MRSK2

UNIVERSITY OF TEXAS AT ARLINGTON CSE 5324 SOFTWARE ENGINEER 1 - SUMMER 2014


BOOKHAUS

TABLE OF CONTENT

PROJECT DESCRIPTION

We will be building a simple Online Web Application for a bookstore by using java based technologies using Model View Controller (MVC) architecture. We will implementing a simple Agile process during class. The instructor will act like a scrum master and/or customer (for requirements) throughout the process.

TECHONOLOGIES USED

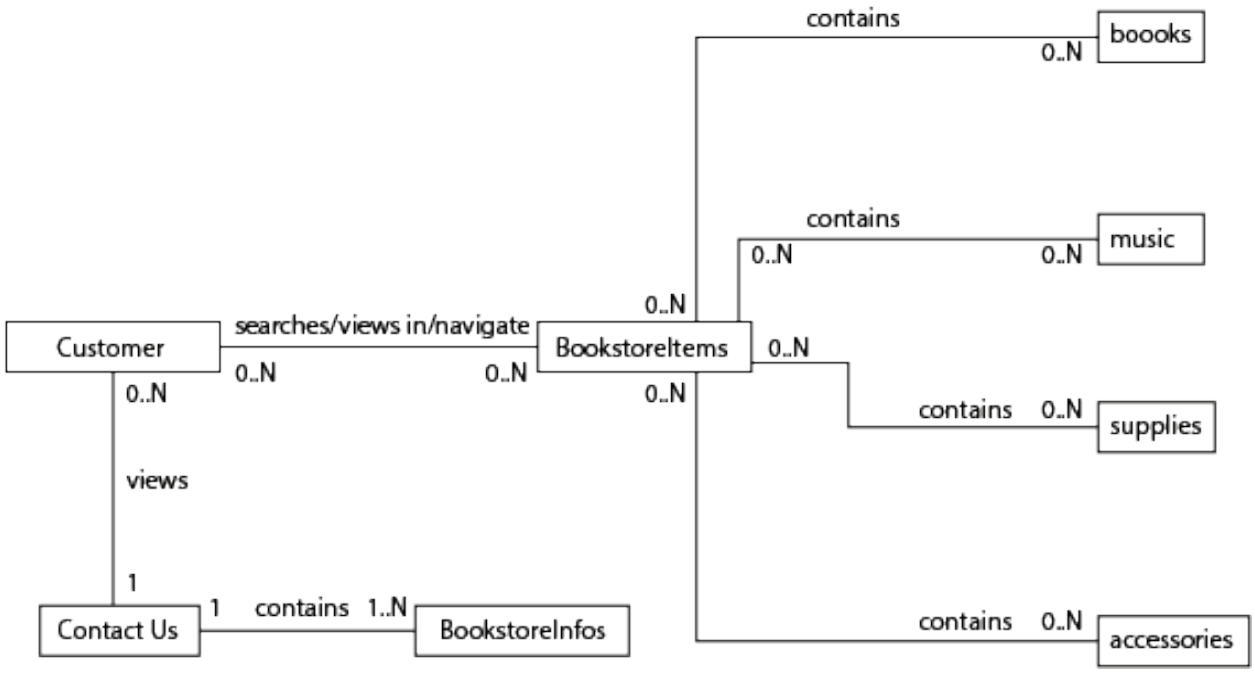
View (User Interface): HTML, CSS, Bootstrap, Javascript
Model and Controler: Java, JSP, Servlet
Backend: MYSQL

PROJECT REQUIREMENTS

The customer wants to grow their business and believes having an “online presence” will help them. The customer wants you to build a website for them. At this points, the initial requirements the manager gathered are:

1. Product search (basic and advanced).
    a. In basic search, a textbox where user can enter any search text.
    b. In advance search, allow users to choose the product attributes (e.g. product name, size, price, etc) to search and display, along with options on how the results are to be displayed – in ascending or descending order.
2. Display clickable listing of results.
    a. Users can click one result and it brings them to a new page for that particular product.
3. Navigation pane on the left side of the UI, for different product category. Each clickable link will display all the products of the category.
4. In the home page and all subsequent pages, a button/ULF link that says “Store Location” must be visible. When a user clicks on that button/link, a page displays informatoin for all of the stores. Store information to include: address, phone number, store hours. Put five stores. If the stores are in the same city, distinguish the stores by City-Street. E.g “Arlington-Cooper” and “Arlington-Abram” (Additional Reqirement After Iteration 1)
5. In the description page, list all the stores along with the quantity available for that product. If a store does not have the product, display 0 as the quantity. (Additional Reqirement After Iteration 1)

DOMAIN MODEL



Customer can search or navigate through the book store items on the application. The bookstore item contains books, music, office supplies, and accessories. Customer will also be able to view the contact us page to get the bookstore information including name, address, hours of operation, phone number and email address.

USE CASE 01 - SEARCH BASIC

Use Case UC#01	Search Basic
Scope	Search
Level	User-goal
Primary Actor	Customer

Stakeholders & Interests

- Customer: Finding product with keyword easily without any other specific knowledge.
- Company: Display all the possible products correctly and efficiently.

Pre-Condition

1. Customer is on the application and found the Search box.
2. Customer is typing the keyword in ‘English’ Language.

Post-Condition

1. Desired product and its description is displayed on the web page
2. If product is not found then it should display some error message.
3. Most relevant search should appear at the first.

Main Success Scenario

1. Customer gets on the web application.
2. Customer types some keyword in search box.
3. Customer clicks on Search button.
5. Web page shows most relevant search results with its name, price, pictures etc.
6. Customer repeat step 2 to 5 for more searches.
7. Customer leaves web application happily with the found product in mind.

Extensions

a. No results found.

1. If no results found in database for product then web page will show ‘No result found’ error message.
2. Right below the error message web page will ask customer ‘Did you mean’ option for some other relevant searches.
3. If customer clicks on one of those options then it will again go to step 4 of main success scenario.

USE CASE 02 - SEARCH ADVANCE

Use Case UC#02	Search Advance
Scope	Search
Level	User-goal
Primary Actor	Customer

Stakeholders & Interests

- Customer: wants to search specific details about products so that he can find result according to their particular need.
- Company: wants to give best search result to their customers to satisfy their need.

Pre-Condition

1. Customer is on the application and has navigated to the Advance Search form by clicking the link near Basic Search box.
2. Customer has a basic idea about the product attributes and values for his query.

Post-Condition

1. Customer can see list of available products on the page.

Main Success Scenario

1. Customer clicks on the advance search button.
2. Customer selects a category and the desired checklist then enter search keyword.
3. System provides a list of the matching products on the page to the customer.

Extensions

a. Attribute value format error:

1. Customer enters value of the product attribute in the invalid format. For example, customer enters invalid year or negative price value.
2. System validates the input values entered on the form and shows the error message “<attribute name> is in invalid format” in case of input validation error.
3. Customer corrects the value to match the required attribute format and searches again.

b. No product found:

1. If no matching product is available in the database, the system shows the message “No matching product found”.
2. Customer can update the search query and searches again.

c. System timeout error:

1. If system times out while searching the product, the error message “System has timed out. Please try again.” is displayed on the web page.
2. Customer can click on the Search button again to re-execute the query.

USE CASE 03 - NAVIGATE ITEMS

Use Case UC#03	Navigate Items
Scope	Navigation
Level	User-goal
Primary Actor	Customer

Stakeholders & Interests

- Customer – Wants to browse through the products and page easily and efficiently.
- Company - Wants the products to be displayed in a correct way with correct description, details information and price information.

Pre-Condition

1. Customer is on the application and found the navigation bar.

Post-Condition

1. Desired product is found by browsing through the list of product in different categories.

Main Success Scenario

1. Customer gets on the web application and located the navigator on the left
2. Customer start clicking on certain category
3. Browser shows product list with picture, title, and price in selected category per page.

USE CASE 04 - GET PRODUCT DETAILS

Use Case UC#05	Get Product Details
Scope	Search
Level	User-goal
Primary Actor	Customer

Stakeholders & Interests

- Customer – wants to see product details without affecting search results.
- Company – Provides convenient and easy way to get complete details about product to customer.

Pre-Condition

1. Customer has navigated to the search result page and is able to see the desired product in the list.

Post-Condition

1. Customer is able to see the detailed description of the product like name, author, price, year of publication, ISBN, subject, copies left at each store etc.

Main Success Scenario

1. Customer clicks on the particular product in the product list.
2. Systems redirects customer to the detailed page of the product.
3. Customer sees the detailed description of the product.
4. Customer sees how many copies of items are left in each store.

Extensions

1. Customer clicks on the particular product in the product list.
2. Systems redirects customer to the detailed page of the product.
3. Customer sees the detailed description of the product.

USE CASE 05 - CONTACT US

Use Case UC#05	Contact Us
Scope	Contact
Level	User-goal
Primary Actor	Customer

Stakeholders & Interests

- Customer: Finding store location and information.
- Company: Display all store location and information correctly for the customer to find them easily.

Pre-Condition

1. Customer is on the application and found the Contact Us.

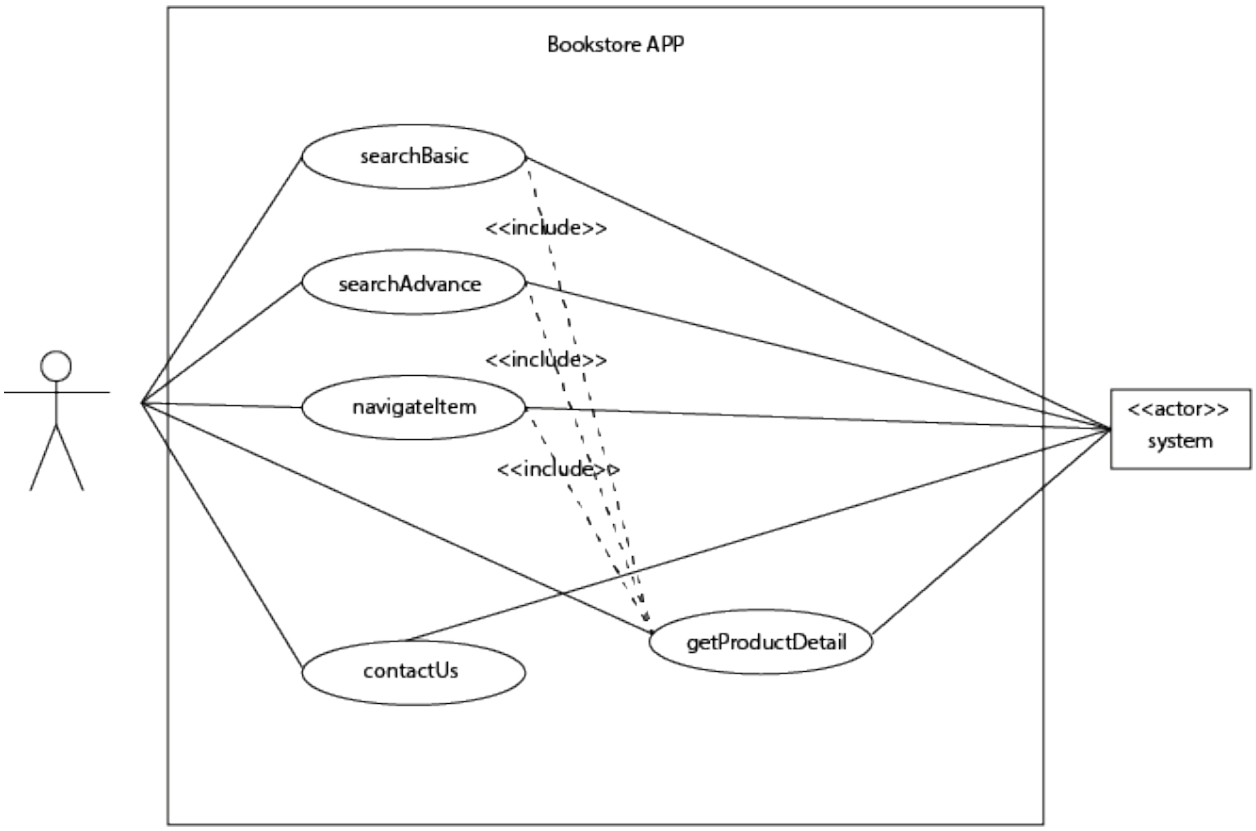
Post-Condition

1. Customer found the correct stores information.

Main Success Scenario

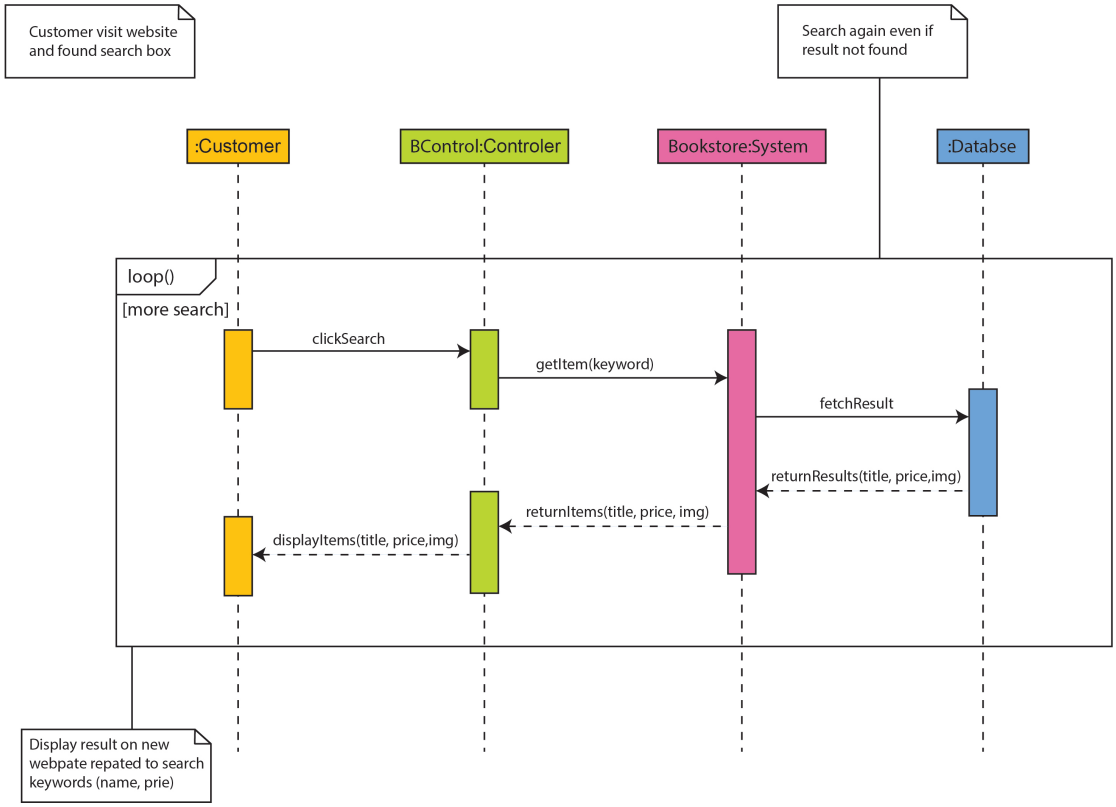
1. Customer gets on the web application.
2. Customer clicks on the contact us button.
3. Store locations are displayed in the browser.

USE CASE DIAGRAM



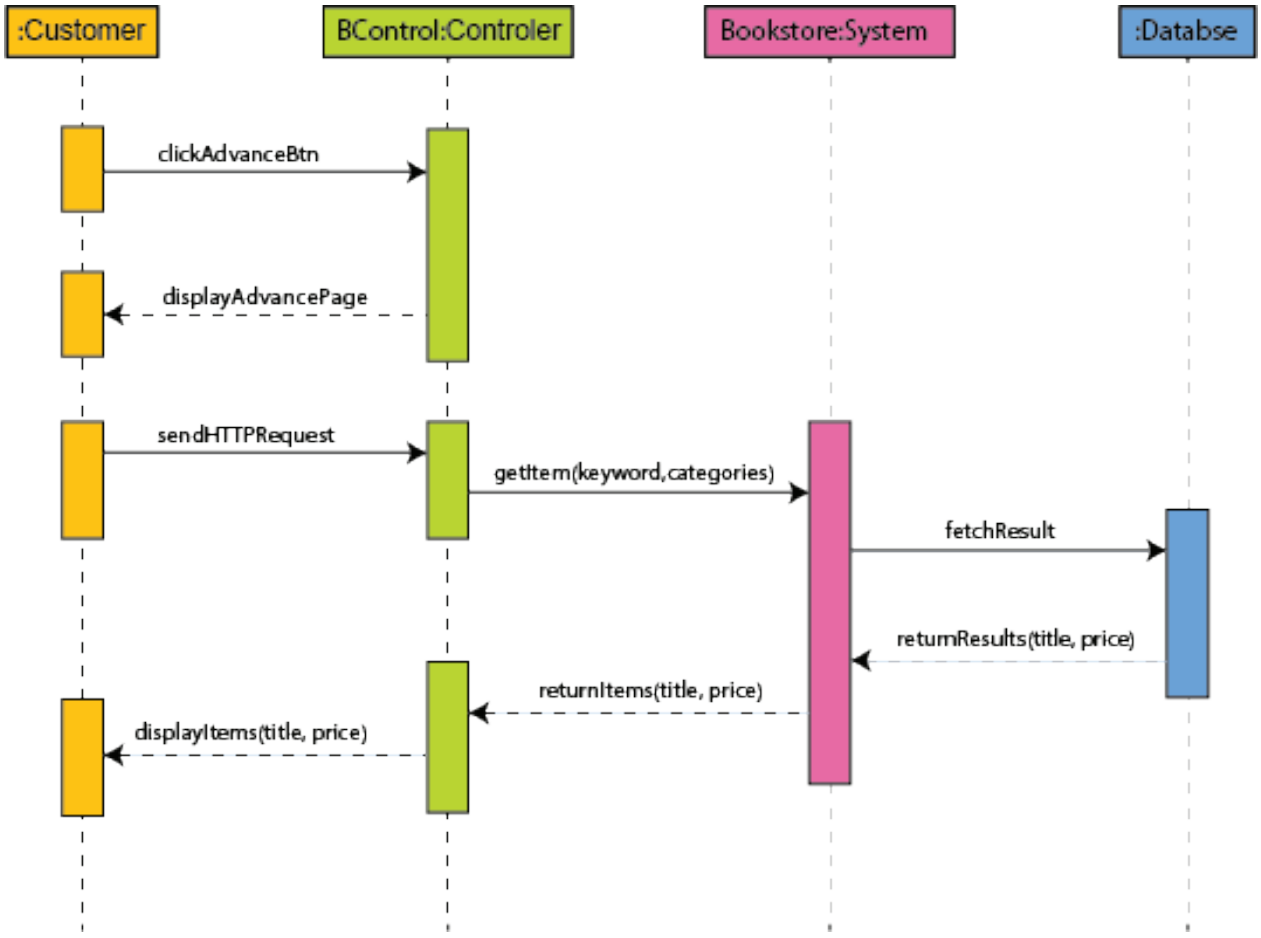
The bookstore APP domain contains five use cases which are searchBasic, searchAdvance, navigateItem, contactUs and getProductDetails. The actor interact with all of them directly, and they all interact with the system. Furthermore, client can access the getProductDetail use case from the search-Basic, searchAdvance, and navigateItem use case.

SEQUENCE DIAGRAM 01 - SEARCH BASIC



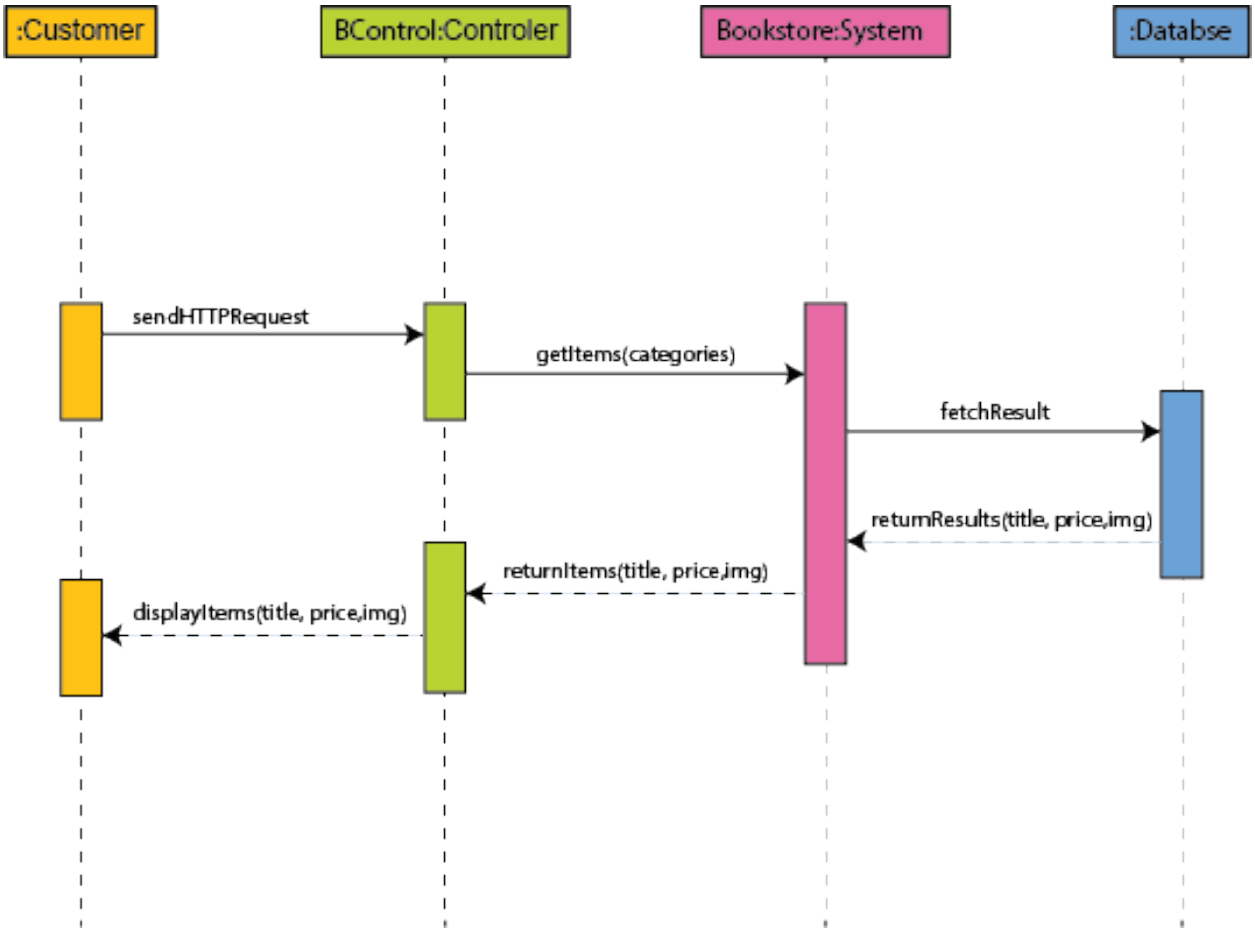
The customer enter the keyword and click on the search button to start the search sequence. Then the controler get the request from the GUI and send the corrosponding method with request to the business logic (system). The systen then fetch result from the database. After the system get the result back, it will return the result to the controler which then return the result to the GUI.

SEQUENCE DIAGRAM 02 - SEARCH ADVANCE



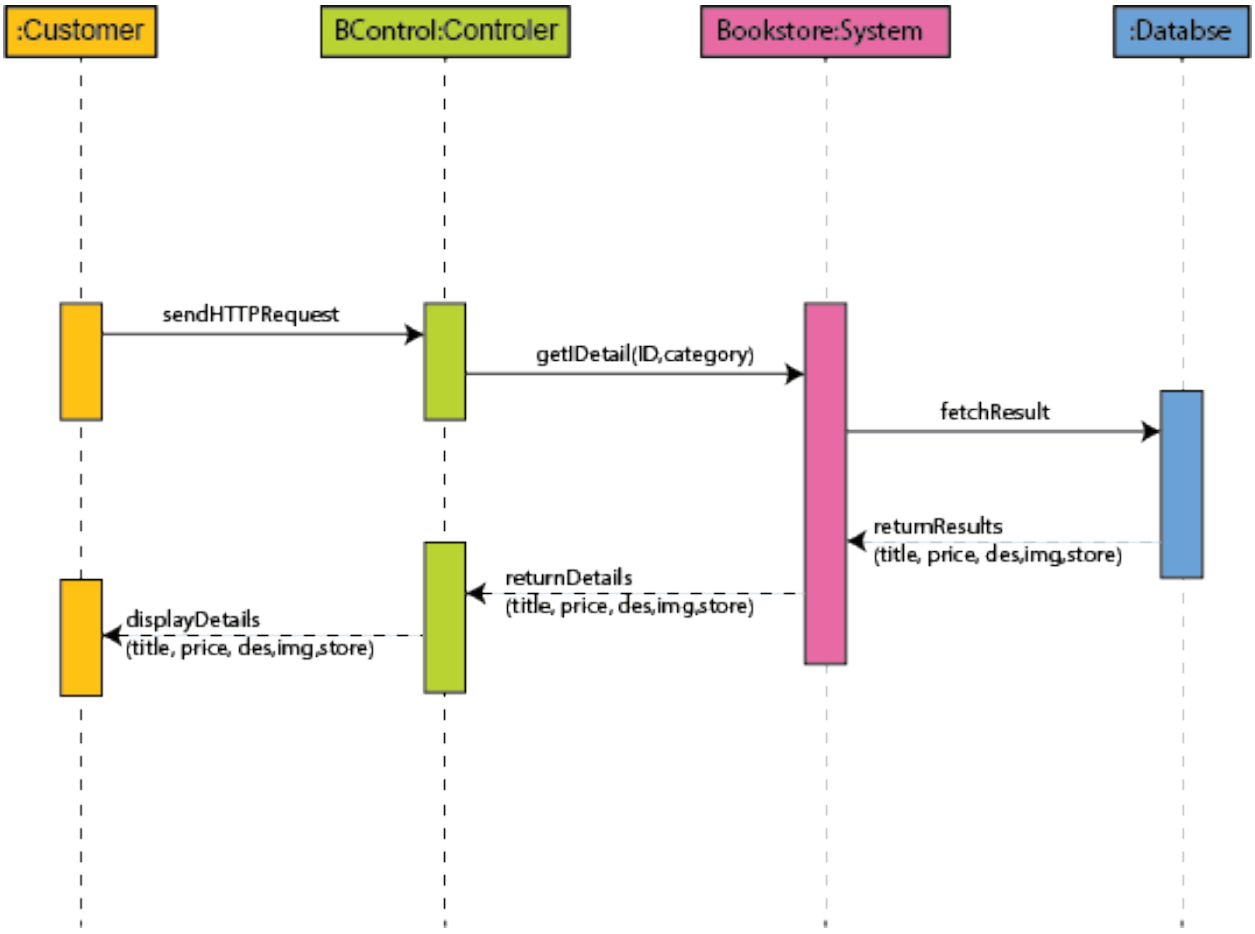
The customer enter the keyword and click on the advance search button to start the search sequence. The controler return the advance page to the GUI. GUI send the HTTP request to the controler after get all the form information from the customer. The controler then send the arguments to the corresoponding business logic. The advance search logic will then fetch the result form the database. After getting the query result returned from the database, the system will return the result back to the controler which then return it to the GUI for display.

SEQUENCE DIAGRAM 03 - NAVIGATE ITEMS



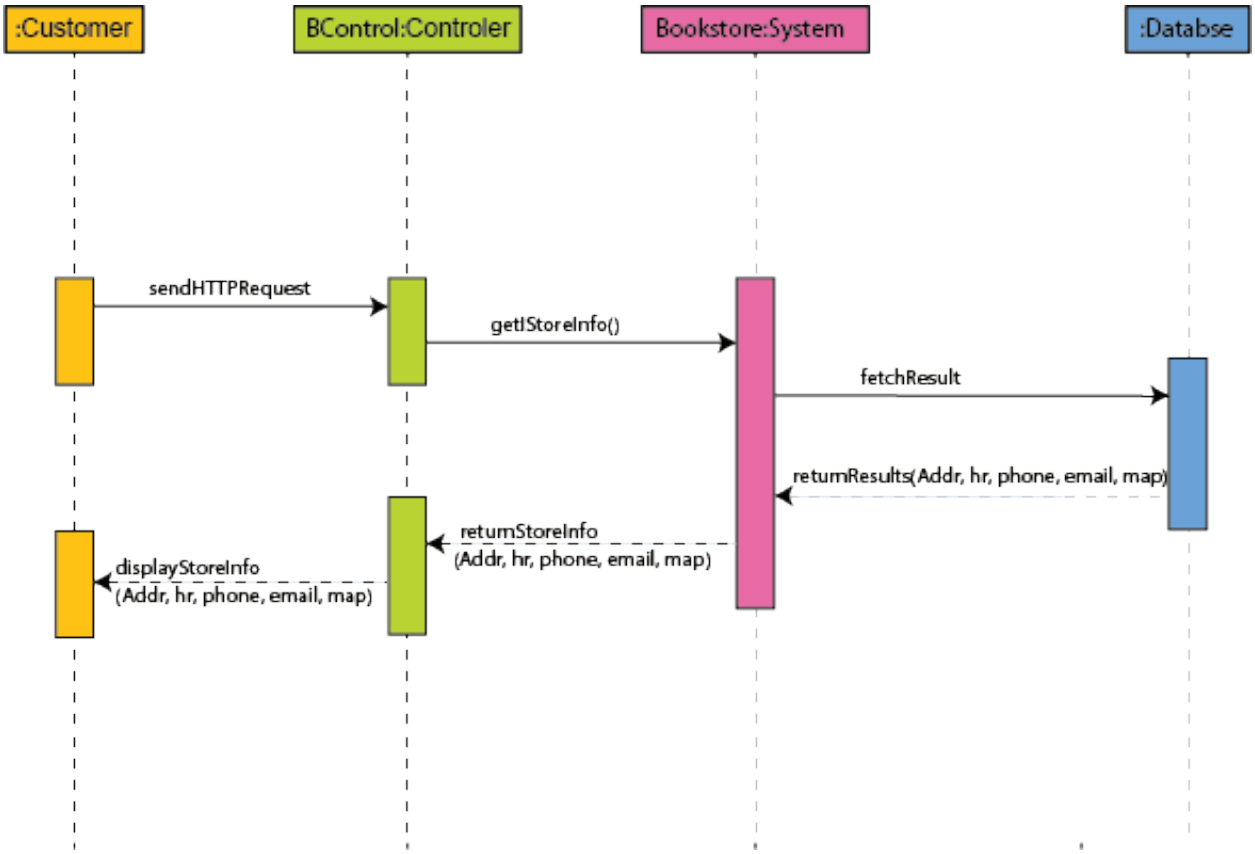
The customer click on the category from the navigation pane which send a HTTP request to the controler. The conetroler then send the request with the category arguments to the system. The system then query the database for the items in that category. After the resutls are returned to the system, the system return the items back to the controler then to the GUI.

SEQUENCE DIAGRAM 04 - GET PRODUCT DETAILS



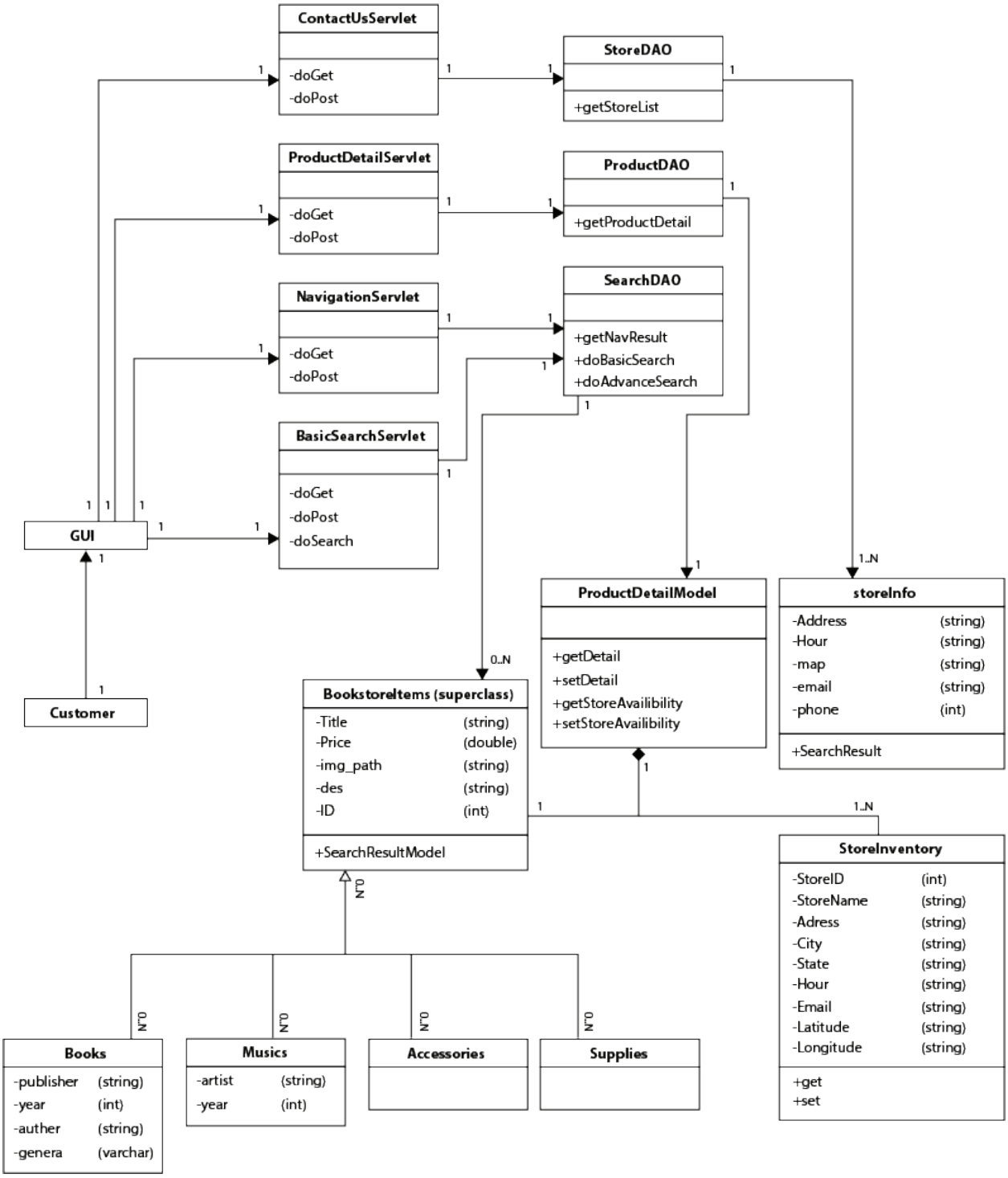
The customer click on individual item on the page sending a HTTP request to the controler with the itemID and category as arguments. Controler send the getDetail method to the system which query the result form the database. The database return the result set to the controler. The controler return the result set to the GUI, and dpeneding on the category GUI will display different attribute items form the result set.

SEQUENCE DIAGRAM 05 - CONTACT US

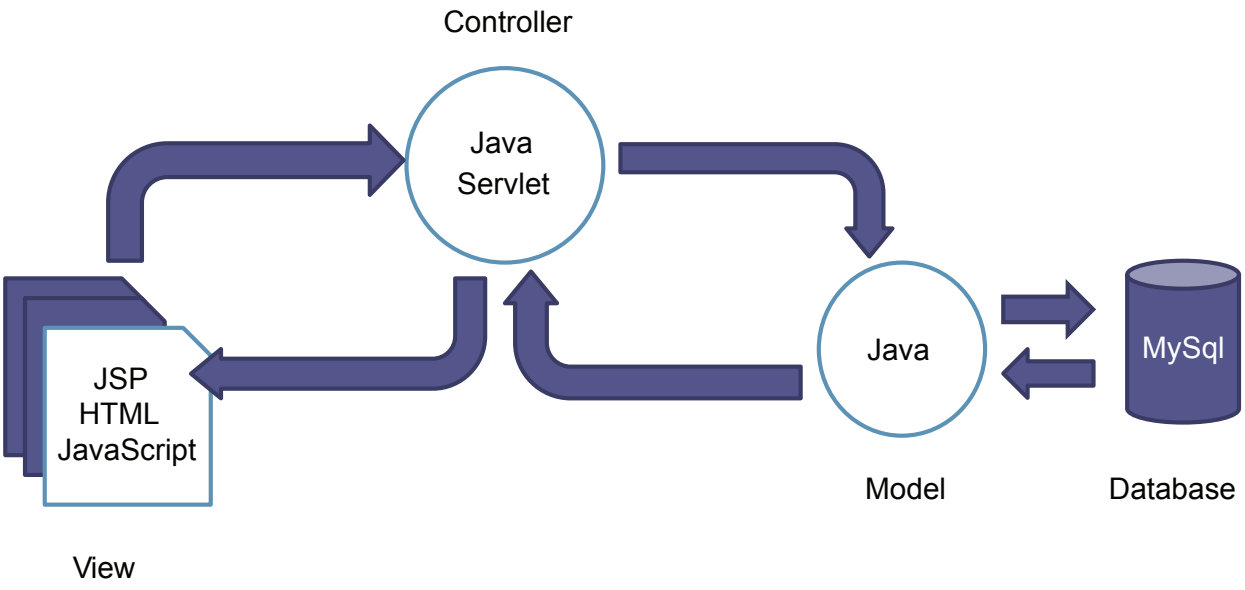


The customer click on the contact us button. It send a HTTP request to the controler. Controler send the getStoreInfo method to the system. System send the query to the database to fetch the result. After the result are returned to the system, the system return it to the controler. Then the controler return it to the GUI for display.

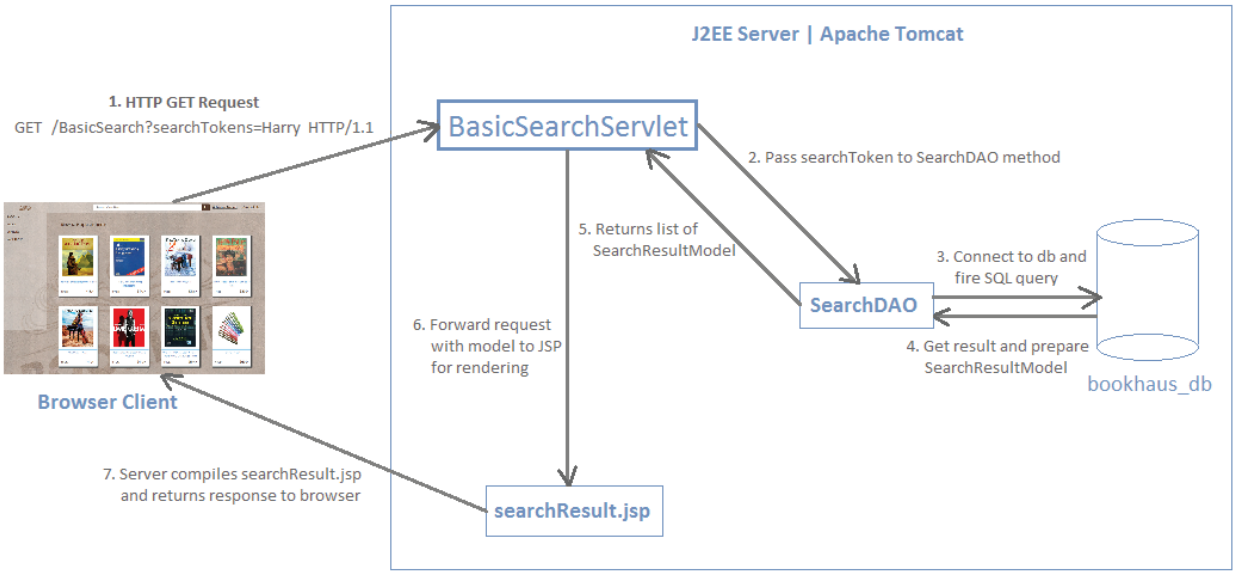
CLASS DIAGRAM



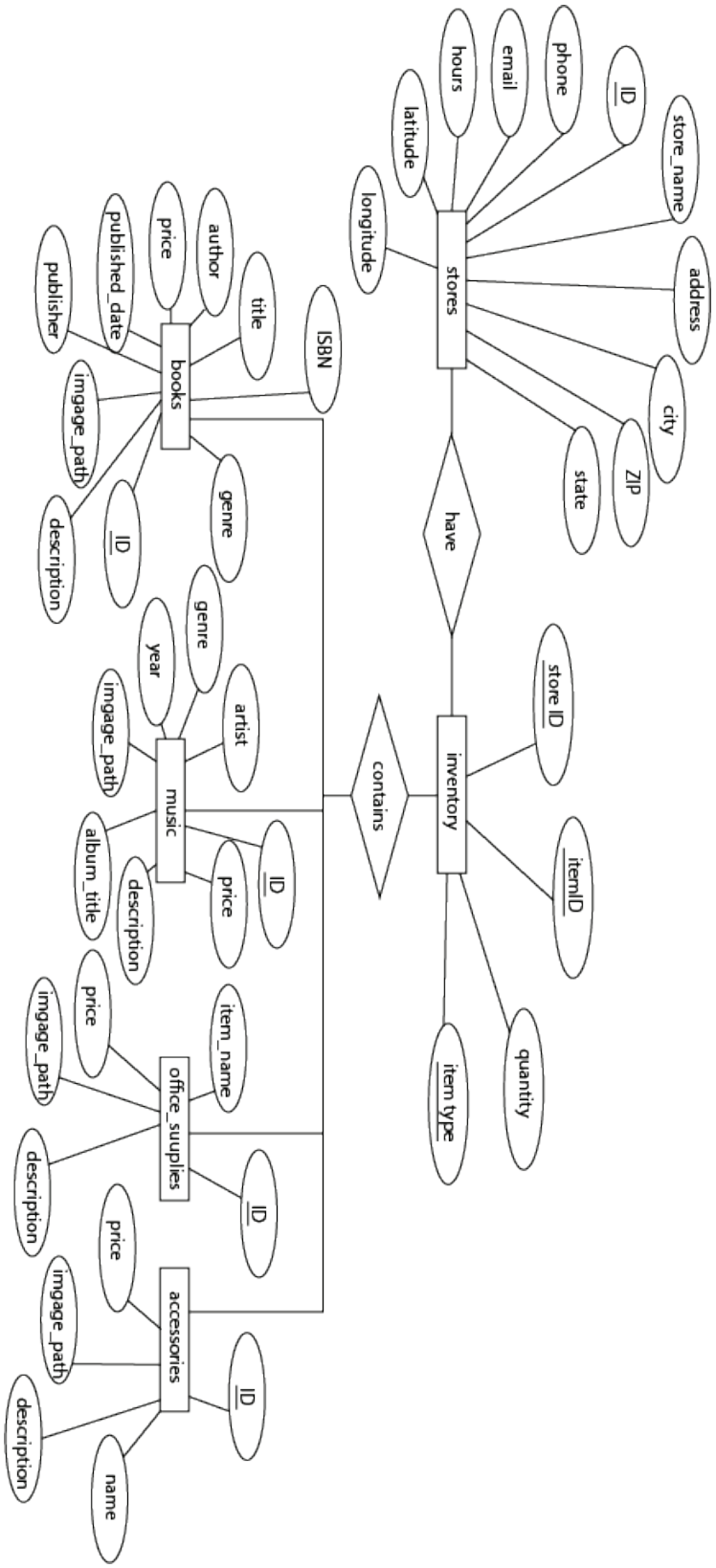
MVC ARCHITECTURE



The MVC architecture solution we are using is to have a java servlet class act as the controler between the JSP and the java model class. The JSP (View) is resposible for the view with HTML and javascript. The click event will send HTTP request to the different controlers depend on the action. Each servlet controler will then pass on the arguments to the java class (Model) which act as the business logic. Then the model classes are the only classes that interact with the database.



ER DIAGRAM


store_name
address
ID
phone
email
hours
latitude
longitude
stores
city
ZIP
state
have
store ID
inventory
itemID
quantity
item type
contains
ISBN
books
title
author
price
published_date
publisher
imgage_path
description
ID
genre
genre
year
artist
imgage_path
music
album_title
description
ID
price
price
imgage_path
description
item_name
office_suuplies
ID
price
imgage_path
description
accessories
ID
name

JUNIT

JUNIT is a unit testing framework for java programming language. 1 test case were developed for each use case. The test cases are the following:

1. Basic Search Test Case - we created a test case for the getCleanToken function to test if the function is blacklisting out the specific words from the search string.

2. Advance Search Test Case - we are testing if the doAdvanceSearch take the arguments (map and queryMap) and return the proper result form the database.

3. Navigate Test Case - we are testing the return result size of the doNavSearch function with the test arguments for category and genre.

4. Product Detail Test Case - we are testing the return result size of the doNavSearch function with the test arguments for id and category.

5. Store Location Test Case - we are testing if the return number of stores is correct from the database.

BUILD SCRIPT

{

Mao, Wesley

Ratnaparkhi, Akshay

Salitra, Jyoti

Kadia, Anald

Kulkarni, Sneha

}