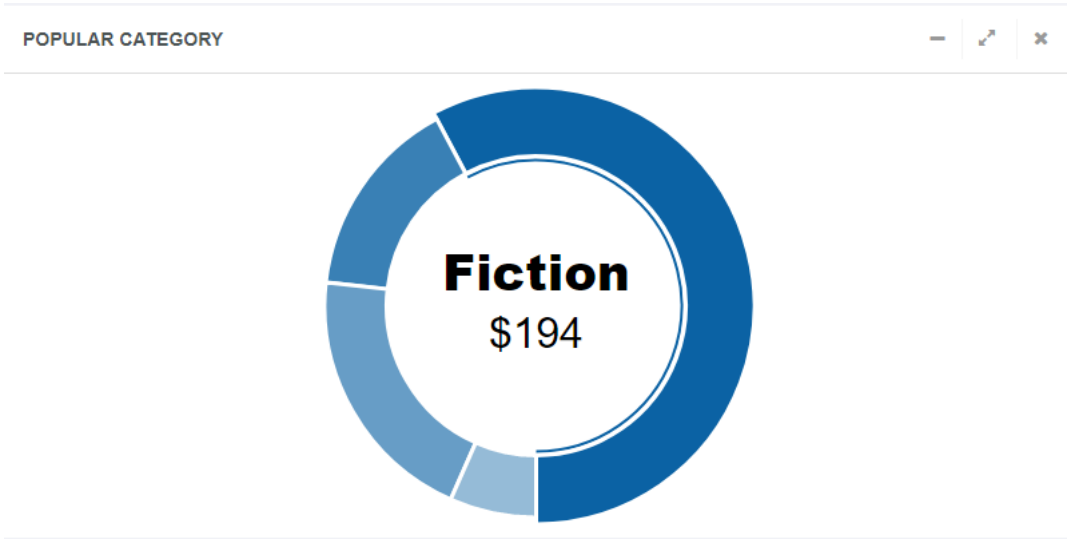


figure21. Demo1

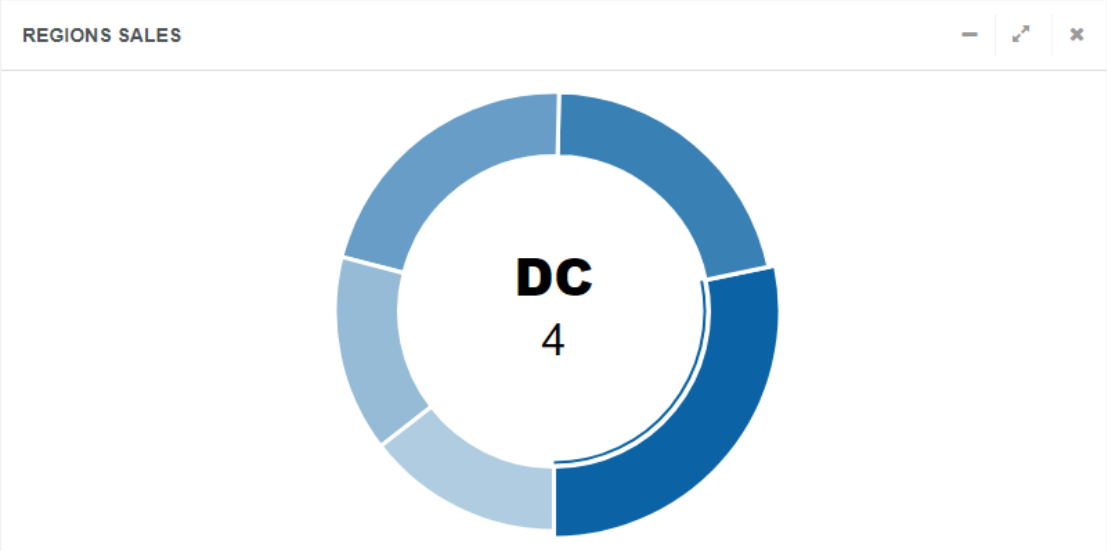


figure22. Demo2

Part VIII: Testing efforts and erroneous cases

1. low inventory control

Low inventory control means when a book is out of stock, the customer cannot add it into the shopping cart. Or the customer can't add books more than the stock.

2. register control

When a user is registering an account, he can't enter a zip code which is more than 5 digits, a phone number more than 10 digits, or a username which has been existing in the database.

3. Inserting error control

In some cases, a customer writes a message to a specific salesperson after this salesperson is fired for some reason, it may cause inserting error. In case of this situation, we have added several preventive measures in both front-end and back-end. The front-end will control the availability of the submit button, and the back-end will also return an error notification before inserting.

Part IX: Limitations & possibilities for improvements

Since the shopping cart function is based on session storage, once the customer adds a book into shopping cart, it is firstly stored in session and then written into database when the customer places the order. Thus, if the customer closes the browser, the session will be emptied which means the shopping cart will be emptied as well. It still needs future improvements.

This system hasn't been added anti-injection function. If someone maliciously utilizes our search engine to attack our system, it may face SQL injection risk.

Because of time limitation, we didn't add writing book review and comment function. With this function, the customer who has bought one specific book can write a review and rate this book. It can be added in future construction.