

PRACTICAL 2 – REPORT

FRONT END IMPLEMENTATION AND DESIGN

In this practical we were asked to design and produce a web-based user interface for the database for the online streaming service created in Practical 1.

Tools used

I used MySQL and Java to make a connection to the database, and subsequently GWT to build the front end and connect to the server. I used GWT charts to build charts.

Tested on – Google chrome

Running the project

Depending on IDE used or via an ANT script, GWT has provided the way to run the code in file README.txt. I used IntelliJ idea to store and work on the code, and the ANT script provided to run it.

Design

Overview

A user streaming the site can

- View all audiobooks
- View audiobooks sorted by title, price, popularity, genre, etc.
- View all audiobooks by all authors (sorted by title, popularity)
- Search for audiobooks by name, or name of author or narrator
- View all the information regarding an audiobook (reviews, age rate, publisher ID, ISBN, image, etc.)
- Sign up if not already a customer

If the user is a customer (has an account), they can

- Log in/Log out
- Add one or more audiobooks to their cart (which stays the same after log out too)
- Purchase them together (this removes them from cart)
- View their account details
- Update their address
- View all their purchases, with links to stream them
- Get tailored recommendations based on purchases.

Finally, if the user is an employee, he can

- Do everything a basic user can

- Add a contributor (author/narrator)
- View trend graphs for purchases (customers who bought the most books, books bought the most frequently, etc.)

Code Summary

I referenced the GWT website to set up GWT, build an application and UI. Only the skeleton file structure (com.mycompany.mywebapp), skeleton HTML and CSS are unchanged, but the subsequent classes, HTML file additions, CSS sheets and RPC transactions were written by me.

<http://www.gwtproject.org/doc/latest/tutorial/style.html>

To draw charts, I referenced
<http://gwt-charts.appspot.com/#area>

Database Modifications

In order to make the website user friendly and add more functionality, I modified the DB as follows.

- Audiobook
 - I added actual links to the audiobooks in my database, so that I could hyperlink the audiobook and allow a user to “stream” it.
 - A new column with the audiobook’s actual image for user friendliness of the website.
- Cart
 - I added a table “cart” which stores the ID of the customer and the ISBNs of the books.
- Customer/Employee
 - Added address, email and password fields for updates/authentication.

Interface Design, Implementation & Testing

I used the GWT framework to build the website, which is a lot like Java Swing and uses panels, grids, widgets etc. with HTML and CSS to build a website. There are 3 parts to my code-

Client

The client represents the UI. This stores all the functionality for the interface. Since I used RPC (as shown in the tutorial for GWT), the application is a one-page HTML application, i.e., the page gets updated in the same window without the URL changing for GET/POST/PUT/DELETE requests. Callbacks are made to the server for each button/widget click, which requests the data.

Server

The server uses Java and SQL commands and queries to connect to the database. A connection to the server is established using a properties file (for

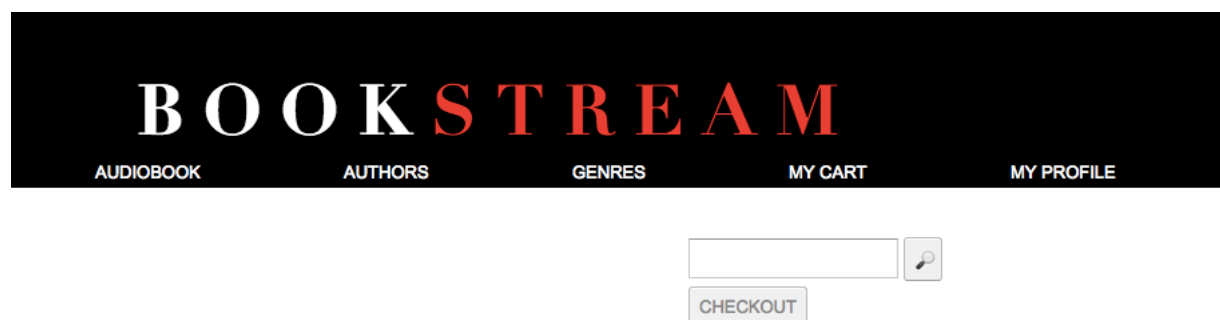
Shared

All the classes representing the datatypes that are passed between server and client are stored in the shared folder. All data collected by requests made by the server is stored as objects of these classes and then passed to the client to be read and de-serialised.

I tested the front-end visually (screenshots provided), and the backend using the ANT script provided by GWT and on IntelliJ.

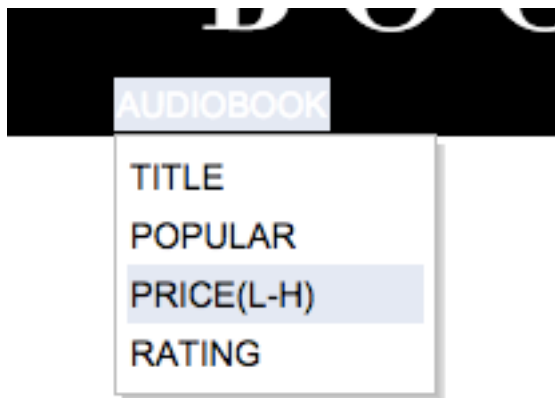
The Interface

The page opens with 4 most popular audiobooks, a menu bar representing the different options, and a search bar.

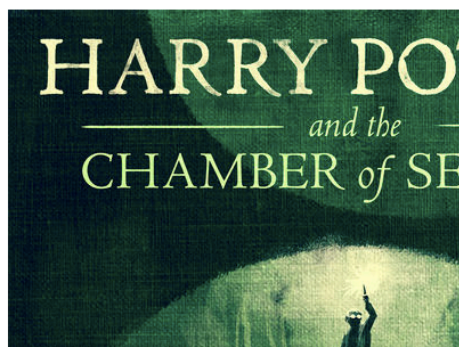


Since this is the introductory page, not all audiobooks are displayed. The 4 (arbitrary) most popular audiobooks (based on purchases) is shown, and upon clicking on the image for these, all information for it is displayed.

The different categories for a user to filter his/her search from are listed in the menu bar, with further filters in the drop down menu. For example, to view all audiobooks, a user can click on “BY TITLE” under “AUDIOBOOK”. To view by popularity, “POPULAR” can be clicked. The same functionality applies to Authors.



Upon clicking any of these options, the user can view all audiobooks in the system in the order selected.
For all books listed, the reviews and associated information for it is displayed.



Harry Potter and the Chamber of Secrets, Book 2

Narrated by
Arthur Conan Doyle
Published by 1202
Genres

Fantasy [Childrens Reads](#)

Written by
[Jakob Enerhaug](#)

ISBN - 1303
PRICE - £33.4

PUBLICATION DATE - 2015-01-08

AGE RATING - Family friendly!

Add to cart

REVIEWS

Rating- 3.5

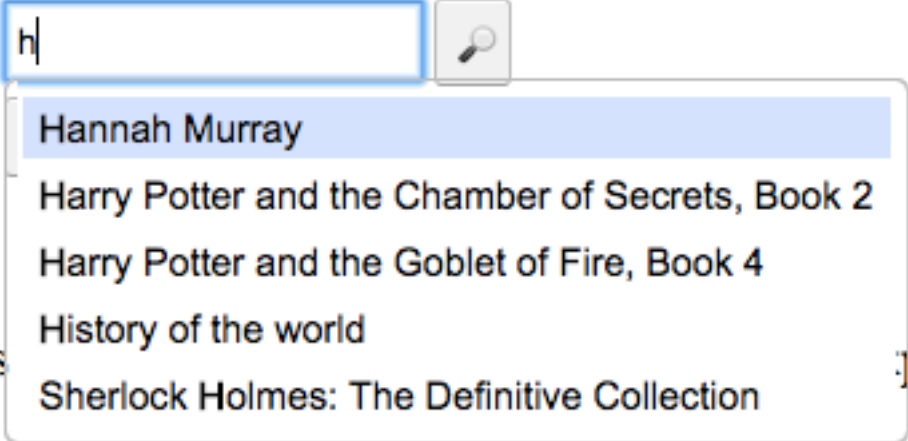
Comment- "Interesting read"

Given By- 1404

As an extension, I queried the database to implement the most commonly used functionality in online markets.

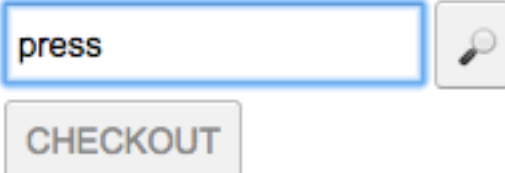
If the user wants to search by a keyword, he can type the keyword in. The best matches are suggested in the drop down panel. In order to achieve this

functionality I queried the database to get the names of all the authors, narrators and audiobooks in each of the tables. This information is then passed on to the client.




A search interface with a text input field containing the letter 'h'. To the right of the input is a magnifying glass icon. A dropdown menu is open below the input, displaying a list of suggestions: 'Hannah Murray' (highlighted in blue), 'Harry Potter and the Chamber of Secrets, Book 2', 'Harry Potter and the Goblet of Fire, Book 4', 'History of the world', and 'Sherlock Holmes: The Definitive Collection'. The words 'books' and 'explore.' are partially visible on the left and right sides of the dropdown menu respectively.

If the user selects one of the options suggested, the book(s) are displayed. If not, the database is queried to get the closest matches and the books are returned to be displayed to the client.



A search interface with a text input field containing the word 'press'. To the right of the input is a magnifying glass icon. Below the input field is a button labeled 'CHECKOUT'.



CHECKOUT

books for purchase and streaming. Browse to explore.

Murder on the Orient Express

Narrated by

Arthur Conan Doyle

Published by 1200

Genres

Murder-Mystery

Thriller

Written by

Jackob Enerhaug

Peter Davison

ISBN - 1300

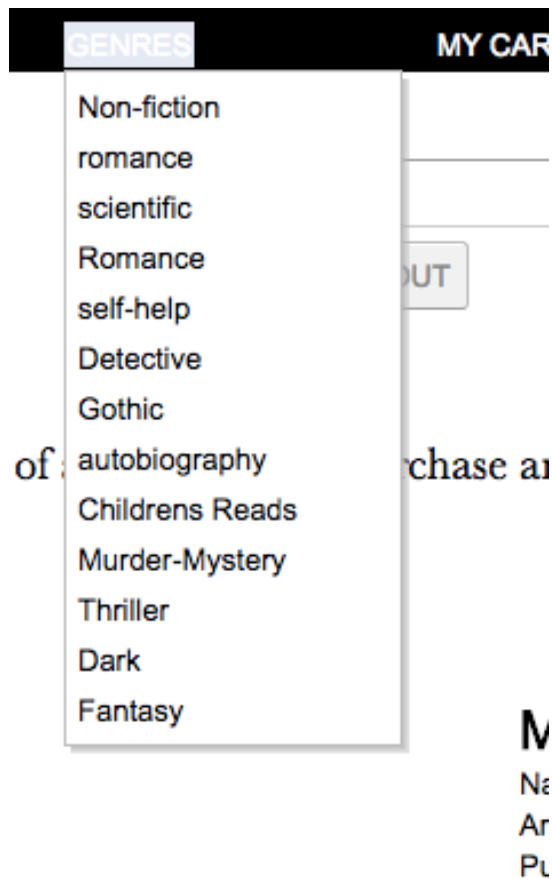
PRICE - £33.45

PUBLICATION DATE - 2017-11-01

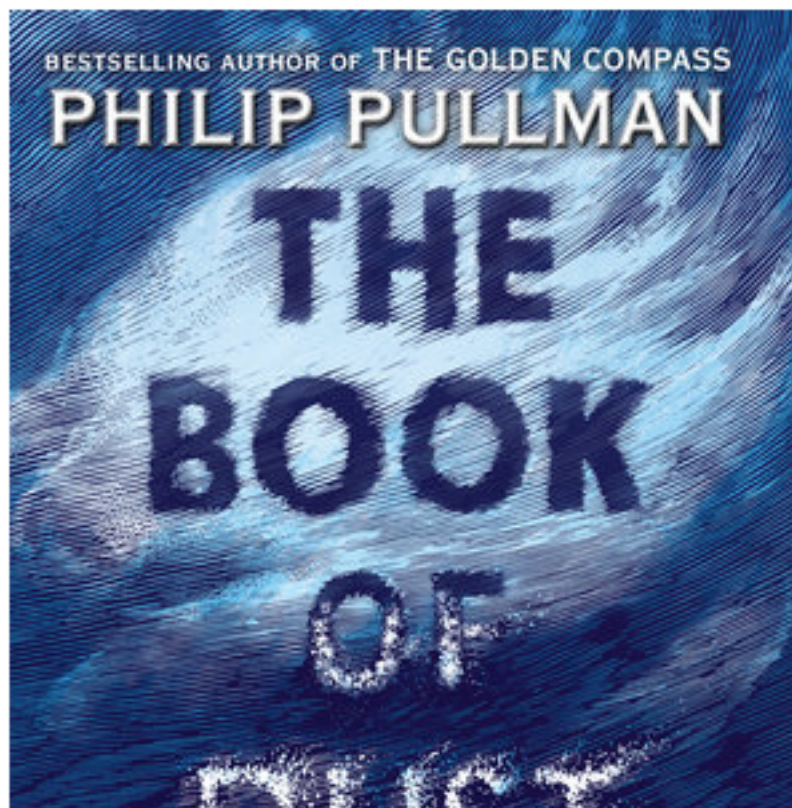
AGE RATING - 10

Another commonly used functionality is sorting- to get books by price and rating.

The books can also be searched by genre. In order to do this, I first query the database to get all the genres of books to display. Then if the user clicks on one, the database is queried again to get all books for which one of the genres match the selected one.

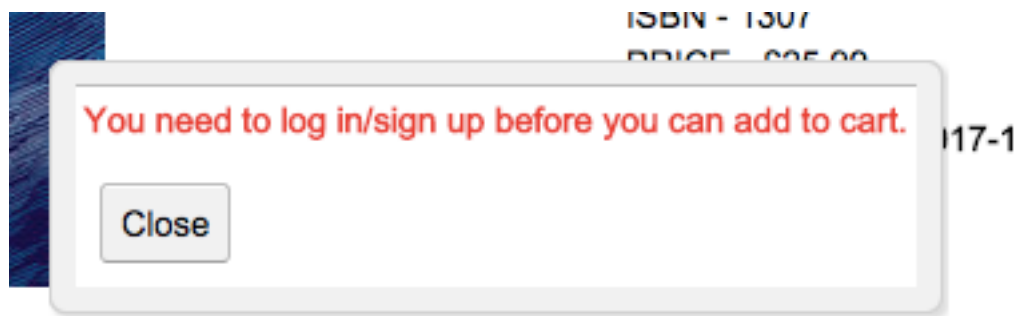


On clicking on these links, the page refreshes to show the audiobooks by whatever order selected by the user. Information is displayed for each book, along with reviews if any. A button can be clicked to add the book to the cart.



Add to cart

As it is with most online purchasing sites, a user has a cart or list of items he intends to buy. To implement this, I added a table to the database named *cart*, which stores the ID of the customer and the ISBNs of the books he intends to buy as foreign keys (M-M relationship), much like the purchase table. The cart table gets updated every time the user does so. He/she can add items to the cart which will remain even if the user has logged out, to reappear when they log in again.

A screenshot of a 'LOG IN' form. The form has a title 'LOG IN' at the top. Below the title, there are two input fields. The first field is labeled 'Enter username:' and contains the text 'ah@mail.com'. The second field is labeled 'Enter password:' and contains three dots '...'. Below the password field, there are two buttons: 'Log In' and 'Cancel'.

The user can, as a customer, log in to add something to the cart. This query is an insertion to the cart table of the database. It is the case with most websites as in real life – purchasing an item is usually not possible unless the user has registered.

The log-in or sign up buttons ask for preliminary details such as email and password. Some are auto-filled for ease of testing.

LOG IN

Enter username:

iajfckL|

Enter password:

...

Log In

Cancel

TRY AGAIN

Sorry, you're not in our records.

Enter email:

mt@gmail.com

Enter password:

...

Enter name:

Marcus telford

Enter address:

london

Sign Up

Close

Try Again

Signing up is again, another insertion to the database. A new cart is created for the customer.

Once the user has logged in/signed up, he can add books to the cart and then checkout.

CHECKOUT

POPULAR TODAY

Welcome to BookStream! We offer a vast range of audiobooks for purchase and stream



Murder on the Orient Express

Narrated by

Arthur Conan Doyle

Published by 1

Genres

Murder-Mystery

Written by

Agatha Christie

ISBN - 1300

PRICE - £33.49

PUBLICATION DATE

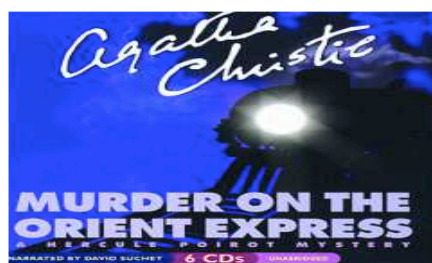
AGE RATING - 10

Add to cart

CHECKOUT

POPULAR TODAY

Welcome to BookStream! We offer a vast range of audiobooks for purchase and stream



Murder on the Orient Express

Narrated by

Arthur Conan Doyle

Published by 1

Genres

Murder-Mystery

Written by

Agatha Christie

ISBN - 1300

PRICE - £33.49

PUBLICATION DATE

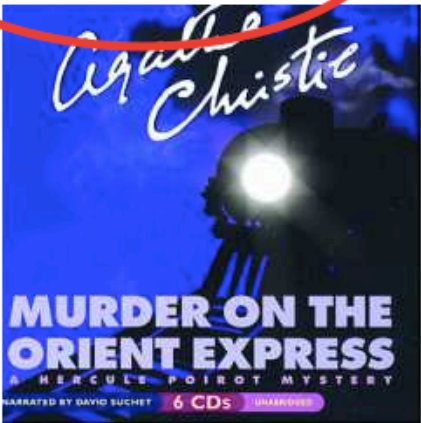
AGE RATING

Add to cart

On purchasing, the customer's homepage ("MY ACCOUNT") is displayed, with all his purchases and the links to stream the audiobooks (I used Anchors to do this, which currently links to the amazon.co.uk page of the audiobook).

Your Purchases

Purchase date : 1998-09-09
[Murder on the Orient Express](#)



Murder on the

Narrated by
Arthur Conan Doyle
Published by 1200
Genres
Murder-Mystery
Written by
Jackob Enerhaug
ISBN - 1300
PRICE - £33.45
PUBLICATION DATE - 2017-1
AGE RATING - 10

Add to cart

The customer can also update his address on the page. This is an update query carried out on the database. The page is automatically updated with the new address.

Welcome, Vivian ㅈ Cha

Your registered Home Address :
Houng street 33, seoul

Update address

CHECKOUT

AR TODAY

me to BookStream! We offer a vast range of books and audiobooks. Please use the address below to place your order. If you wish to change your address, please click on the "Update address" button.

Welcome, Vivian ㅈ Cha

Your registered Home Address :
Houng street 33, seoul

Update address

Your Purchases

Purchase date : 2009-10-08
[The Ruby in the Smoke](#)

Enter address:
JungDong 27 Seoul

Update

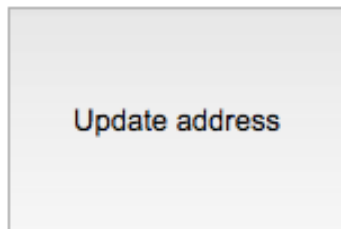
Cancel



Welcome, Vivian ㅈ Cha

Your registered Home Address :

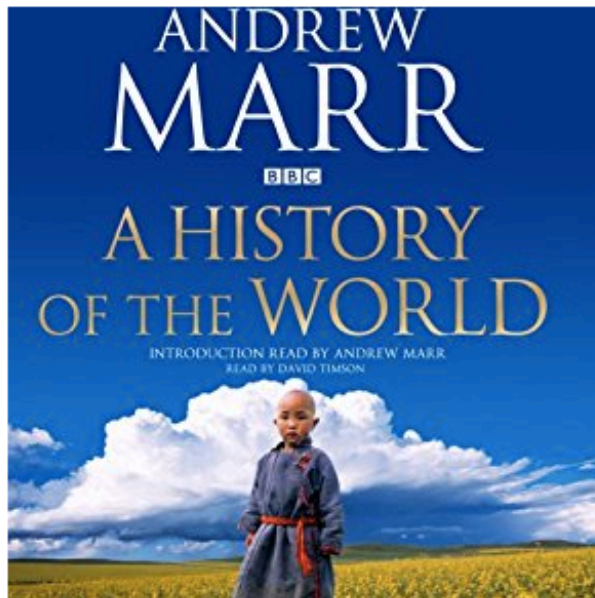
JungDong 27 Seoul



Your Purchases

Another interesting feature of purchasing websites is *suggestions*. I looked into affinity analysis online and found some interesting ways to query the database to find books “tailored” for the user’s taste. The most apt one was using the purchase history of the user- for every purchase added to the database, the genres of the books are compared to the genres of the available books in the database, and then the panel is updated.

Recommended for you



Hist

Narrate

Arthur I

Publish

Genres

Non-fic

Written

Peter C

ISBN -

PRICE

PUBLICATI

AGE RATIN

Add to cart

REVIEWS

No reviews currently.

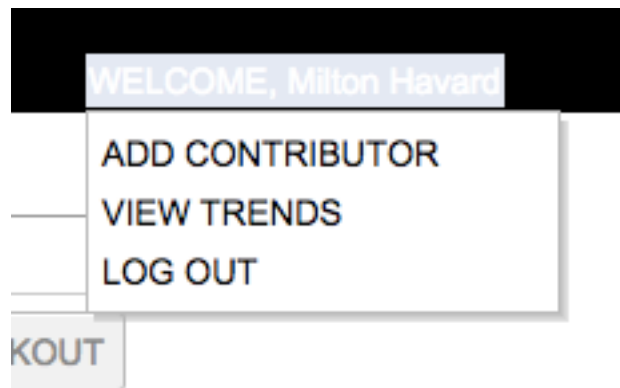


Orig

Narrate

Sylvia I

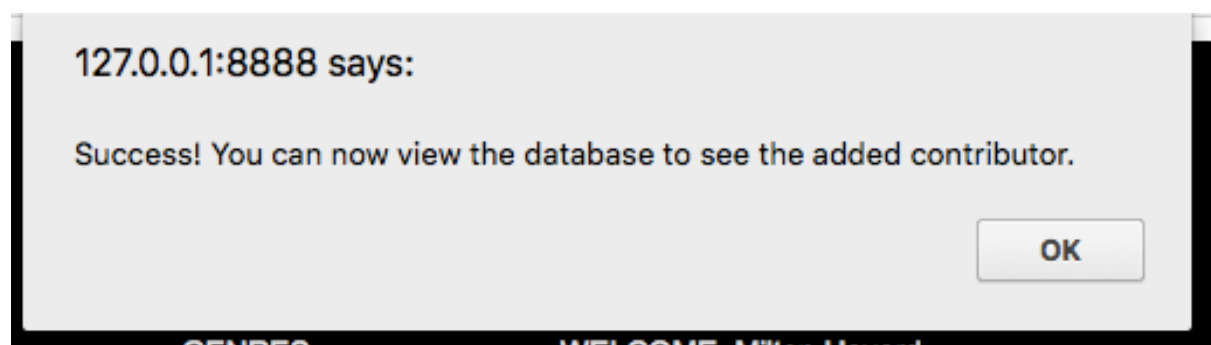
An employee cannot purchase a book through his employee account, but can view all books in the system. In addition to this, he can add to the database. I implemented adding a “contributor” (author/narrator) to the database. If successful, a message is displayed. I tested the insertion by looking at the database table.



Enter name of contributor:

Enter bio/description::

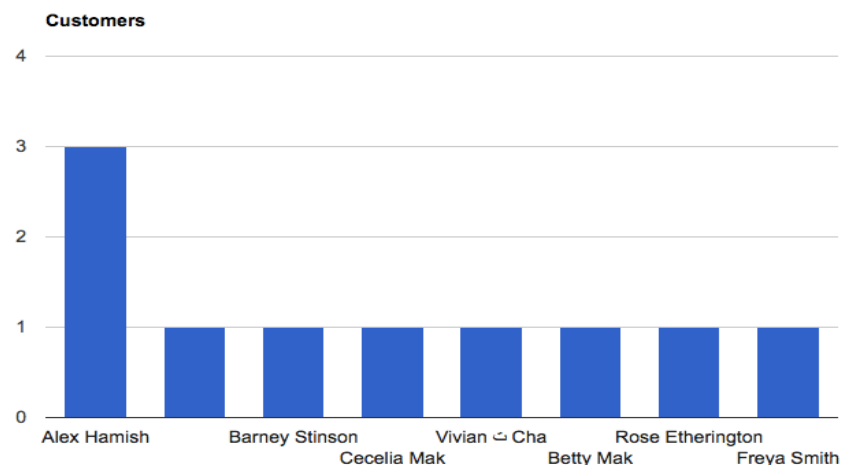
Enter DOB (format: YYYY-MM-DD)



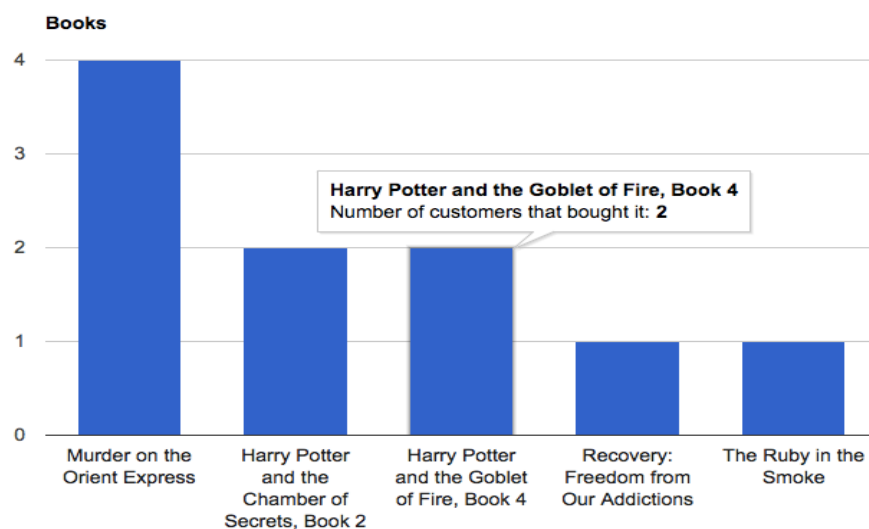
A very interesting prospect for a company is trends in sales and customers. I implemented a feature which queries the database to graph the following-

- Get all the customers who have bought the most items (During my research I found that companies like ASOS do this to provide “points” to customers).
- Get all the popular audiobooks (this can be used to see which items were the most purchased, so as to ask for more stock, increase price, etc.)

CUSTOMERS WHO BUY THE MOST



POPULAR AUDIOBOOKS



A user can logout- this displays the main page again.

Database Operations

The operations a user can do on the front end are described above. The technicalities of it lie in the queries. The database can deal with selection, insertions, deletions and updates.

Selecting of items from database-

To display items on the front end, the database was accessed using SQL commands and the results returned to the client (RPC).

For example, if the user clicks on the following menu items, the following queries are carried out.

GENRE -> FICTION, the database makes a query to the database which filters the books by genre.

```
try {  
    statement = connection.prepareStatement("SELECT * FROM audiobook where LOWER(audio_genre) like LOWER(?)");  
    statement.setString(1, "%" + genre + "%");  
}
```

AUTHOR-> POPULAR

This one gets the popular authors in terms of money spent on the books they wrote. So, it's effectively a collaboration of the author, contributes, purchase and audiobook tables:

The prices for the books were recorded from the audiobook table,

The overall money spent on each book was calculated using the number of purchases per book from the purchase table

This was filtered for each contributor if he played the role of an author for each audiobook.

This was ordered by the money spent.

The query I used:

```
statement = connection.prepareStatement("select cont_ID, GROUP_CONCAT(audio_ISBN SEPARATOR ',') " +
    "from contribute where audio_ISBN in " +
    "(select t2.audio_ISBN from " +
    "(select A.audio_ISBN, count(A.audio_ISBN) as num, CAST((count(A.audio_ISBN)*A.audio_price) " +
    "AS decimal(38,2)) as total from purchase P inner join audiobook A on A.audio_ISBN = P.audio_ISBN " +
    "group by A.audio_ISBN order by total DESC) t2) and cont_type = ? group by cont_ID");
```

RECOMMENDATIONS FOR YOU

The recommendations for each customer based on purchases used queries already made to the database, to get all the purchased books, and then filter by attained genres.

Most of the operations to the database are “SELECT FROM...” requests. All the information gathered from the database is then displayed to the user, or used for affinity analysis, trends, etc.

However, the notable insertions, deletions and updates I implemented are as follows-

- Update

The user can update their address. The server queries the database to “update customer (the table)” where the ID of the customer logged in is a match. The updated object is now sent back to the user so that the front end can get refreshed with the new address.

customer_email	customer_phones	customer_password	customer_address
ah@mail.com	9830068003, 31229099	abc	7 Largo road, St Andrews
bs@hmail.com	9830068003, 31229099	abc	61 West Avenue, CA
cm99@yahk.com	8744467443	abc	2 green park, california
vvc@gmail.com	+44639994884	abc	Houng street 33, seoul
bmak@gmail.com	+91888304844, 02912510505	abc	Bridge street nepal
retherington@gmail.com	07584672472	abc	Manchester avenue. NYC

customer_email	customer_phones	customer_password	customer_address
ah@mail.com	9830068003, 31229099	abc	7 Largo road, St An
bs@hmail.com	9830068003, 31229099	abc	61 West Avenue, CA
cm99@yahk.com	8744467443	abc	2 green park, calif
vvc@gmail.com	+44639994884	abc	JungDong 27 Seoul
bmak@gmail.com	+91888304844, 02912510505	abc	Bridge street nepal

- Insertion

- o Adding a customer

A customer can sign up and this adds him to the database, as well as logs him in.

```
[MariaDB [sks8_cs3101_db]> select * from customer;
```

customer_ID	customer_DOB	customer_name	customer_email
1400	1993-02-04	Alex Hamish	ah@mail.com
1401	1994-03-17	Barney Stinson	bs@hmail.com
1402	1989-05-13	Cecelia Mak	cm99@yahk.com
1403	2003-09-03	Vivian Cha	vvc@gmail.com
1404	1996-11-11	Betty Mak	bmak@gmail.cc
1405	2010-07-10	Rose Etherington	retherington@
1406	2005-12-13	Freya Smith	fs@yahoo.com
1407	1966-11-16	Tom Oliver	bjcootes@gmai

8 rows in set (0.00 sec)

```
[MariaDB [sks8_cs3101_db]> select * from customer;
```

customer_ID	customer_DOB	customer_name	customer_email
1400	1993-02-04	Alex Hamish	ah@mail.com
1401	1994-03-17	Barney Stinson	bs@hmail.com
1402	1989-05-13	Cecelia Mak	cm99@yahk.com
1403	2003-09-03	Vivian Cha	vvc@gmail.com
1404	1996-11-11	Betty Mak	bmak@gmail.cc
1405	2010-07-10	Rose Etherington	retherington@
1406	2005-12-13	Freya Smith	fs@yahoo.com
1407	1966-11-16	Tom Oliver	bjcootes@gmai
1408	NULL	Marcus telford	mt@gmail.com

9 rows in set (0.00 sec)

- o Add To cart

Adding an item to the cart adds it to the table cart, and this change is represented in the menu bar.

```
MariaDB [sks8_cs3101_db]> select * from cart;
```

+-----+-----+	
customer_ID audio_ISBN	
+-----+-----+	
1401	1305
1404	1303
1404	1300
1400	1302
1400	1303
1400	1310
1400	1301
1403	1310
+-----+-----+	

```
8 rows in set (0.00 sec)
```

```
MariaDB [sks8_cs3101_db]> select * from cart;
```

+-----+-----+	
customer_ID audio_ISBN	
+-----+-----+	
1401	1305
1404	1303
1404	1300
1400	1302
1400	1303
1400	1310
1400	1301
1403	1310
1408	1300
+-----+-----+	

```
9 rows in set (0.00 sec)
```

- Adding a contributor
Adds it to the contributor table.

```
[MariaDB [sks8_cs3101_db]> select * from contribu
```

+-----+-----+	
cont_ID	cont_bio
+-----+-----+	
1	Famous for detective fiction stories
2	Norwegian public speaker famous for l
3	Hannah Murray is an English actress l
4	English actor who became famous in tl
5	British novelist and screenwriter who
6	Famous for detective fiction stories
7	Murder-Mystery writer who is well kno
8	Politician turned writer who writes ,
+-----+-----+	

8 rows in set (0.00 sec)

```
MariaDB [sks8_cs3101_db]> select * from contributor;
```

+-----+-----+	
cont_ID	cont_bio
+-----+-----+	
1	Famous for detective fiction stories of Sh
2	Norwegian public speaker famous for his de
3	Hannah Murray is an English actress best k
4	English actor who became famous in the tel
5	British novelist and screenwriter who wrot
6	Famous for detective fiction stories of Sh
7	Murder-Mystery writer who is well known in
8	Politician turned writer who writes courtr
9	New to this world.
+-----+-----+	

9 rows in set (0.00 sec)

- Purchasing an audiobook

A purchase made removes all items from the cart, and subsequently adds them to the purchase table.

```
MariaDB [sks8_cs3101_db]> select * from purchase;
```

customer_ID	audio_ISBN	purchase_date
1400	1300	1998-09-09
1400	1301	1998-09-09
1401	1300	2008-09-09
1402	1300	1990-09-09
1405	1310	2017-08-08
1406	1310	2017-10-08
1404	1303	2016-10-08
1403	1309	2009-10-08
1400	1303	2017-11-30

9 rows in set (0.00 sec)

```
[MariaDB [sks8_cs3101_db]> select * from purchase;
```

customer_ID	audio_ISBN	purchase_date
1400	1300	1998-09-09
1400	1301	1998-09-09
1401	1300	2008-09-09
1402	1300	1990-09-09
1405	1310	2017-08-08
1406	1310	2017-10-08
1404	1303	2016-10-08
1403	1309	2009-10-08
1400	1303	2017-11-30
1408	1300	2017-03-03

10 rows in set (0.00 sec)

I used prepared statements for most of the updates/insertions, to prevent SQL injection.

Conclusion

The system successfully connects the database to a front end, and allows users to log in, view audiobooks sorted by various orders, authors, and genres. A user can add to cart, sign up, view history of purchases and stream links. An employee can also add a contributor and view trend graphs.

I did not have enough time to deploy it, so given more time, I would've tried to succumb the limitations and deploy the site on the school host servers.

Additionally, I would play with the system and power of GWT a lot more- I would have added an administrator to grant access rights, and enabled an employee to add audiobooks, delete authors, publishers etc. I would have also worked more on affinity analysis as the process of "tailored recommendations" interested me a lot. In addition, I used RPC- so the website remains on one page (it doesn't change in the URL bar)- I would have tried working on making it more RESTful.