

Campus Second-hand Book Trading System

Report2

Software Engineering Course Project
Major: IMIS

Term leader: James

Term member: Anthony, Jay, Henry, Walker, Damon

Damon

Class Diagram and Interface Specification

And part of Project Management

James

System Architecture and System Design

And part of Project Management

Walker

Algorithms and Data Structures

And part of Project Management

Anthony

User Interface Design and Implementation

And part of Project Management

Henry

Design of Tests

And part of Project Management

Jay

Interaction Diagrams

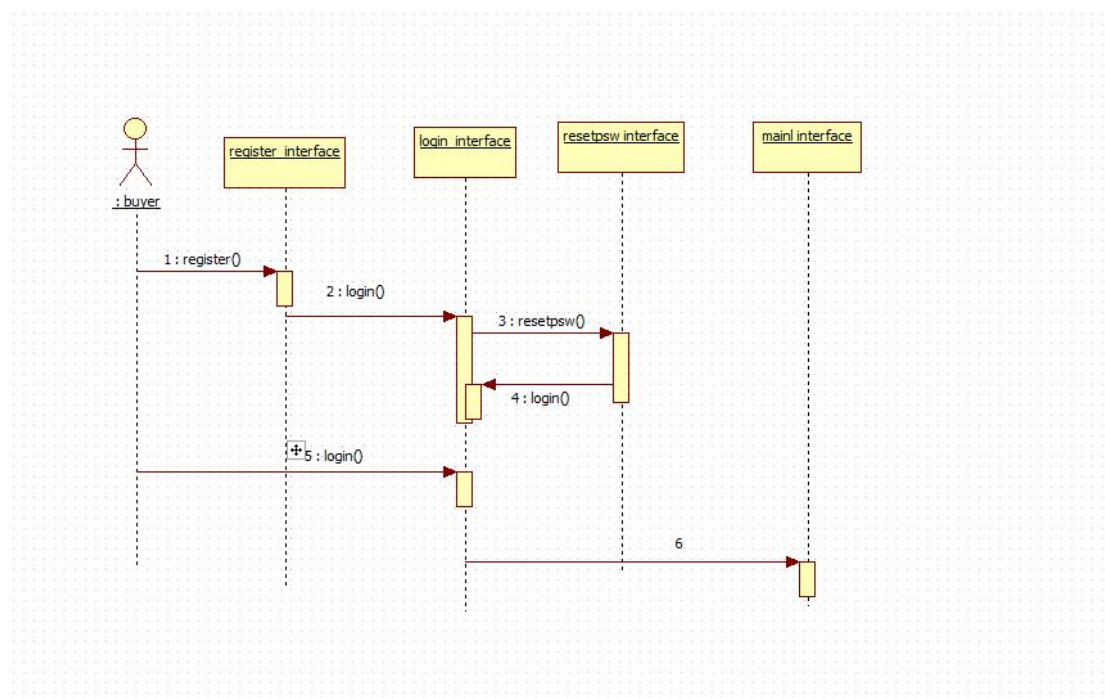
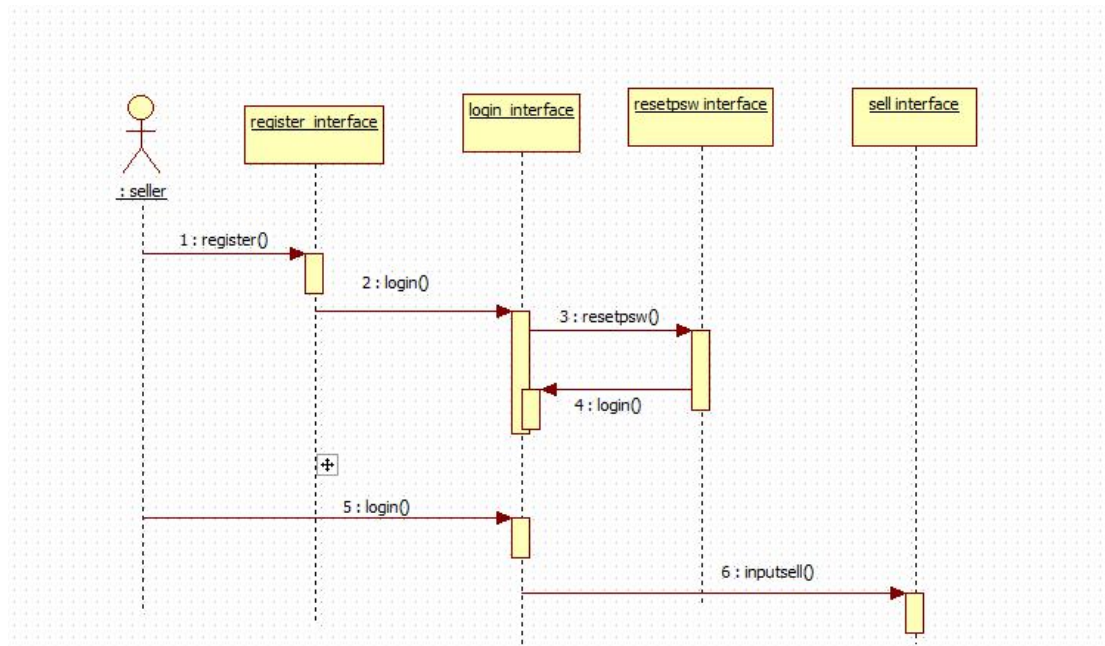
And part of Project Management

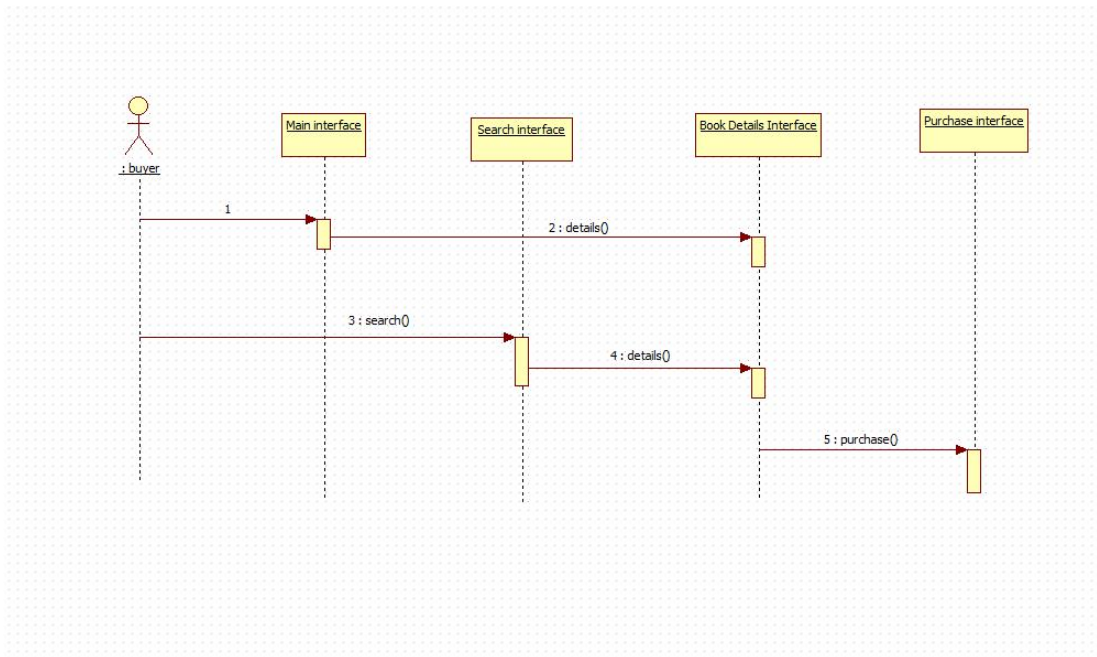
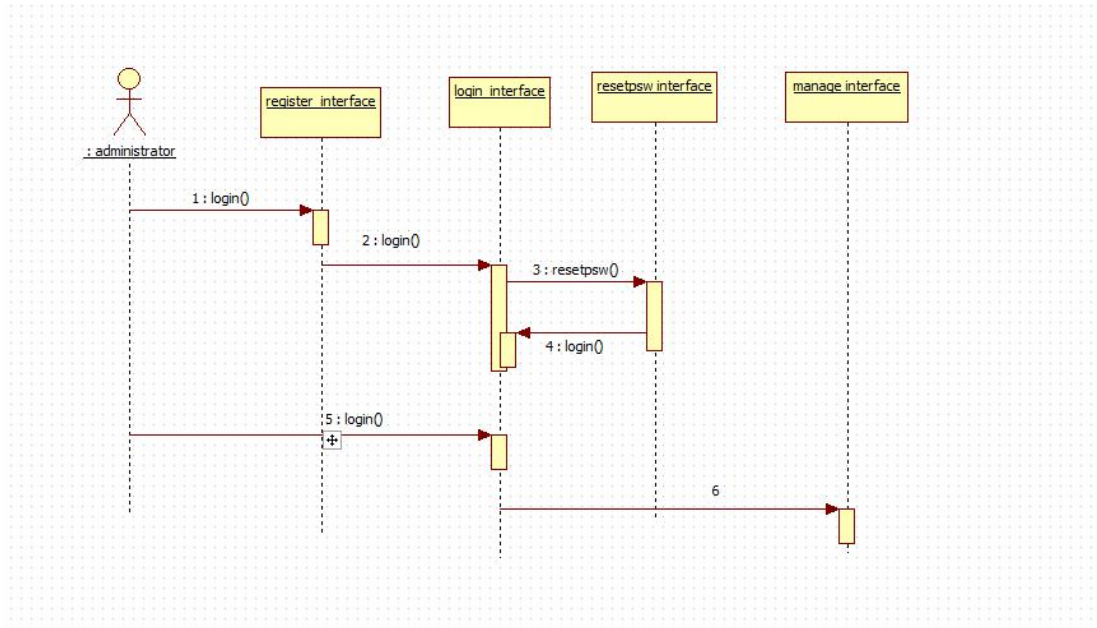
List

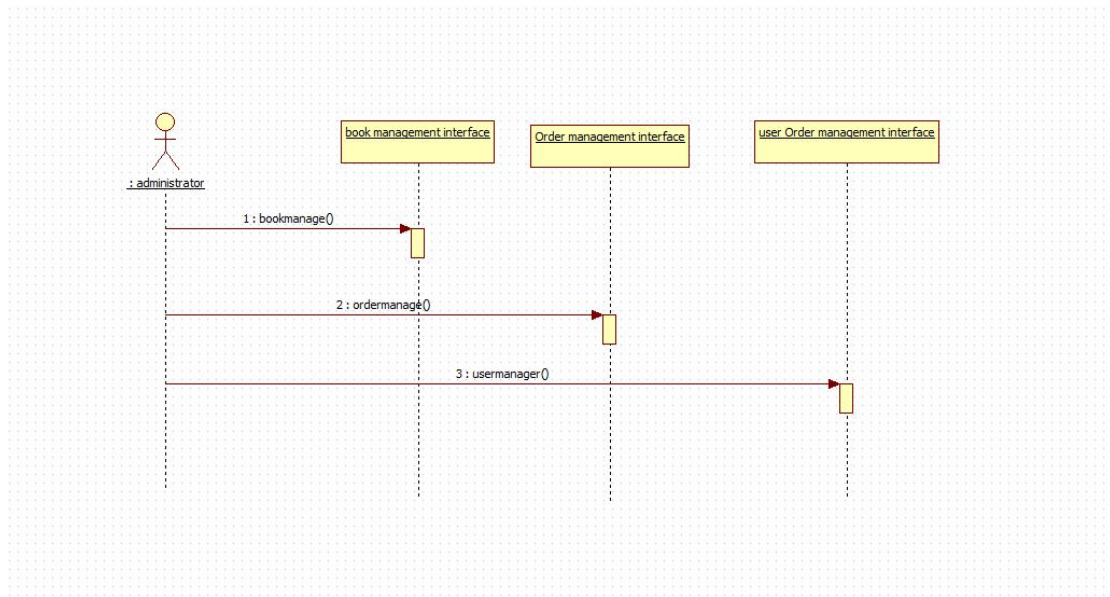
Report2.....	4
Section 1: Interaction Diagrams.....	4
Section 2: Class Diagram.....	7
Section 3: System Architecture.....	10
Section 4: Algorithms and Data Structures.....	16
Section 5: User Interface Design and Implementation.....	17
Section 6: Design of Tests.....	20
Project Management.....	23
References.....	24

Report2

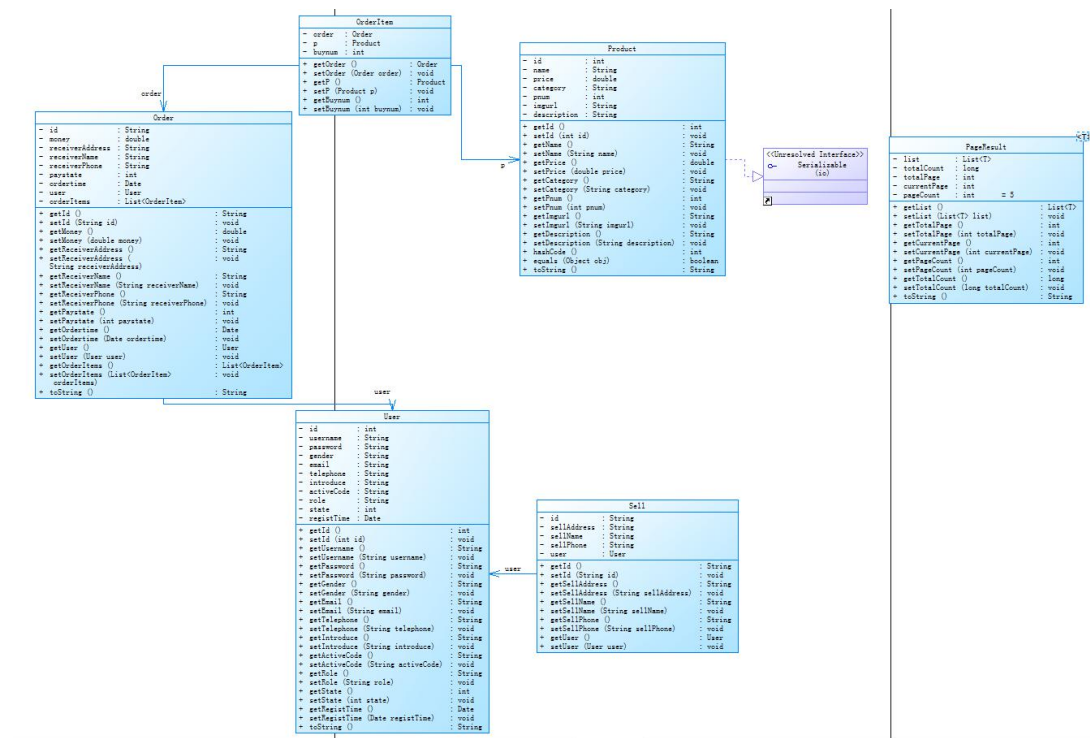
Section 1: Interaction Diagrams







Section 2: Class Diagram



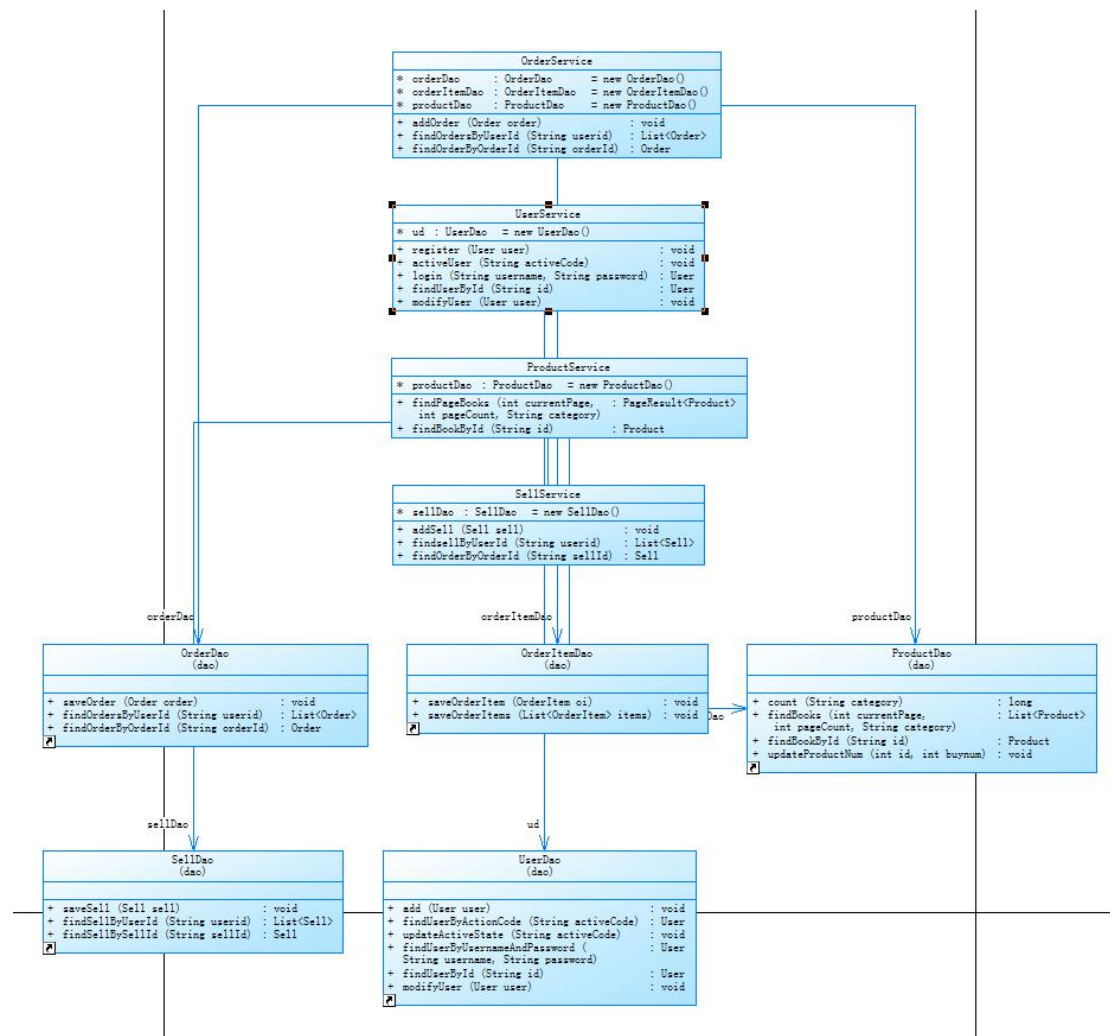
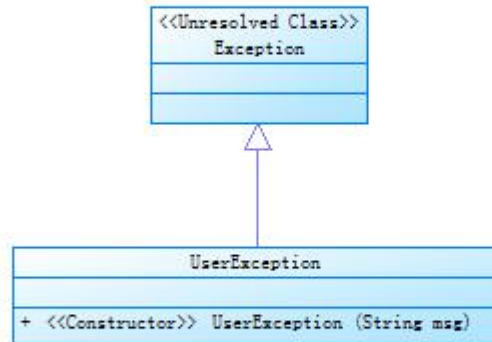
UserDao	
+ add (User user)	: void
+ findUserByActionCode (String activeCode)	: User
+ updateActiveState (String activeCode)	: void
+ findUserByUsernameAndPassword (String username, String password)	: User
+ findUserById (String id)	: User
+ modifyUser (User user)	: void

ProductDao	
+ count (String category)	: long
+ findBooks (int currentPage, int pageCount, String category)	: List<Product>
+ findBookById (String id)	: Product
+ updateProductNum (int id, int buynum)	: void

OrderDao	
+ saveOrder (Order order)	: void
+ findOrdersByUserId (String userid)	: List<Order>
+ findOrderByOrderId (String orderId)	: Order

SellDao	
+ saveSell (Sell sell)	: void
+ findSellByUserId (String userid)	: List<Sell>
+ findSellBySellId (String sellId)	: Sell

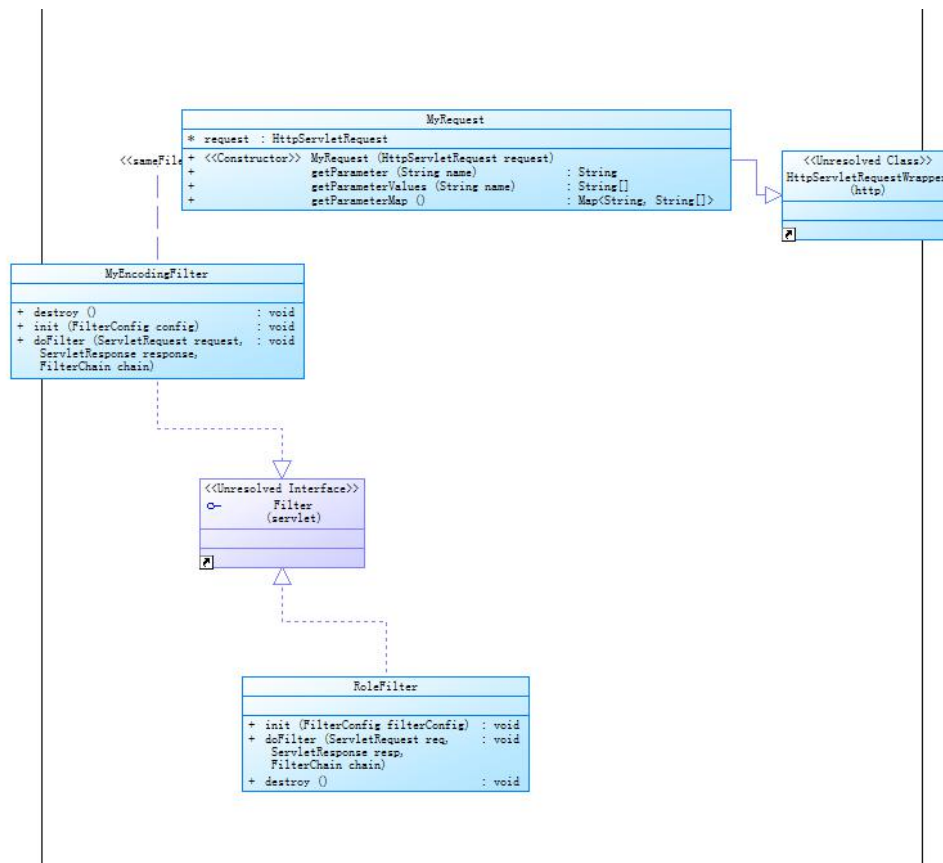
OrderItemDao	
+ saveOrderItem (OrderItem oi)	: void
+ saveOrderItems (List<OrderItem> items)	: void

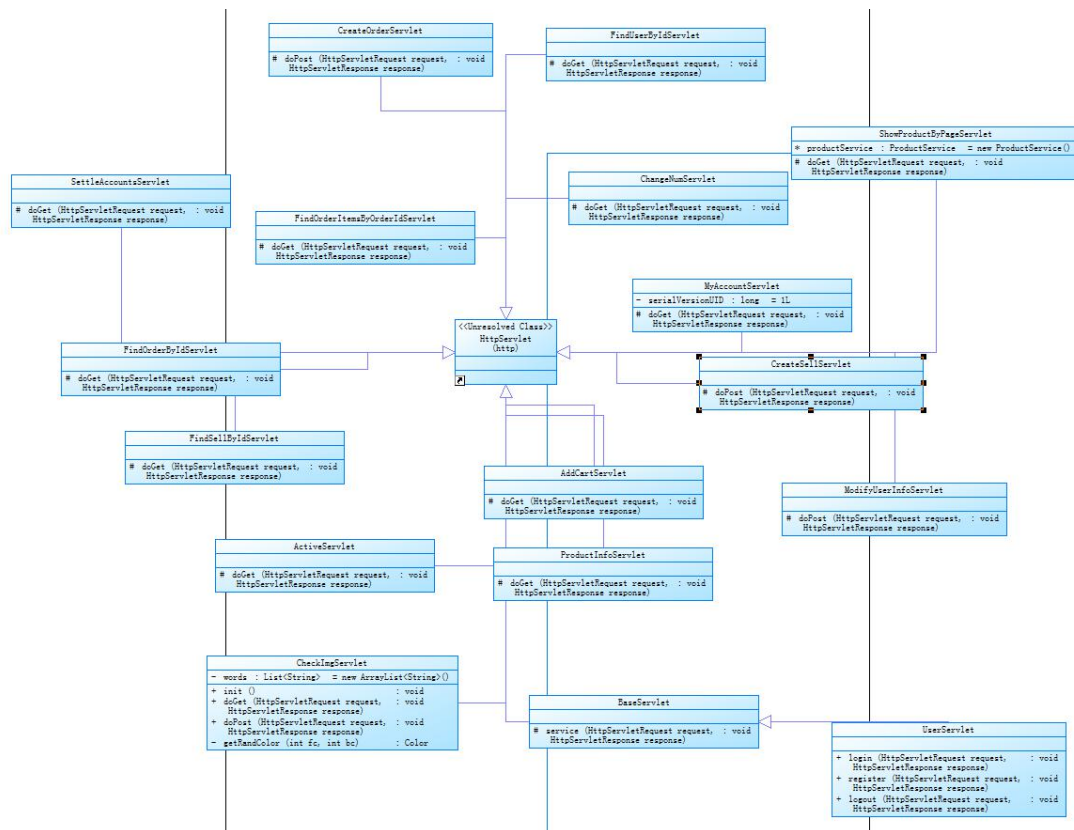


ManagerThreadLocal	
- tl : ThreadLocal<Connection>	= new ThreadLocal<Connection>()
+ getConnection ()	: Connection
+ beginTransaction ()	: void
+ commitTransaction ()	: void
+ rollback ()	: void
+ close ()	: void

CSFOUtils	
- ds : DataSource	= new ComboPooledDataSource()
+ getDataSource ()	: DataSource
+ getConnection ()	: Connection
+ closeAll (Connection conn,	: void
Statement statement,	
ResultSet resultSet)	

SendJMail	
+ sendMail (String email, String emailMsg)	: boolean



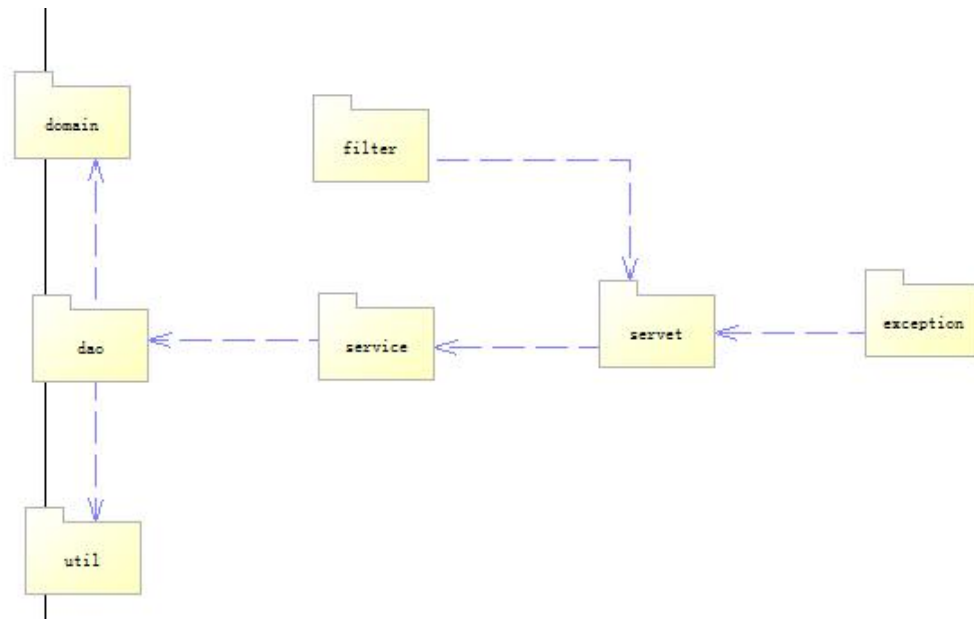


Section 3: System Architecture

A.System Architecture

Our project uses the client/server style in the software architecture.

B. Identifying Subsystems

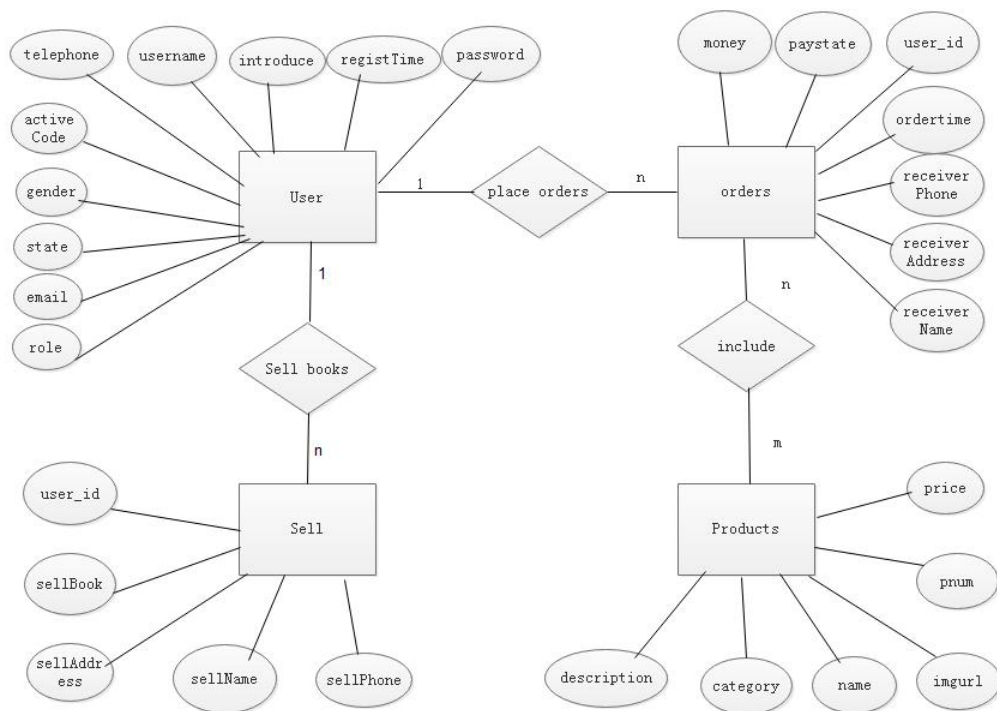


C. Mapping Subsystems to Hardware

Yes, our server side runs on its own server, and the client side is the browser of users who visit our web pages.

D. Persistent Data Storage

Yes, MYSQL database is used



create database bookstore

user

```

CREATE TABLE `user` (
  `id` INT(11) AUTO_INCREMENT,
  `username` VARCHAR(20) ,
  `PASSWORD` VARCHAR(20) ,
  `gender` VARCHAR(10) ,
  `email` VARCHAR(50) ,
  `telephone` VARCHAR(20) ,
  `introduce` VARCHAR(100),
  `activeCode` VARCHAR(50) ,

```

```
`state` INT(11) ,  
`role` VARCHAR(10) DEFAULT '普通用户',  
`registTime` TIMESTAMP ,  
PRIMARY KEY (`id`)  
)
```

products

```
CREATE TABLE `products` (  
  `id` VARCHAR(100) ,  
  `name` VARCHAR(40) ,  
  `price` DOUBLE ,  
  `category` VARCHAR(40) ,  
  `pnum` INT(11) ,  
  `imgurl` VARCHAR(100) ,  
  `description` VARCHAR(255) ,  
  PRIMARY KEY (`id`)  
)
```

orders

```
CREATE TABLE `orders` (  
  `id` VARCHAR(100) ,  
  `money` DOUBLE ,  
  `receiverAddress` VARCHAR(255) ,  
  `receiverName` VARCHAR(20) ,
```

```

`receiverPhone` VARCHAR(20) ,

`paystate` INT(11) ,

`ordertime` TIMESTAMP ,

`user_id` INT(11) ,

PRIMARY KEY (`id`),

FOREIGN KEY (`user_id`) REFERENCES `user` (`id`)

)

```

orderitem

```

CREATE TABLE `orderitem` (

`order_id` VARCHAR(100) ,

`product_id` VARCHAR(100),

`buynum` INT(11) ,

PRIMARY KEY (`order_id`, `product_id`),

FOREIGN KEY (`order_id`) REFERENCES `orders` (`id`),

FOREIGN KEY (`product_id`) REFERENCES `products`

(`id`)

)

```

sell

```

CREATE TABLE `sell` (

`id` VARCHAR(100) ,

`user_id` INT(11),

`sellBook` VARCHAR(20) ,

```

```
`sellAddress` VARCHAR(255) ,  
`sellName` VARCHAR(20) ,  
`sellPhone` VARCHAR(20) ,  
PRIMARY KEY (`id`),  
FOREIGN KEY (`user_id`) REFERENCES `user` (`id`)  
)
```

E. Network Protocol

We use the HTTP protocol because our website requires users to use browsers as clients. In the communication process, we need to use the HTTP protocol to transfer requests and data.

F. Global Control Flow

1.It is an event-driven system that waits for events in a loop, and each user can send different requests to the server to obtain different event responses.

2. There is no timer, it is an event response system.

3. Multithreading is not used

g.Hardware Requirements

Hardware requirements:

Server side: Pentium4 1.6Hz and above processors,

256MB or more memory, hard disk space increases or decreases depending on the amount of user data, to be networked.

Client: IE8 and above and various mainstream browsers, to be networked

software requirement:

Server side: TOMCAT,MYSQL8.0 and above are required

Section 4: Algorithms and Data Structures

a. Algorithms

No other complicated algorithms are used.

b. Data Structures

The primary data structure we are using is arraylist. This is because it has a simple $O(n)$ look up time which does not add much delay to our performance. More so, this data structure is very flexible. It is very easy to add onto an arraylist as it only takes $O(1)$ time. Arraylists are also compatible with listview. This is very important as we are constantly using listviews to display menu, ordered items, and other information. An arraylist can be easily passed into an arrayadapter to be displayed in a listview. Other data structures such as hash tables or linked lists require more work to

convert. We wanted a list type structure also due to the compatibility with the database. Since our database is SQL based, it can be looked at as a list. The values on the tables in our database can be very easily read into lists. It also does not take much effort to write the list back into the database

Section 5: User Interface Design and Implementation

User Interface Design and Implementation

No major changes have been made

Registration interface

The screenshot displays the 'New Membership Registration' form on the eSHOP website. The page features a green header with navigation links: Life, Management, social sciences, learning, juvenile, Art, Original edition, Examination, Computer, and Catalogue of all commodities. A search bar is located on the right side of the header. The registration form itself is a light yellow box with the following fields and labels:

- Email: (with a note: Please enter a valid email address)
- UserName: (with a note: User name set to at least 6 bits)
- password: (with a note: Password setting at least 6 bits)
- Repeat the password:
- Gender: ☒ male ☐ female
- Contact number:
- Personal introduction:

Below the registration fields is a section titled '注册校验' (Registration Verification) with the label 'Enter the check code:' and a captcha image. A link 'Can't see clearly and change one' is provided next to the captcha. At the bottom of the form is an orange button labeled 'Agree and submit'.

The footer of the page includes the eSHOP logo, a 'CONTACT US' link, and a copyright notice: 'COPYRIGHT 2008 © eShop All Rights RESERVED.'.


Login interface

eSHOP [I want to sell](#) | [Shopping Cart](#) | [My account](#) | [registration](#)

Life Management social sciences learning juvenile Art Original edition Examination Computer Catalogue of all commodities

Search

home page » Personal user login



用户名:

密 码:

Please register

Register new members, enjoy more preferential prices!

Thousands of books for you to choose! Register to enjoy rich discounts and concessions, cheap and good goods! More than ten thousand books are for your choice.

Super popularity community! Great activities every day. Business is more secure! Allpay trading is super secure.

Book catalog interface

eSHOP [I want to sell](#) | [Shopping Cart](#) | [My account](#) | [registration](#)


Life Management social sciences learning juvenile Art Original edition Examination Computer Catalogue of all commodities

Search


home page > > List of books

Catalogue

PRODUCT LIST



书名:java
Selling price:168.88




书名:javaee
Selling price:168.88

第1页共 1页

eSHOP CONTACT US
COPYRIGHT 2019 © BookStore All Rights RESERVED.

Shopping cart interface


eSHOP

 [I want to sell.](#) | [Shopping Cart](#) | [My account](#) | [registration](#)

LifeManagementsocial scienceslearningjuvenileArtOriginal editionExaminationComputerCatalogue of all commodities

SearchSearch

home page > Shopping Cart



Shopping process

shopping cartloginSettlement centerConfirm order

Serial number	Trade name	Price	Number	Stock	Subtotal	exit
						Total: 0


Continue shoppingSubmit order

eSHOP

CONTACT US
COPYRIGHT 2019 © BookStore All Rights RESERVED.

Selling interface

eSHOP

 [I want to sell.](#) | [Shopping Cart](#) | [My account](#) | [registration](#)

LifeManagementsocial scienceslearningjuvenileArtOriginal editionExaminationComputerCatalogue of all commodities

SearchSearch

首页 > 出售信息

Names of books sold

Seller's name

Seller's address

Seller's mobile phone number

Confirm order

eSHOP

CONTACT US
COPYRIGHT 2019 © BookStore All Rights RESERVED.

Personal center interface

The screenshot displays the eSHOP website's personal center interface. At the top, the eSHOP logo is on the left, and navigation links for 'I want to sell', 'Shopping Cart', 'My account', and 'registration' are on the right. Below this is a green navigation bar with categories: Life, Management, social sciences, learning, juvenile, Art, Original edition, Examination, Computer, and Catalogue of all commodities. A search bar is located below the navigation bar. The main content area shows the 'My account' section on the left with links for 'User Information Modification', 'Order inquiry', and 'exit'. The 'User Information Modification' page is active, displaying the user's email (1481734165@qq.com), member name (huzhiwei), and contact information (18639760925). It includes fields for changing the password and a 'sure' button. The footer contains the eSHOP logo, 'CONTACT US', and a copyright notice for 2019.

home page > My account > User Information Modification

My account

- User Information Modification
- Order inquiry
- exit

Email: 1481734165@qq.com

Member name: huzhiwei

Change Password: Password Settings at least 6 bits

Repeat the password:

Gender: ☒ male ☐ female

Contact information:

sure

eSHOP CONTACT US
COPYRIGHT 2019 © BookStore All Rights RESERVED.

Section 6: Design of Tests

The following are the test cases to be used for unit testing:

TC-1: Test registration function

TC-2: Test login function

TC-3: Test selling function

TC-4: Test the function of searching books

TC-5: Test shopping cart function

TC-6: Test ordering function

TC-7: Test modification personal information function

TC-8: Test View Order Function

Test Case Identifier:TC-1 Use Case Tested: UC-3 Pass/Fail Criteria: Test passed whether the user can register with e-mail, user name, password, gender, cell phone number and personal profile. If the user can register successfully without entering information, the test will fail. Input Data: email,username, password,repasword,gender,contact number,personal profile		
Test Procedure:	Expected Result	Actual Result
1. Don't enter anything and click Register	Registration failed	Prompt Please Enter Email
2. Don't enter email, user name, password or duplicate password	Registration failed	Prompt for information not entered
3. Enter complete information	Registration successful	Enter the registration success page

Test Case Identifier:TC-2 Use Case Tested: UC-5 Pass/Fail Criteria: Test whether the user can log in Input Data: username, password		
Test Procedure:	Expected Result	Actual Result
1. Do not enter username or password and Click login.	Login failed	The pop-up prompts that the user name or password is incorrect.
2. Enter username and password and Click login.	Login successful	Login Successful Return to Home Page

Test Case Identifier:TC-3 Use Case Tested: UC-14 Pass/Fail Criteria: Test whether users can sell goods. Input Data: Names of books sold, Seller's name, Seller's address, Seller's mobile phone number		
Test Procedure:	Expected Result	Actual Result
1. Do not enter Names of books sold ,Seller's name, Seller's address or Seller's mobile phone number and click Confirm order	The sale failed.	Prompt Please Enter Information Not Entered
2. Enter Names of books	The sale failed.	Jump to the successful

sold, Seller's name, Seller's address, Seller's mobile phone number and click Confirm order		sale page
---------------------------------------------------------------------------------------------------------	--	-----------

Test Case Identifier:TC-4 Use Case Tested: UC-1 Pass/Fail Criteria: Test whether users can search for books 搜索书籍 Input Data: book name		
Test Procedure:	Expected Result	Actual Result
1.Enter book name and click Search	Show corresponding books	Show corresponding books

Test Case Identifier:TC-5 Use Case Tested: UC-6, UC-8,UC-9 Pass/Fail Criteria: Test whether users can modify and delete shopping carts or place orders.		
Test Procedure:	Expected Result	Actual Result
1. Click “+”or “-”	Changes in the number of goods	Changes in the number of goods
2. Click “X”	Delete goods	Do you want to delete the goods when it pops up
3. Click “Submit Order”	Enter the order placing interface	Enter the order placing interface

Test Case Identifier:TC-6 Use Case Tested: UC-7 Pass/Fail Criteria: Test whether users can place orders . Input Data: receiver address,receiver Name,receiver Phone		
Test Procedure:	Expected Result	Actual Result
1. Do not enter receiver address,receiver Name and receiver Phone.	Failed to place order	Prompt for information not entered
2. Enter receiver address , receiver Name and receiver Phone	Order placed successfully	Jump to the successful interface for ordering

Test Case Identifier:TC-7 Use Case Tested: UC-10 Pass/Fail Criteria: Test whether users can modify personal information . Input Data: email,username, password,repasword,gender,contact number,personal profile		
Test Procedure:	Expected Result	Actual Result
1. Do not enter Names of books sold ,Seller's name, Seller's address or Seller's mobile phone number and click sure	Modification failed	Jump to successful modification interface
2. Enter Names of books sold ,Seller's name, Seller's address or Seller's mobile phone number and click sure	Modification successful	Jump to successful modification interface

Test Case Identifier:TC-8 Use Case Tested: UC-11, UC-12, UC-13 Pass/Fail Criteria: Test whether users can manage orders		
Test Procedure:	Expected Result	Actual Result
1. Click “check”	Jump to the order details interface	Jump to the order details interface
2. Click “delete”	Order deleted	Order deleted

Project Management

A. Merging the Contributions from Individual Team Members

The insufficient number of use cases leads to the failure of the website to implement many functions normally. So we added enough use cases.

B. We have implemented use cases such as login, registration, forgetting passwords and logout, and we are dealing with the problem of Book Mall (how to call the database to display the information of books).

We are discussing the functional details and designing the database.

C. The use case of the mall is expected to be completed next week

Complete evaluation and shopping cart use cases next week, then it begins to enter the system testing phase.

D. Breakdown of Responsibilities

James and Damon responsible for developing, coding, and testing.

Jay and Anthony responsible for coordinate the integration.

Henry and Walker responsible for perform and integration testing

References

James: <https://github.com/pluuuuus-ultra/James.git>

Damon: <https://github.com/pluuuuus-ultra/Damon.git>

Jay: <https://github.com/pluuuuus-ultra/Jay.git>

Anthony: <https://github.com/pluuuuus-ultra/Anthony.git>

Henry: <https://github.com/pluuuuus-ultra/Henry.git>

Walker: <https://github.com/pluuuuus-ultra/Walker.git>