

CS 591 / Spring 2002

Web Technologies and E-commerce

Design Document of Assignment #2

Online Bookstore

**Qingfeng Duan
Qunzhang Li
Jiaye Zhou**

March 27, 2002

CONTENTS

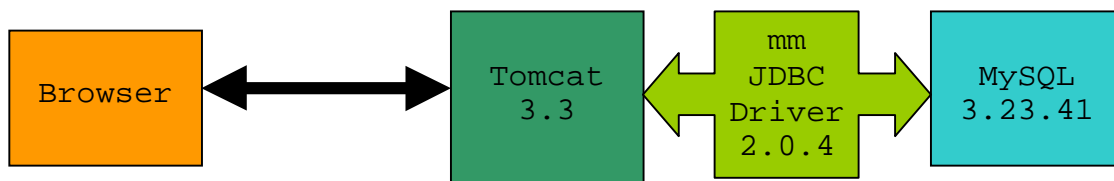
1. Introduction -----	1
2. Architecture -----	1
3. Features -----	1
4. How to Run -----	2
5. Database Design -----	3
6. Directory Structure -----	6
7. Java Servlet / JavaScript Code -----	7

1. Introduction

In this assignment, we developed an online bookstore. We provide a web interface to the user. The user can browse the catalog, check the detailed information of book, put books to the shopping cart, and check out with payment by credit card. The user can also search a book and open and update account information. The user will receive an order confirmation by email.

2. Architecture

This project is developed as a web application. The web server is Tomcat version 3.3. Tomcat supports Java Servlet and Java Server Pages (JSP). We use Java Servlets in our development. We use MySQL database to store the book information, user account and order information. Mark Matthew's JDBC driver `mm.mysql-2.0.4-bin.jar` is used to connect Tomcat and MySQL.



3. Features

3.1 Dynamic HTML

We use Cascade Style Sheet (CSS) and JavaScript to realize the dynamic HTML feature. JavaScript is used to validate form items such as email address and credit card information. All items are checked by JavaScript before submitted to the web server. If some items are not correctly filled, the application will ask the user to enter them again. One interesting thing is the validation of credit card number. Our program can check whether a credit card number is valid or not.

3.2 Search engine

We provide a simple search engine to the user. The user can search a book from our catalog. The supported search keys are book title, ISBN, author's last name and author's first name. The search engine also supports the pattern-matching search. A user can input partial information about the book and do not need to do an exact match searching.

3.3 Personalization

The user could open and update her/his account to store basic information. The information include email address, first name, last name, shipping address, phone, billing address, credit card brand, credit card number and expiration date of the credit card. A returned customer does not need to enter this information when does checking out. The program will get the data from the database and fill the form automatically. If a returned customer wants to update some items, she or he can input those items and submit to the database.

Another personalization is that when a returned customer logs in from the same computer, which she or he used to visit our web store before, the program will provide some recommendations based on her or his former shopping activities. We do this by storing cookies in the customer computer.

3.4 Persistent storage

We use MySQL database to store the book catalog, user account and order information. Please refer to the section 5 "Database Design" for detail.

4. How to Run

The URL of our online bookstore is <http://customer.cs.unm.edu:8080/1group>.

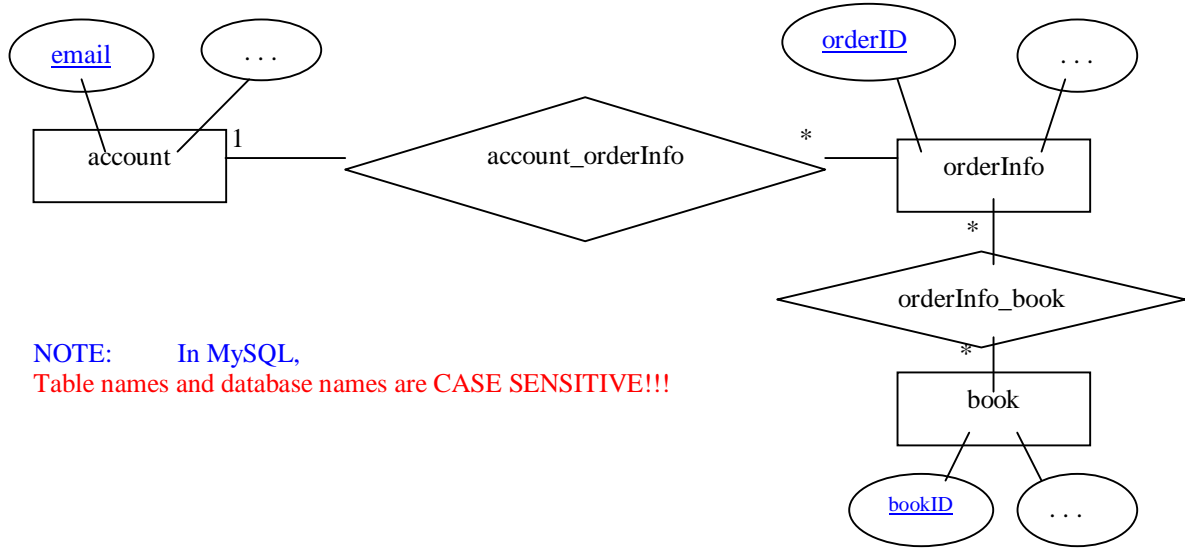
5. Database Design

It seems that we need five tables: Three entity tables, two relationship tables.

Entity: account, orderInfo, book

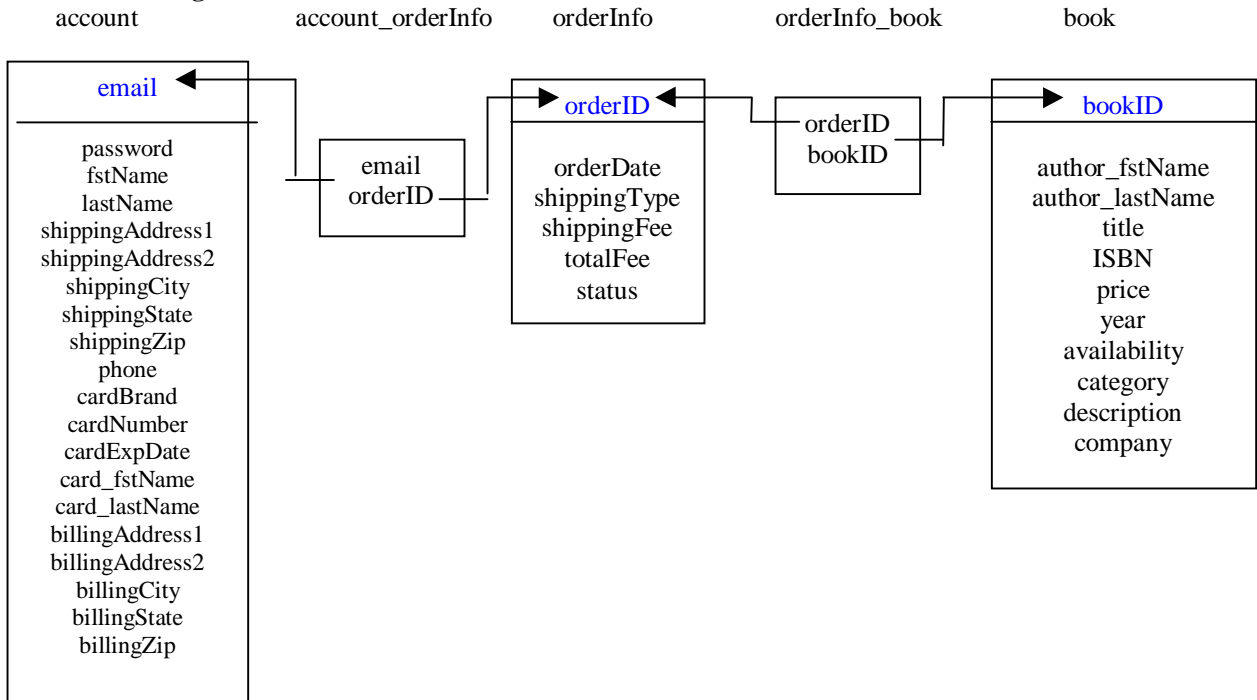
Relationship: account_orderInfo (1-many), orderInfo_book (many-many)

5.1 E-R diagram



NOTE: In MySQL, Table names and database names are CASE SENSITIVE!!!

5.2 Schema Diagram



5.3 MySQL script for creating tables

```
##### CS 591 Project #####
#
#
# File Name: createTables.sql
# Content: MySQL script for creating tables
#
# Usage: mysql> source createTables.sql;
#
#
# NOTE: Table names and database names in MySQL are
#       CASE SENSITIVE!!!
#
#####

##### Drop all tables first #####
drop table book;
drop table account;
drop table orderInfo;
drop table account_orderInfo;
drop table orderInfo_book;

##### Table book #####

CREATE TABLE book ( bookID VARCHAR(10) not NULL primary key,
                    author_lastName VARCHAR(50) not NULL,
                    author_fstName VARCHAR(50) not NULL,
                    title VARCHAR(200) not NULL,
                    ISBN VARCHAR(30) not NULL,
                    price FLOAT(6,2) default '0.00' NOT NULL,
                    year INT(4) default NULL,
                    availability VARCHAR(200) default NULL,
                    category VARCHAR(50) not NULL,
                    description VARCHAR(200) default NULL,
                    company VARCHAR(100) default NULL
                    );

##### Table account #####

CREATE TABLE account ( email VARCHAR(50) not NULL primary key,
                       password VARCHAR(50) not NULL,
                       fstName VARCHAR(50) not NULL,
                       lastName VARCHAR(50) not NULL,
                       shippingAddress1 VARCHAR(200) default NULL,
                       shippingAddress2 VARCHAR(200) default NULL,
                       shippingCity VARCHAR(100) default NULL,
```

```
shippingState CHAR(2) default NULL,  
shippingZip CHAR(5) default NULL,  
phone VARCHAR(20) default NULL,  
cardBrand VARCHAR(20) default NULL,  
cardNumber VARCHAR(16) default NULL,  
cardExpDate DATE default NULL,  
card_fstName VARCHAR(50) default NULL,  
card_lastName VARCHAR(50) default NULL,  
billingAddress1 VARCHAR(200) default NULL,  
billingAddress2 VARCHAR(200) default NULL,  
billingCity VARCHAR(100) default NULL,  
billingState CHAR(2) default NULL,  
billingZip CHAR(5) default NULL  
);
```

```
##### Table orderInfo #####
```

```
CREATE TABLE orderInfo ( orderID VARCHAR(20) not NULL primary key,  
orderDate DATE not NULL,  
shippingType CHAR(1) not NULL,  
shippingFee FLOAT(6,2) default '0.00' not NULL,  
totalFee FLOAT(6,2) default '0.00' not NULL,  
status CHAR(1) default 'N' not NULL  
);
```

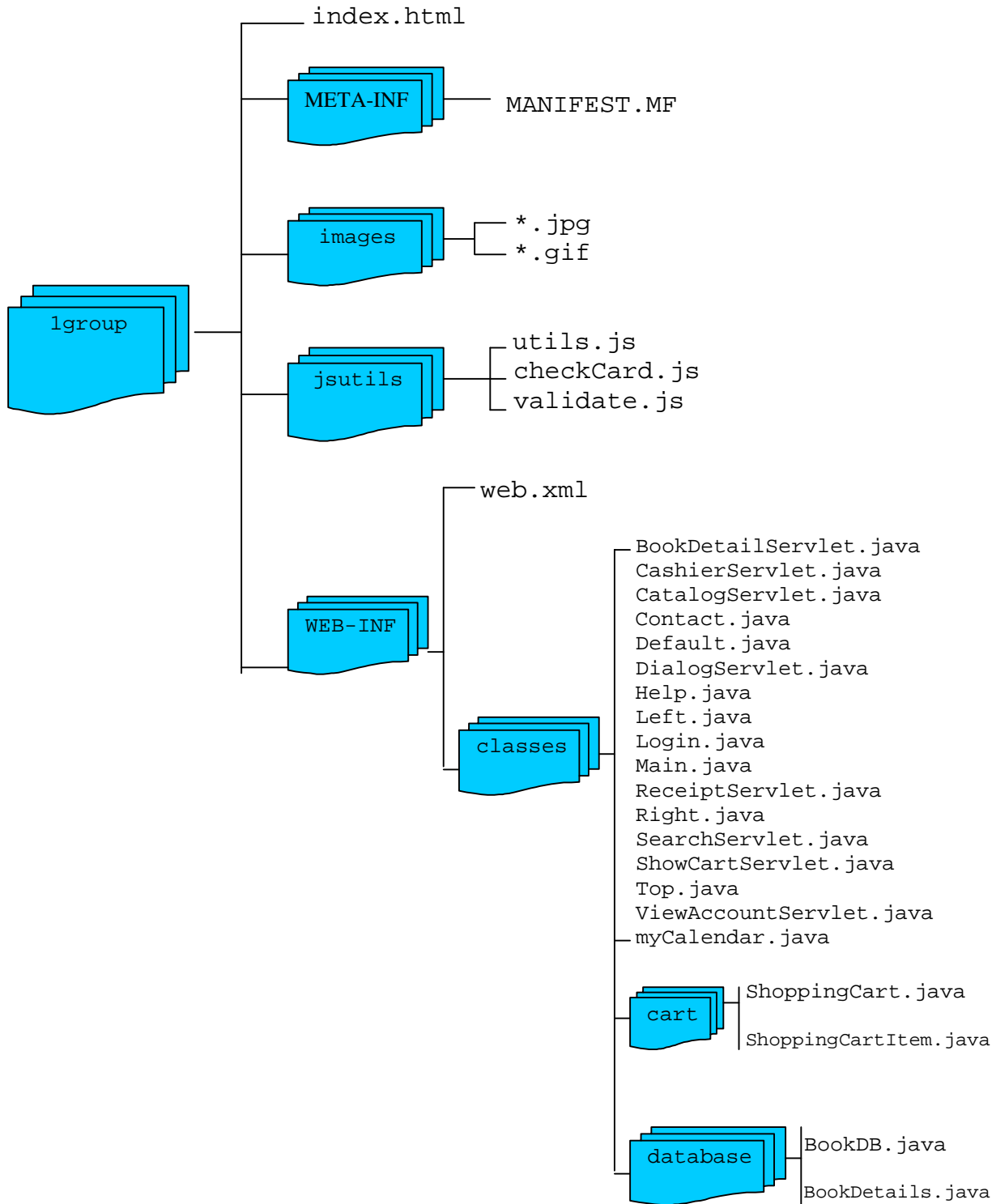
```
##### Table account_orderInfo #####
```

```
CREATE TABLE account_orderInfo ( email VARCHAR(50) not NULL,  
orderID VARCHAR(20) not NULL,  
PRIMARY KEY( email, orderID )  
);
```

```
##### Table orderInfo_book #####
```

```
CREATE TABLE orderInfo_book ( orderID VARCHAR(20) not NULL,  
bookID VARCHAR(10) not NULL,  
PRIMARY KEY( orderID, bookID )  
);
```

6. Directory Structure



7. Java Servlet Code

7.1 Servlets

- ◆ Default.java
- ◆ Top.java
- ◆ Left.java
- ◆ Right.java
- ◆ Main.java
- ◆ Login.java
- ◆ ViewAccountServlet.java
- ◆ Help.java
- ◆ Contact.java
- ◆ SearchServlet.java
- ◆ DialogServlet.java
- ◆ CatalogServlet.java
- ◆ BookDetailServlet.java
- ◆ ShowCartServlet.java
- ◆ CashierServlet.java
- ◆ ReceiptServlet.java

7.2 Utility classes

- ◆ myCalendar.java
- ◆ ShoppingCart.java
- ◆ ShoppingCartItem.java
- ◆ BookDB.java
- ◆ BookDetails.java

7.3 JavaScript

- ◆ checkCard.js
- ◆ validate.js
- ◆ utils.js