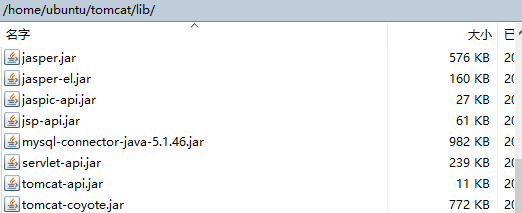
Name: Jiliang Ni

Student ID: 81224742

**Task 1**

* How did you use connection pooling?

First, I copy the jdbc driver’s jar into /home/ubuntu/tomcat/lib



Then I changed some files to achieve connection pooling.

* File name, line numbers as in Github

1. /project2/WebContent/META-INF/context.xml

All lines in this file. I add 2 resources. Jdbc/TestDB is for all read queries, while jdbc/TestDB2 is only for write queries, so both master and slave will only execute write queries on master’s mysql database.

2. /project2/WebContent/WEB-INF/web.xml

From line 25 to line 39.

3. /project2/src/\_dashboard.java

From line 109 to line 130.

From line 221 to line 242.

From line 425 to line 446.

4. /project2/src/CheckOut.java

From line 82 to line 99.

5. /project2/src/Home.java

From line 157 to line 178.

6. /project2/src/Login.java

From line 104 to line 125.

7. /project2/src/MovieList.java

From line 102 to line 123.

8. /project2/src/MyCart.java

From line 76 to line 97.

9. /project2/src/SingleMovie.java

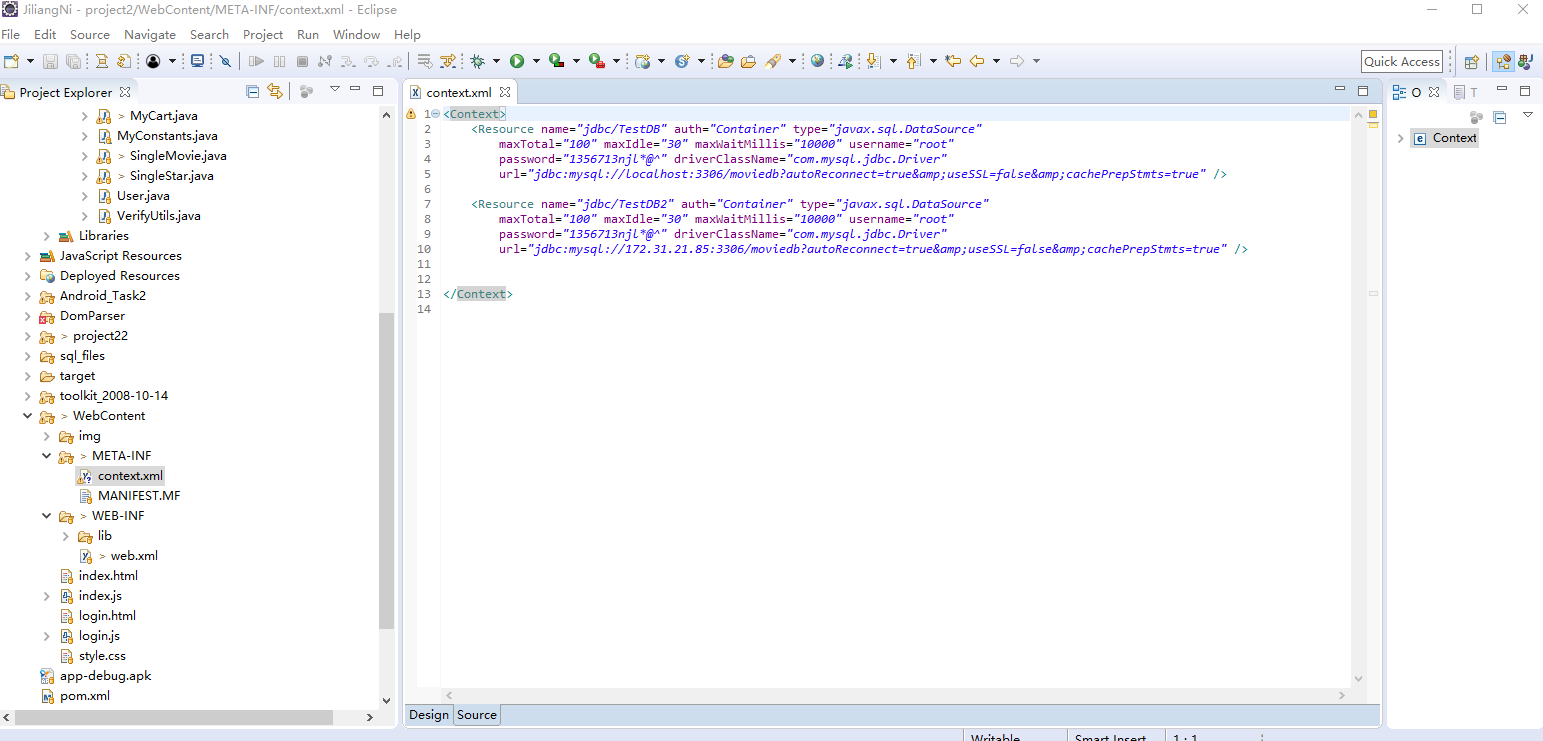
From 51 to line 72.

10. /project2/src/SingleStar.java

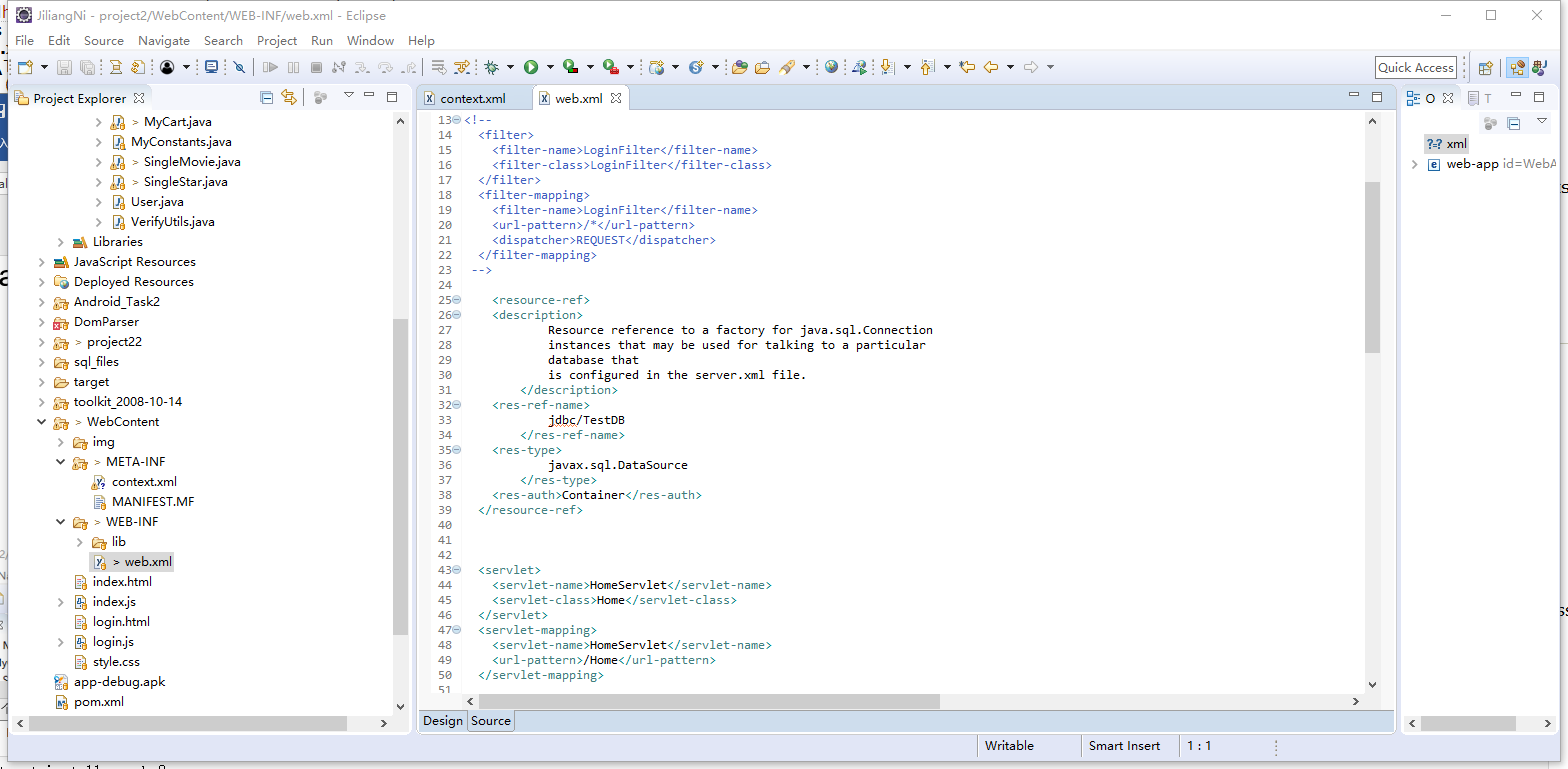
From 51 to line 72.

* Snapshots

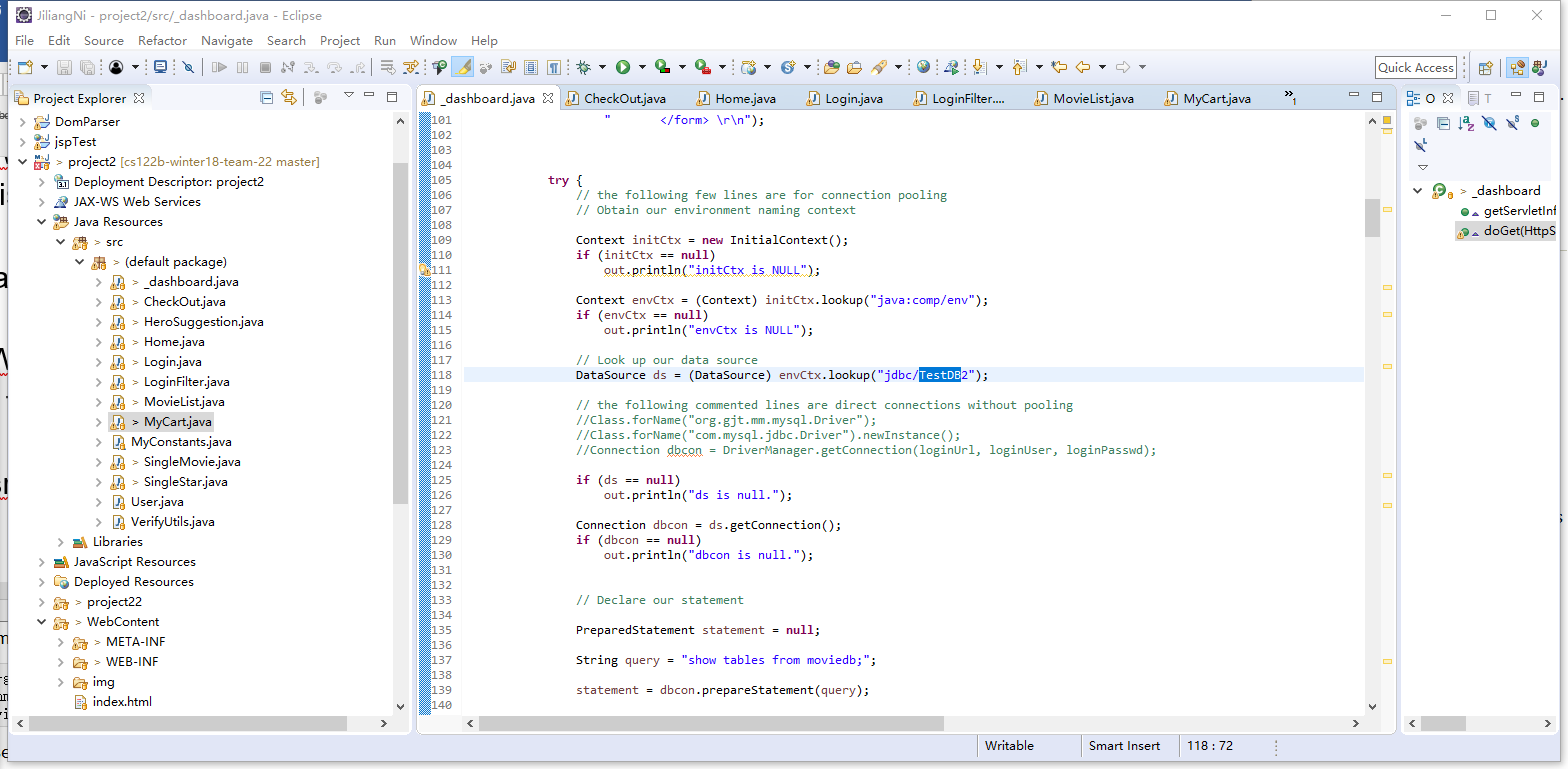
1.

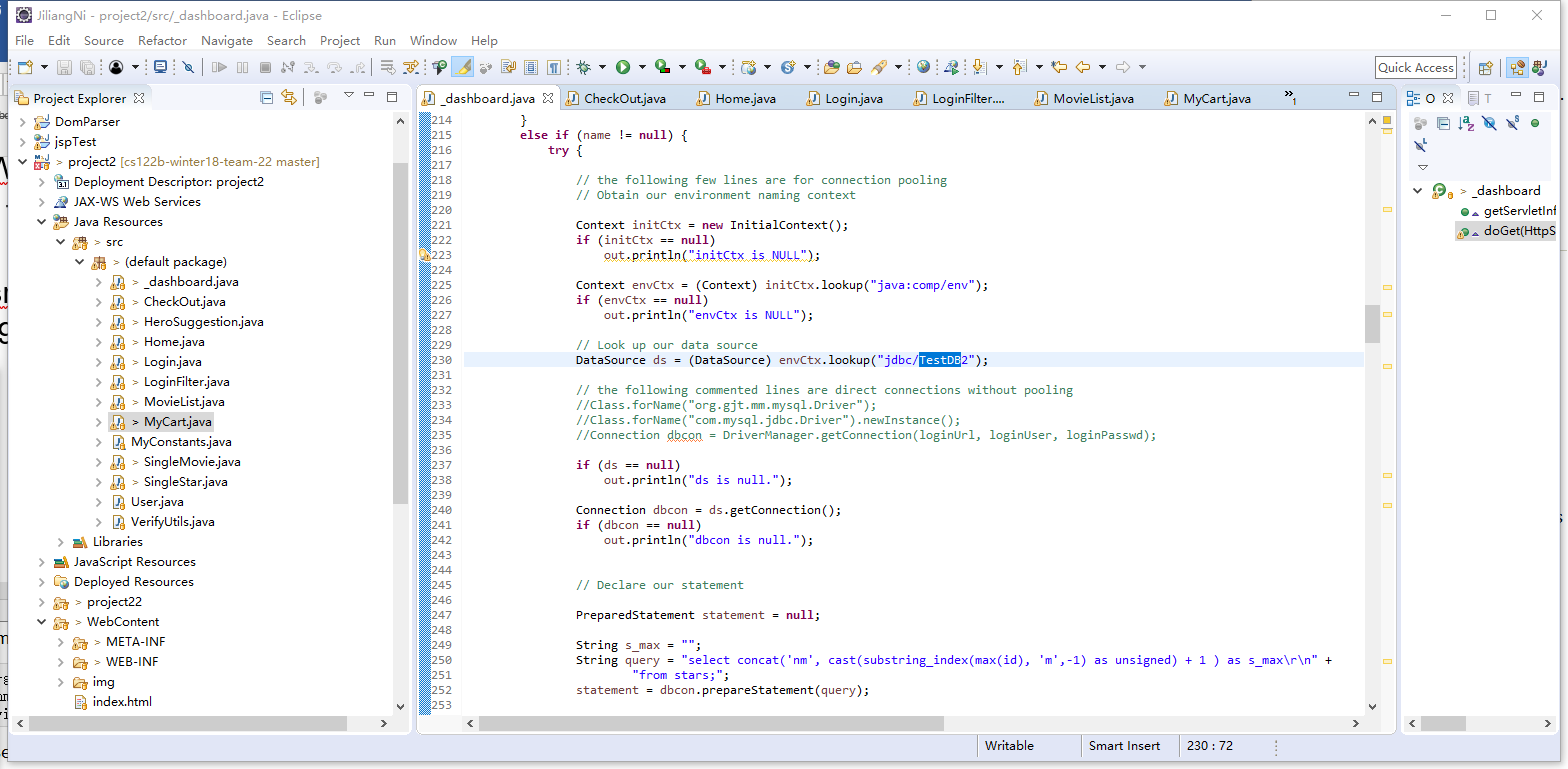


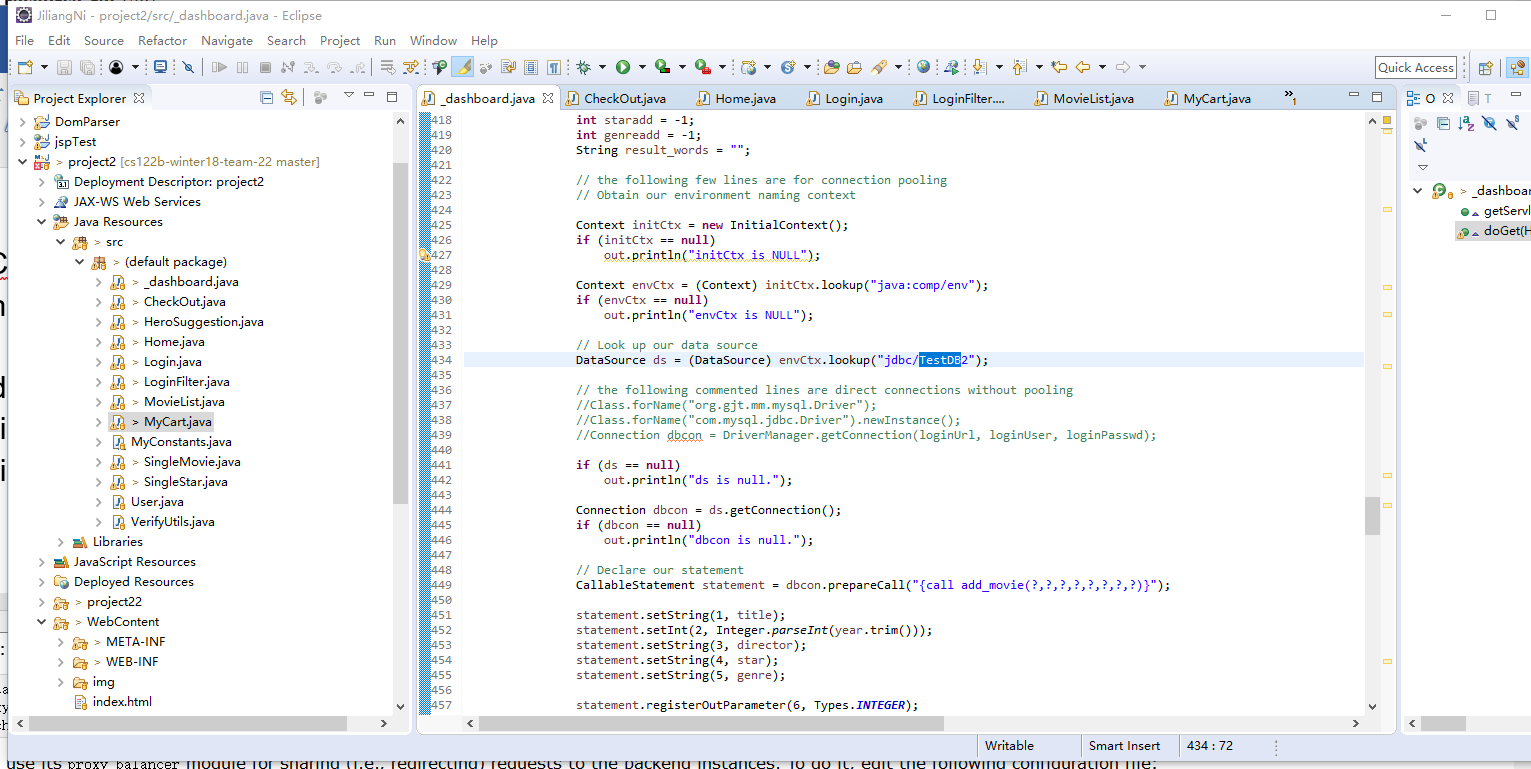
2.



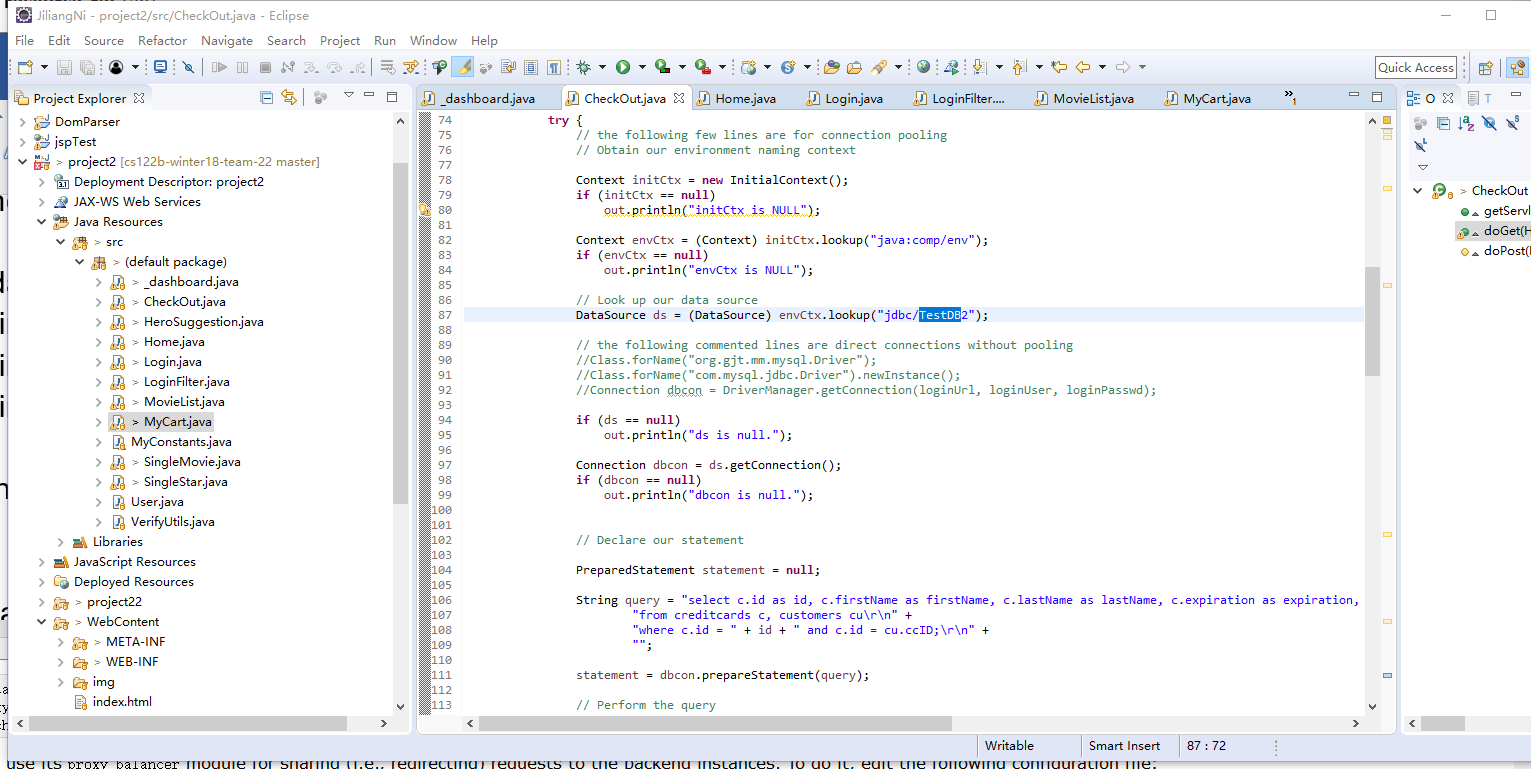
3.



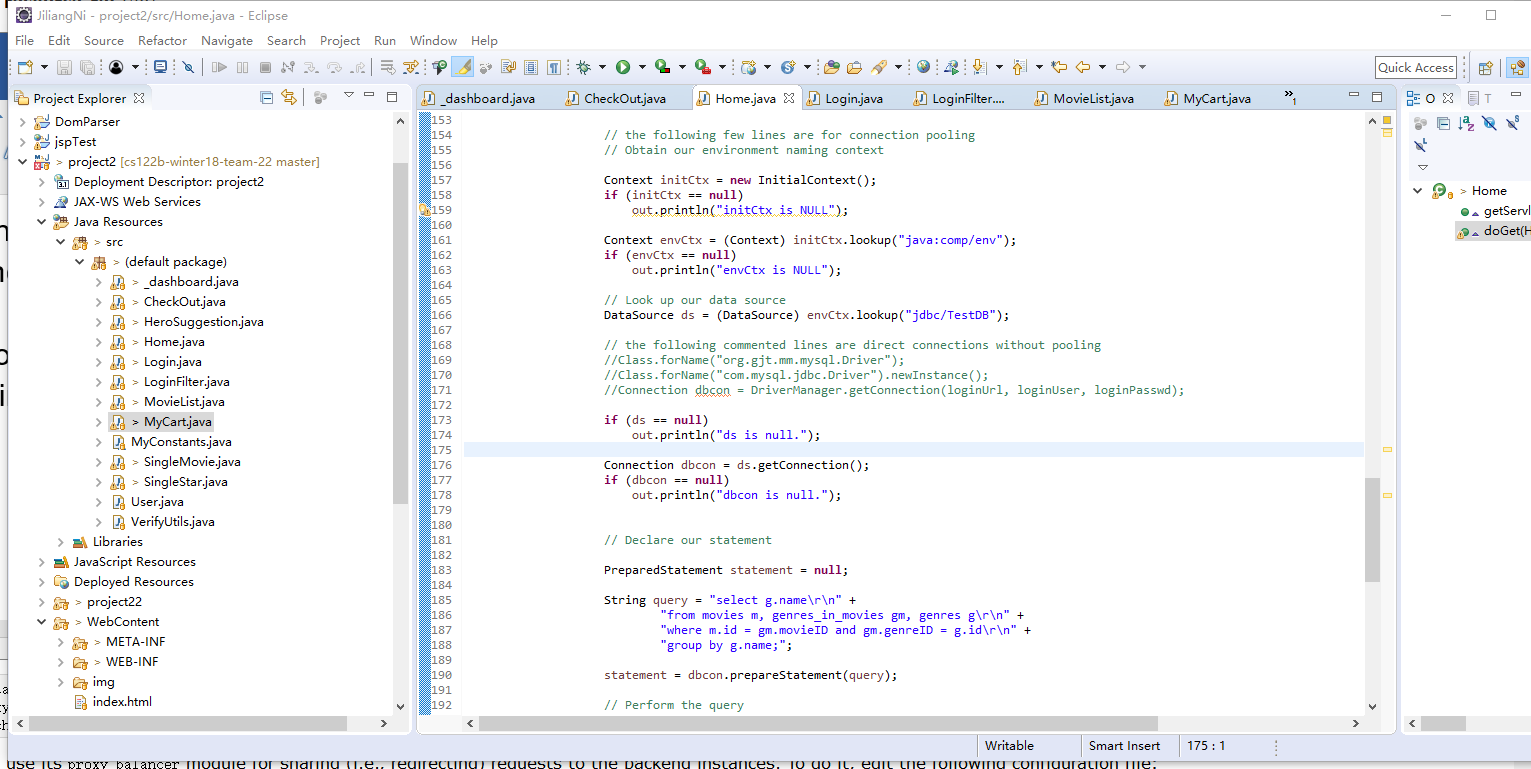




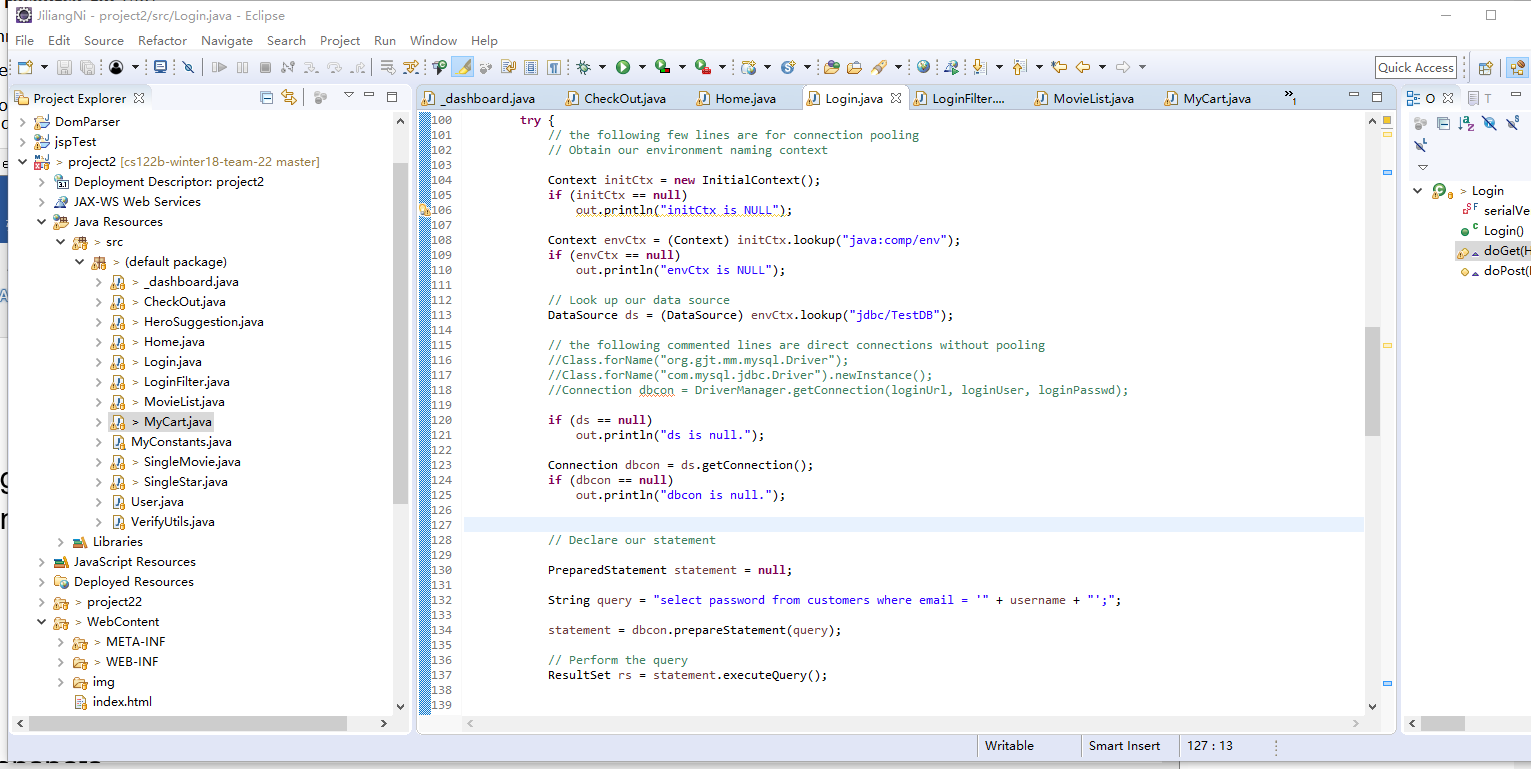
4.



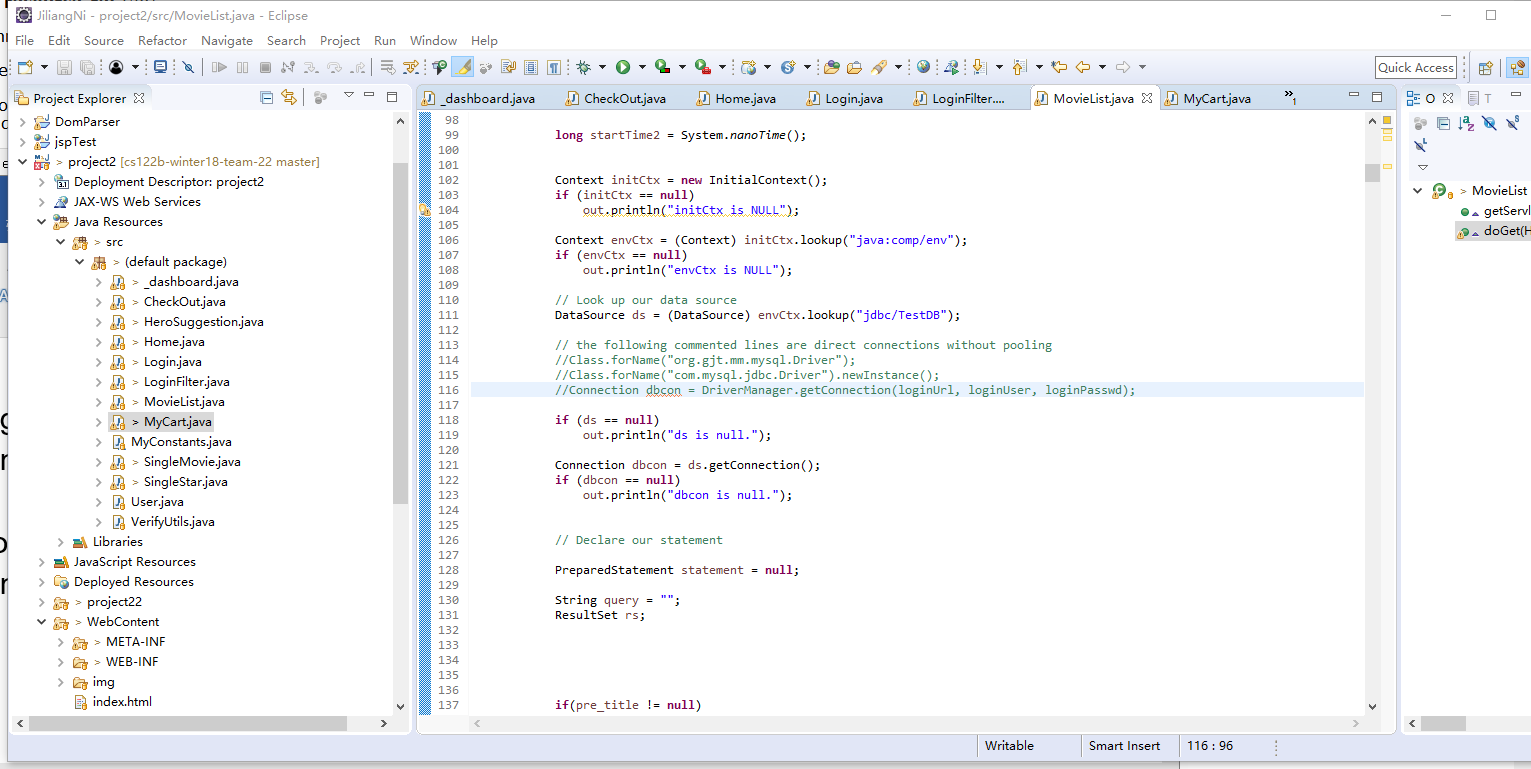
5.



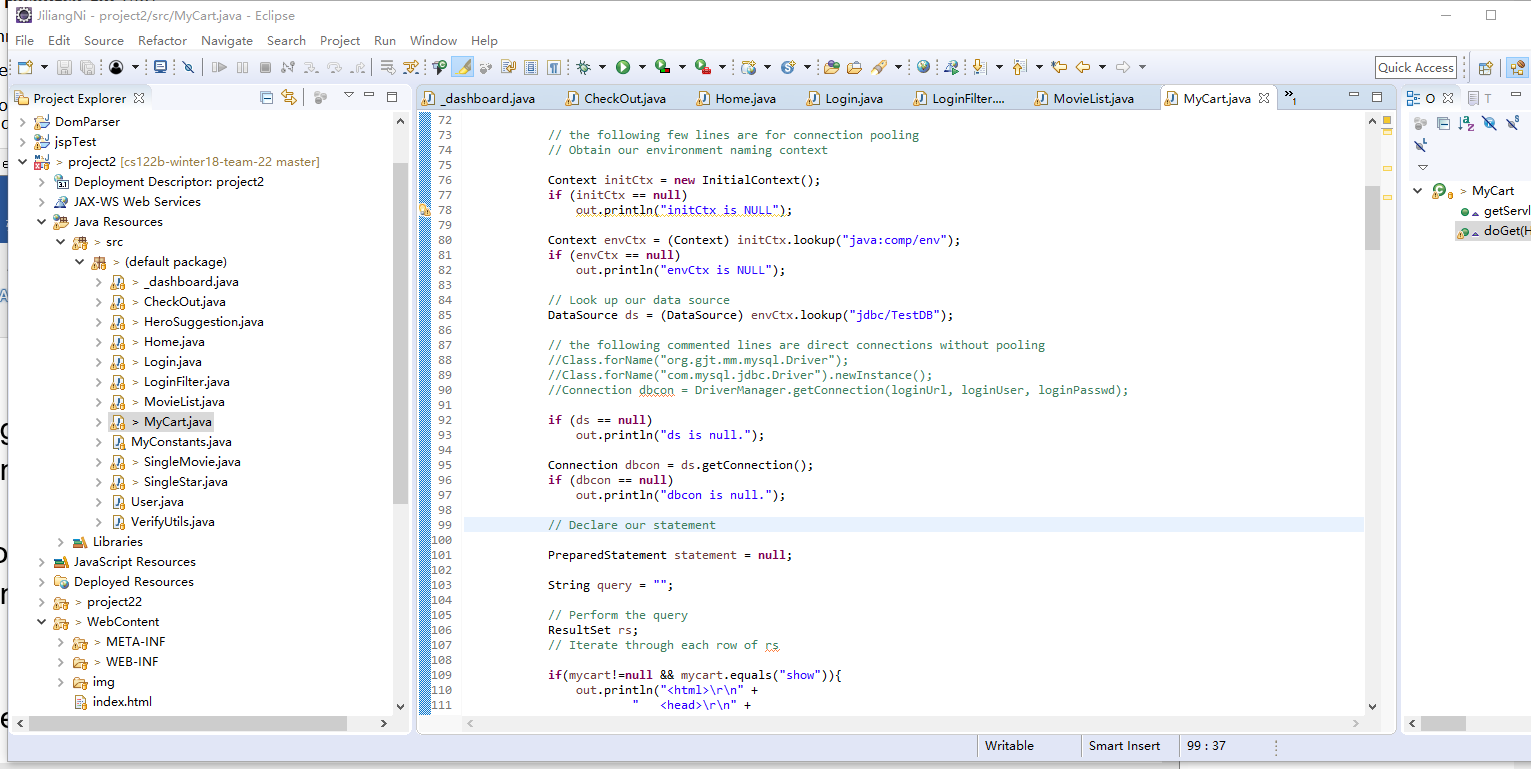
6.



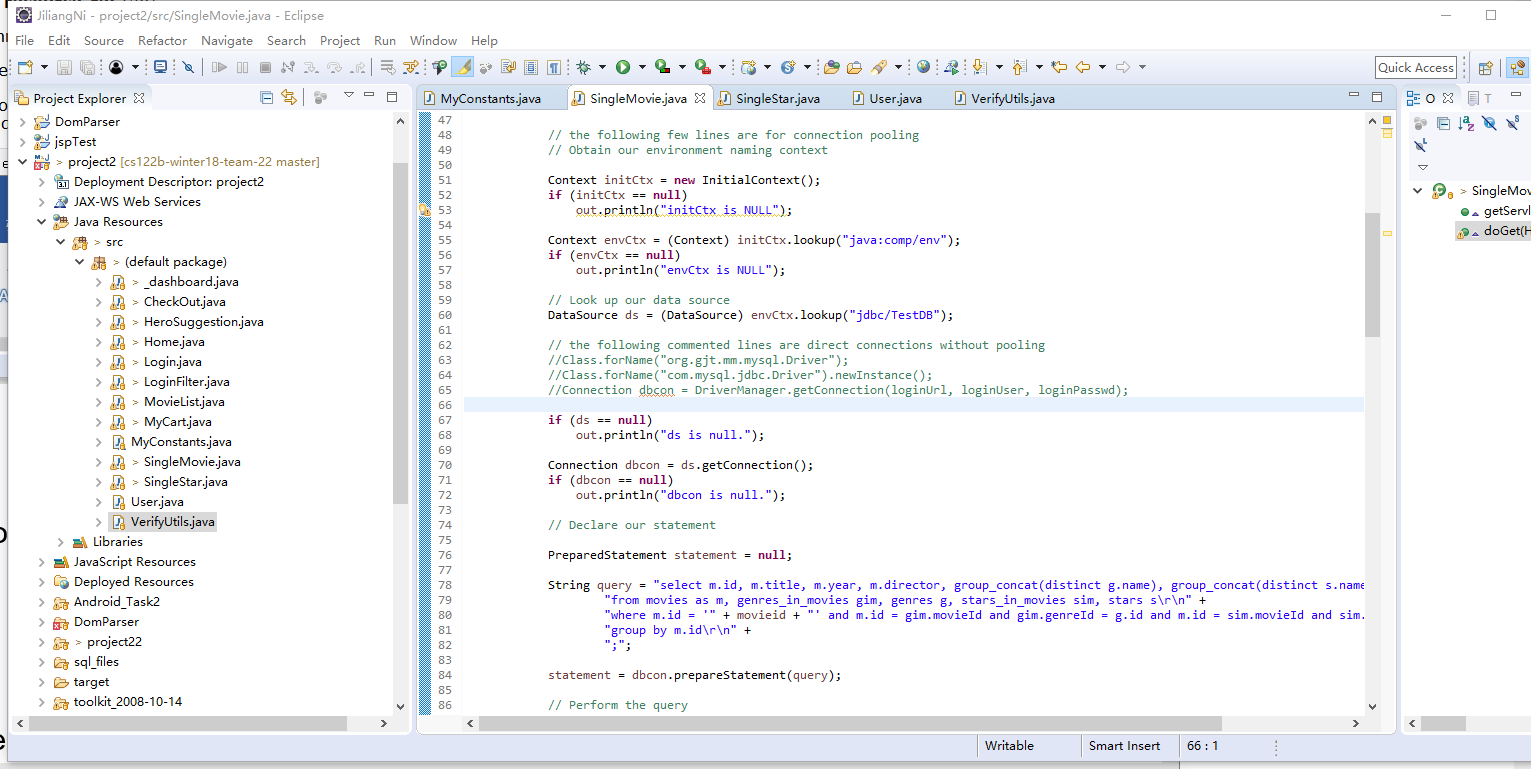
7.



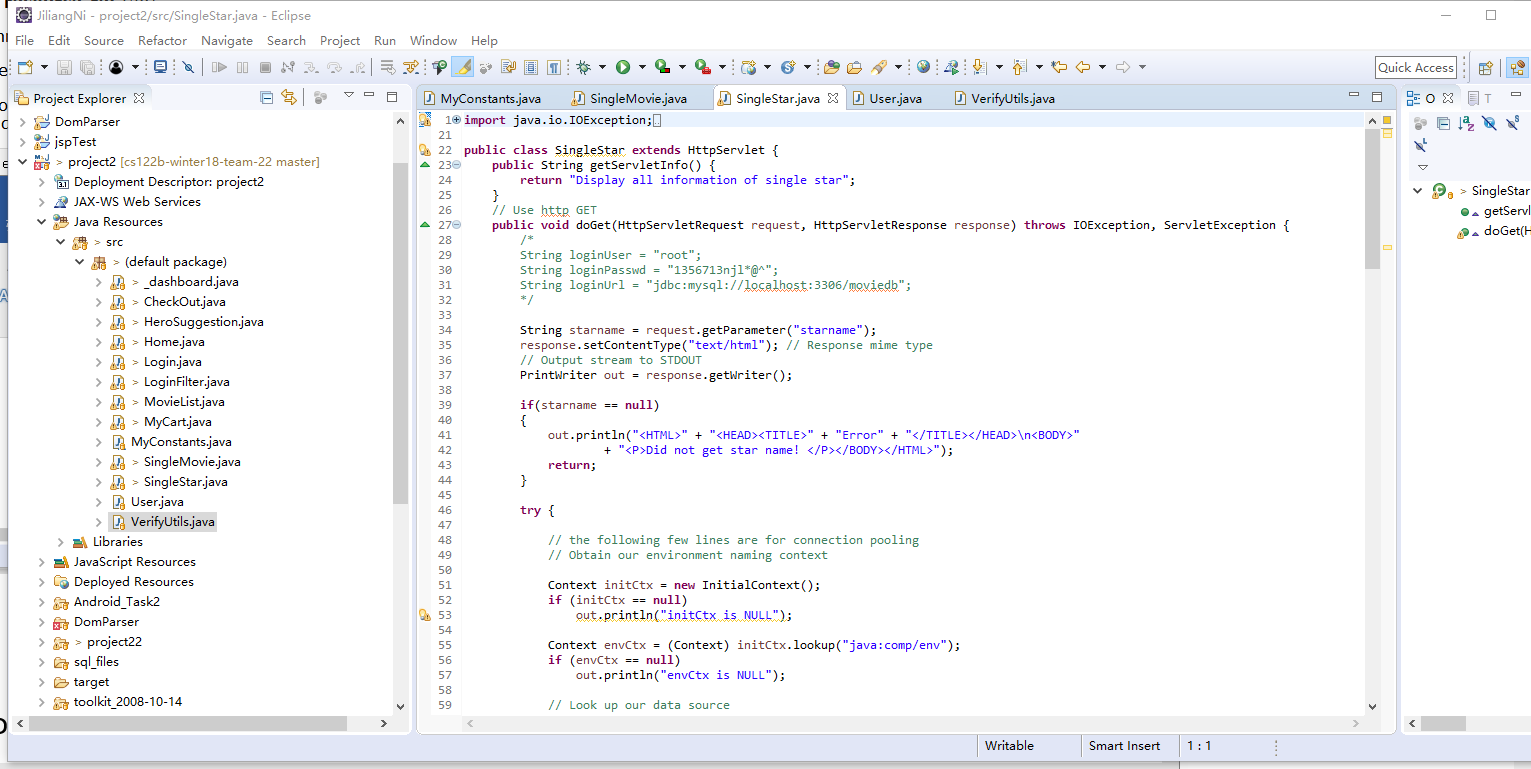
8.



9.



10.



* How did you use Prepared Statements?

I changed files to use PreparedStatement in all places where Statement is needed.

* File name, line numbers as in Github

1. /project2/src/\_dashboard.java

From line 133 to line 142.

From line 163 to line 172.

From line 245 to line 255.

From line 271 to line 278.

From line 327 to line 336.

From line 356 to line 365.

From line 524 to line 533.

From line 554 to line 563.

2. /project2/src/CheckOut.java

From line 102 to line 114.

3. /project2/src/HeroSuggestion.java

From line 157 to line 187.

4. /project2/src/Home.java

From line 181 to line 193.

5. /project2/src/Login.java

From line 128 to line 137.

6. /project2/src/MovieList.java

From line 126 to line 131.

From line 225 to line 245.

7. /project2/src/MyCart.java

From line 99 to line 106.

From line 126 to line 134.

8. /project2/src/SingleMovie.java

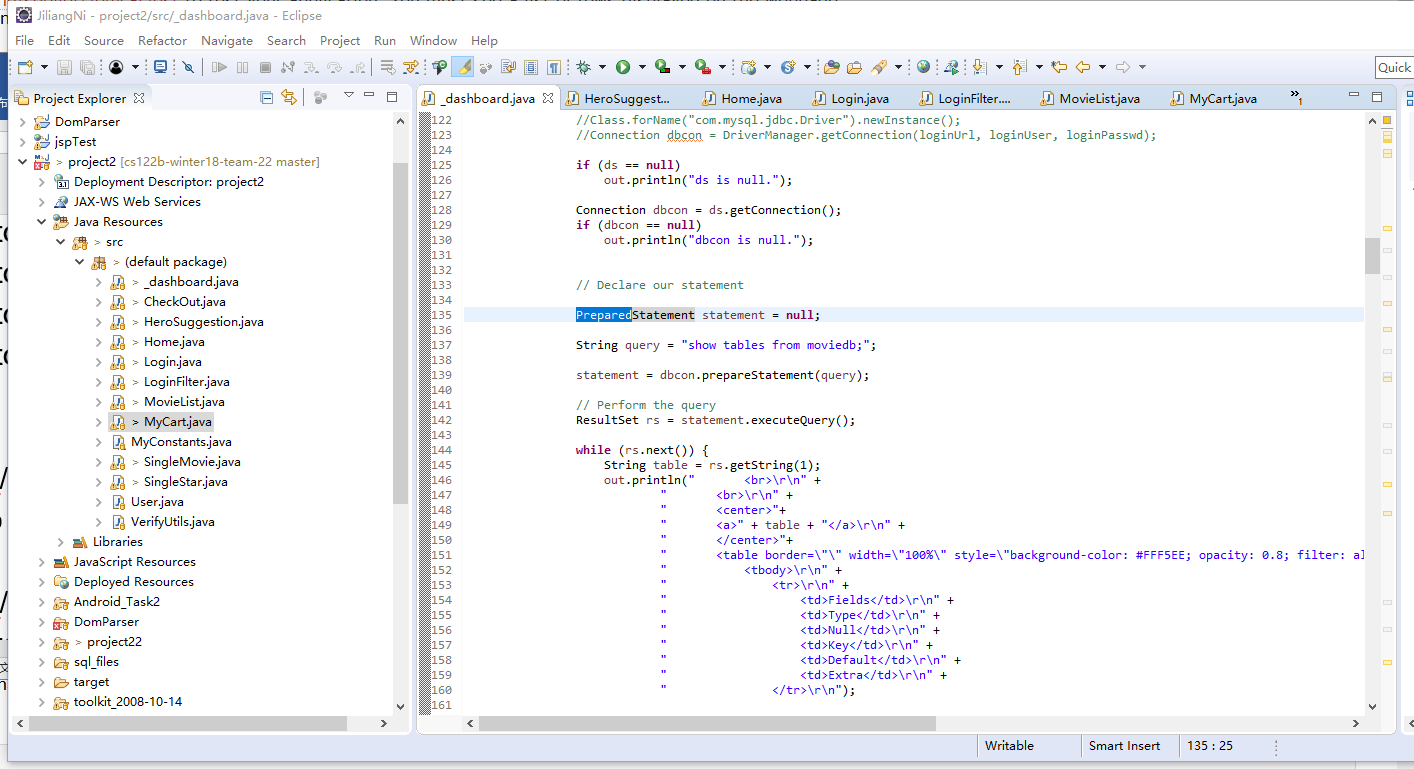
From 74 to line 87.

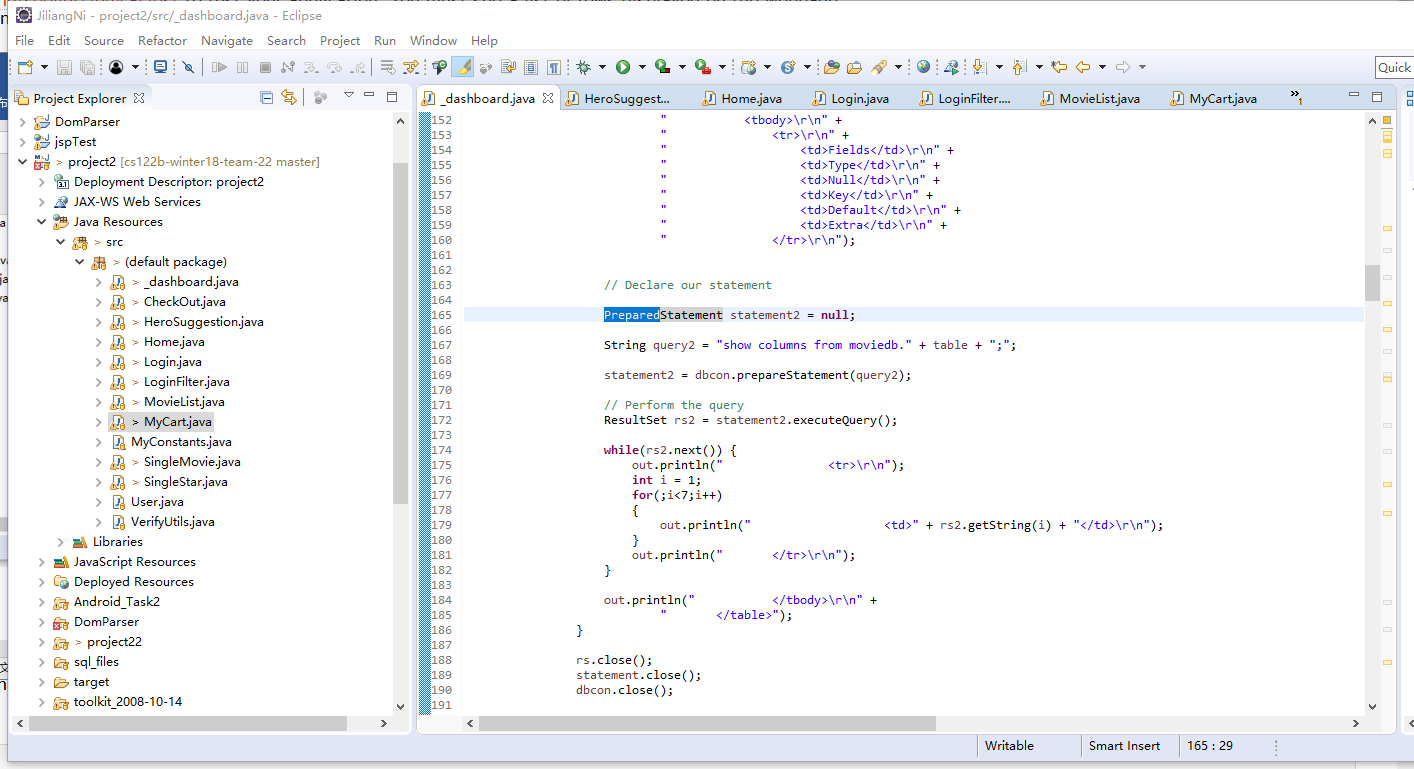
9. /project2/src/SingleStar.java

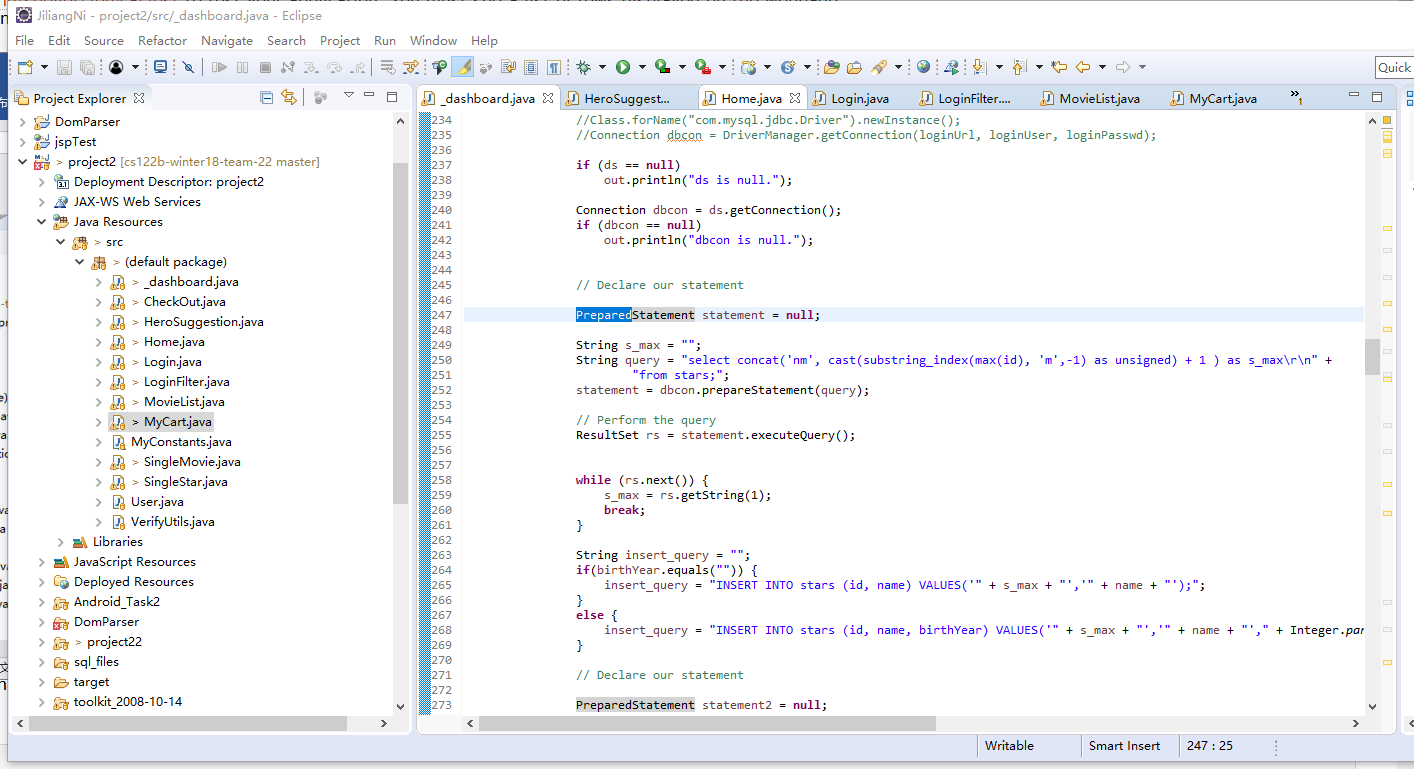
From 74 to line 87.

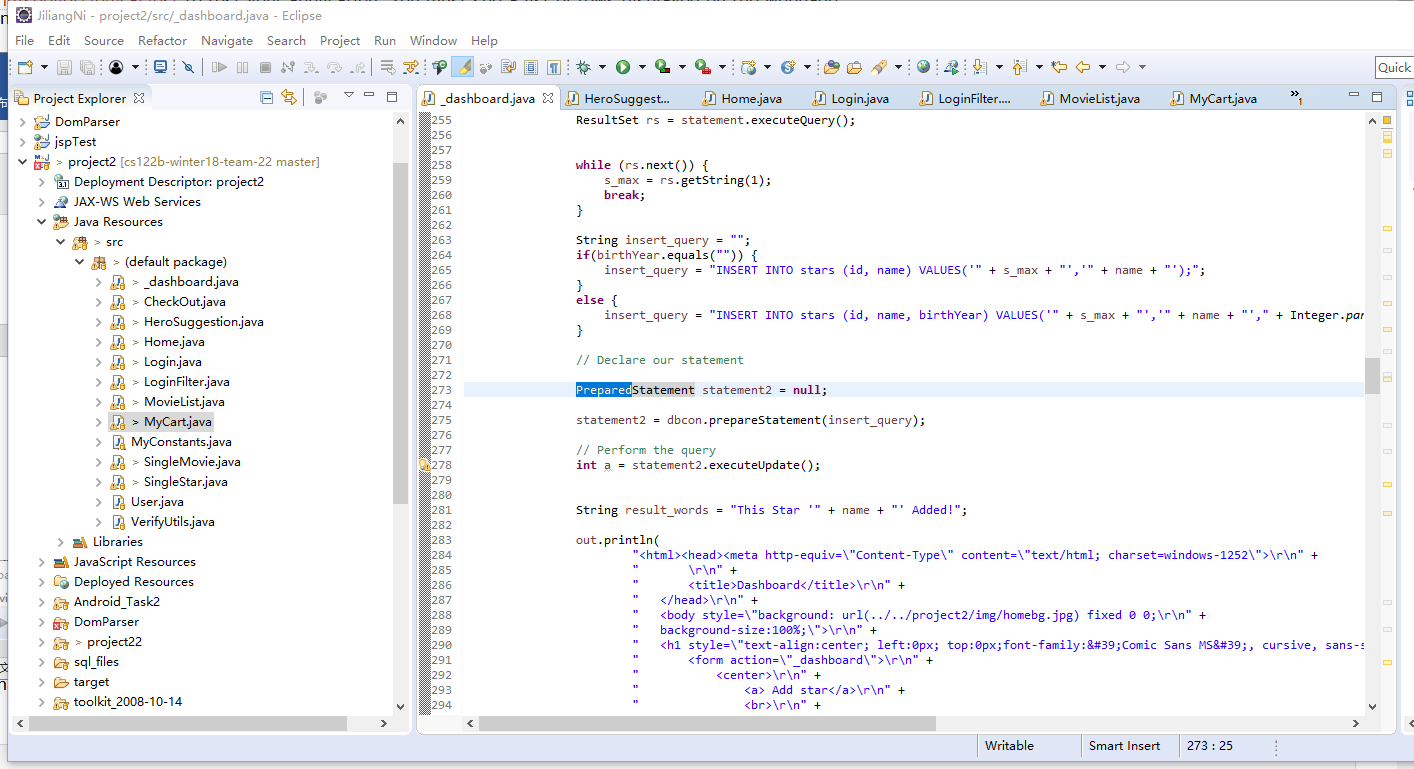
* Snapshots

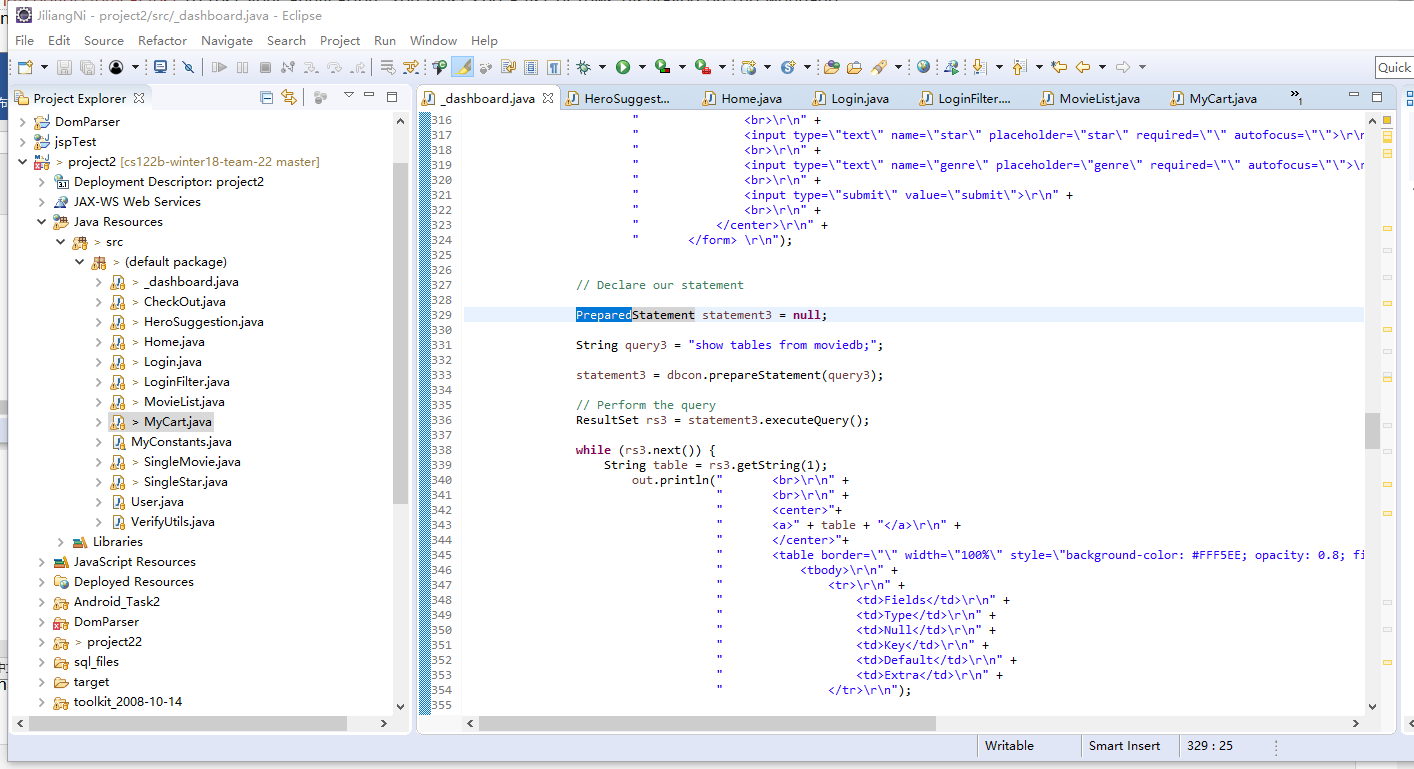
1.

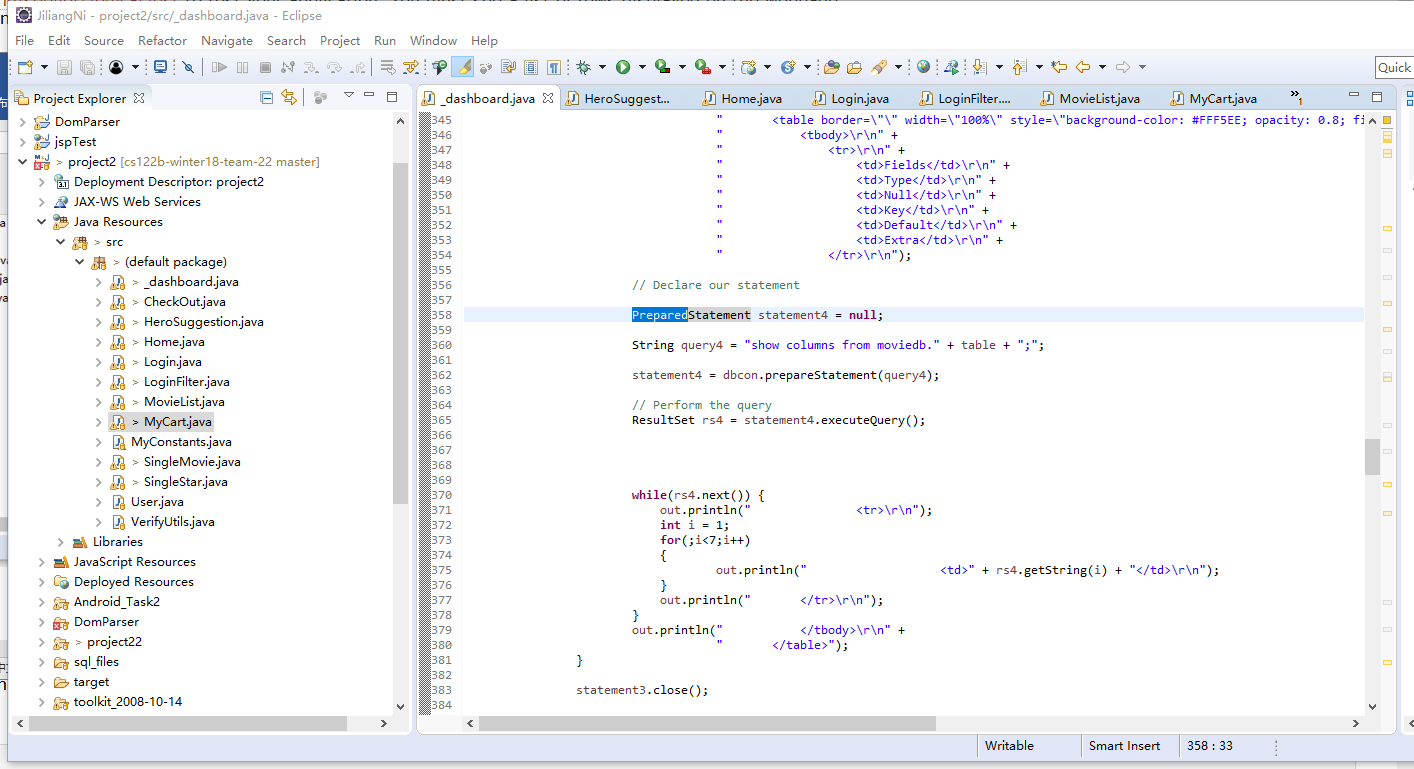


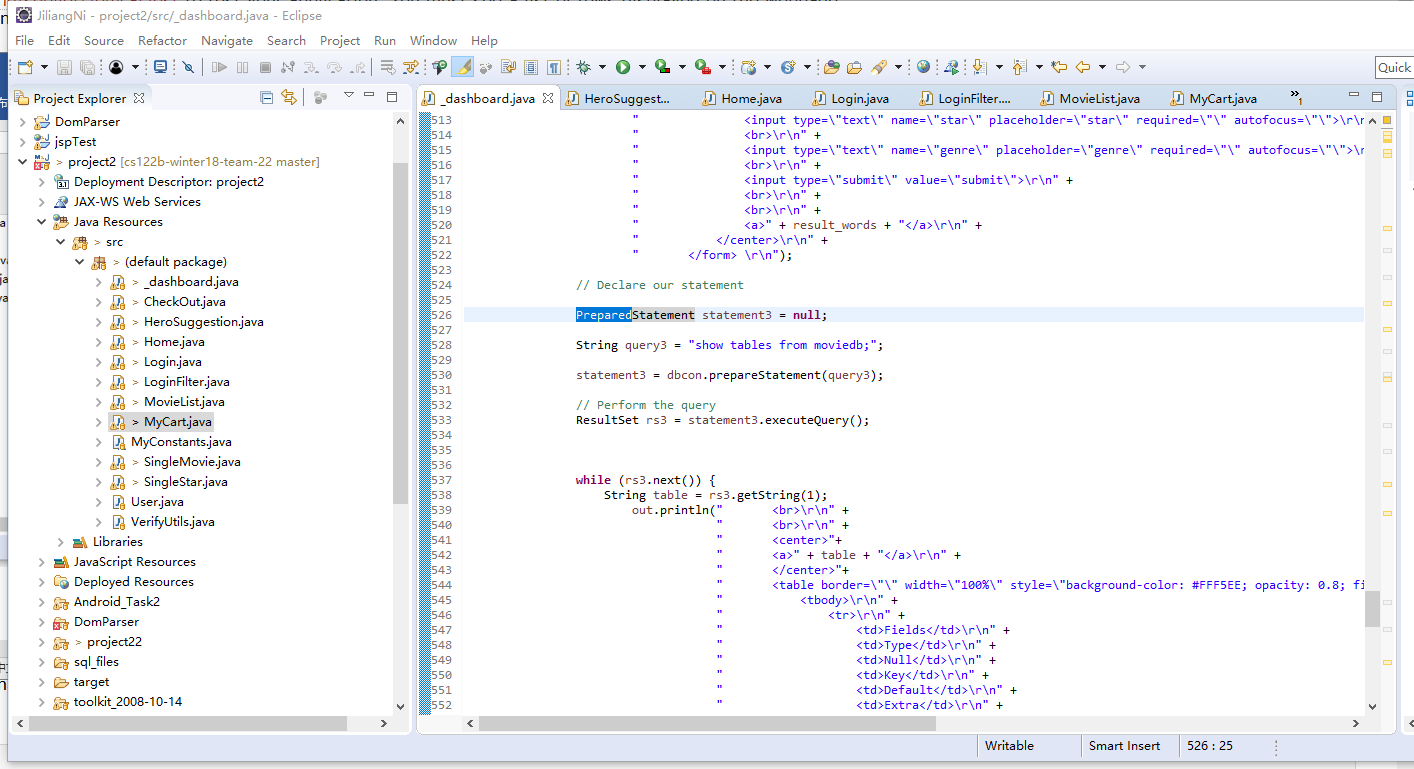


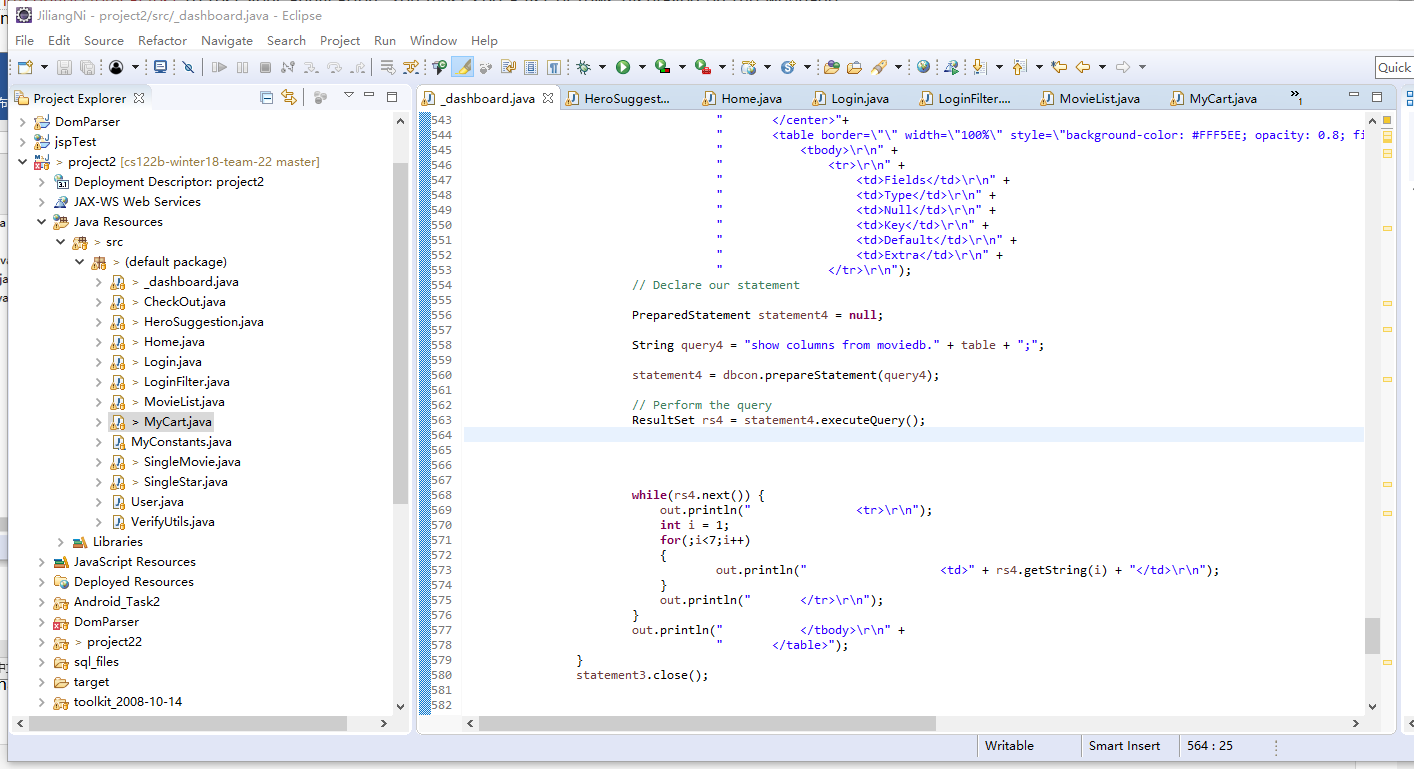




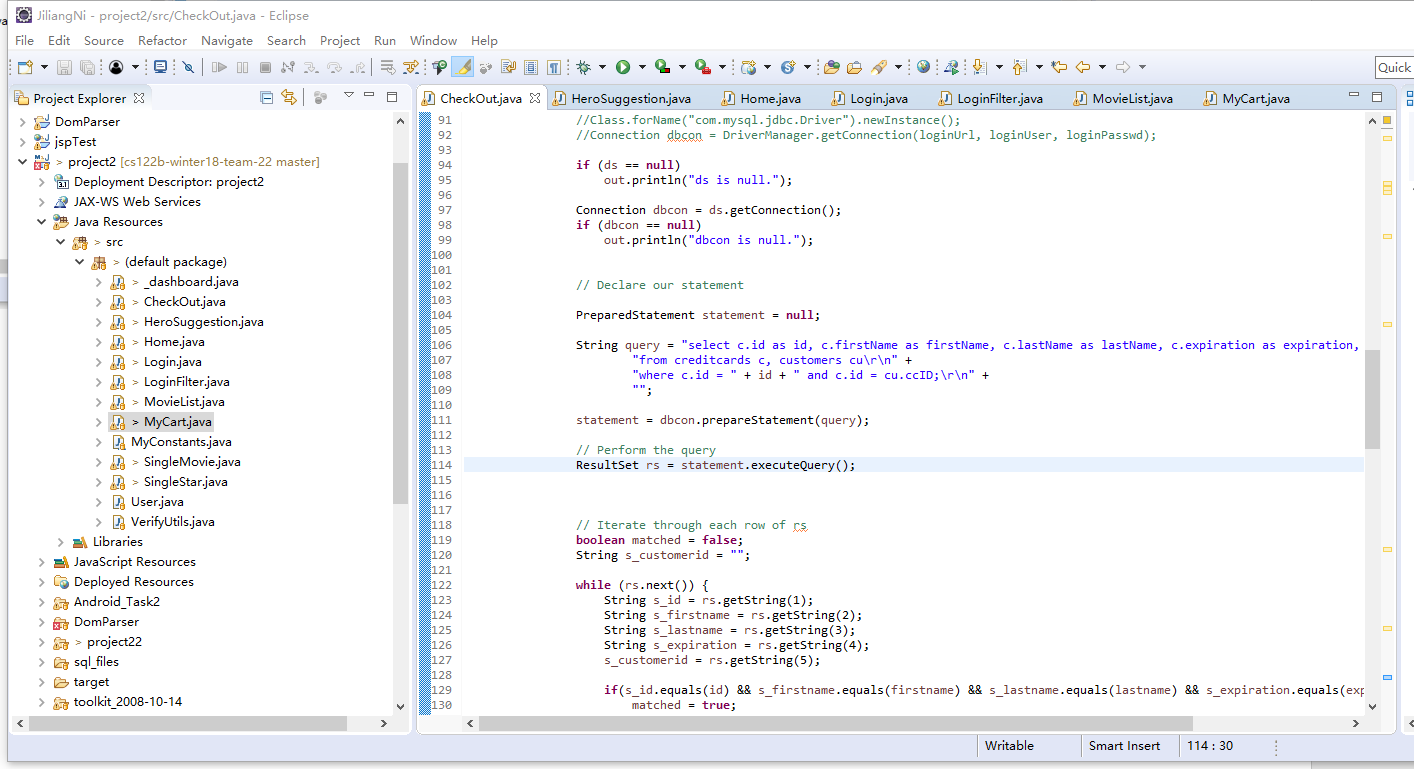




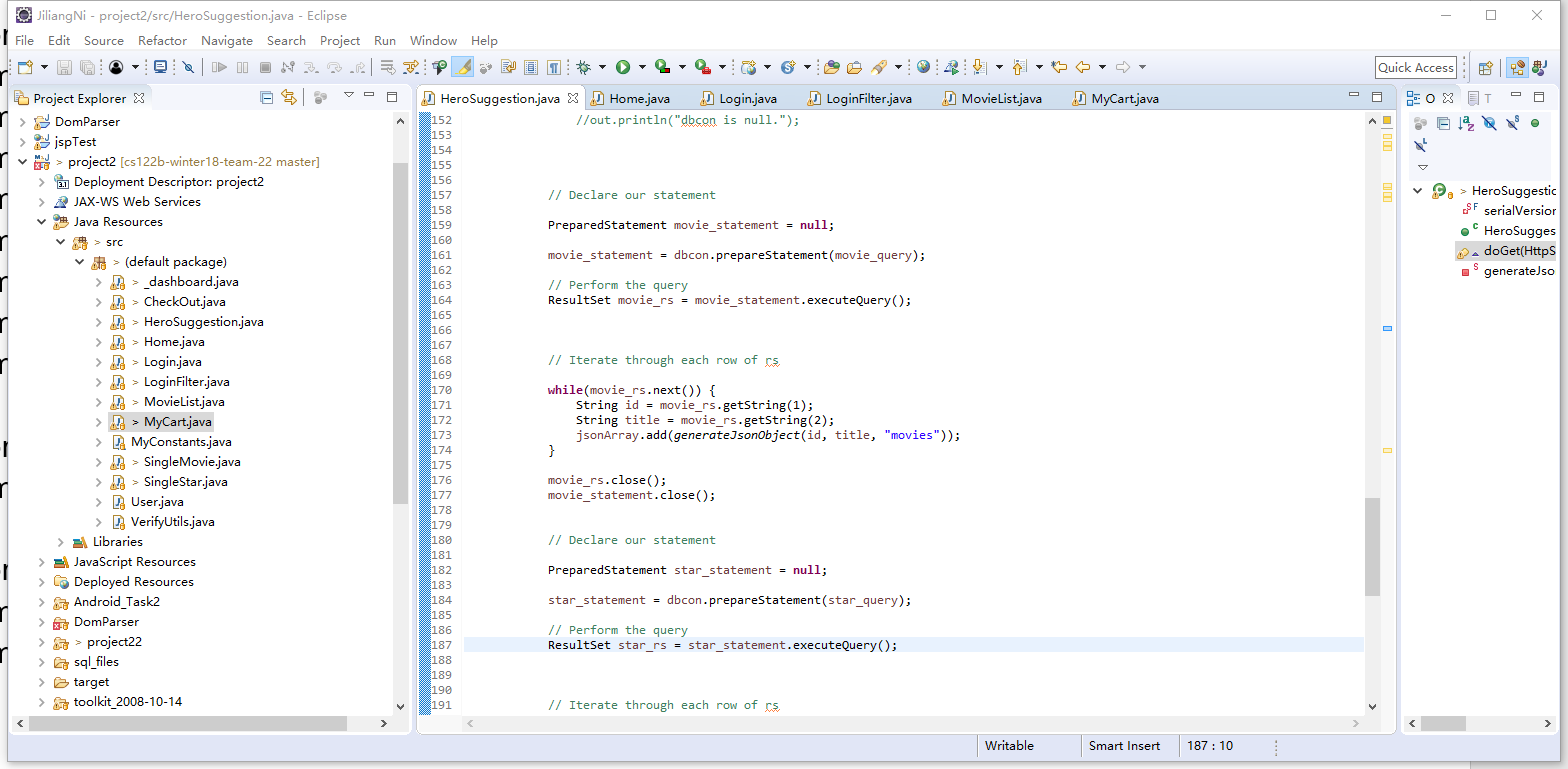




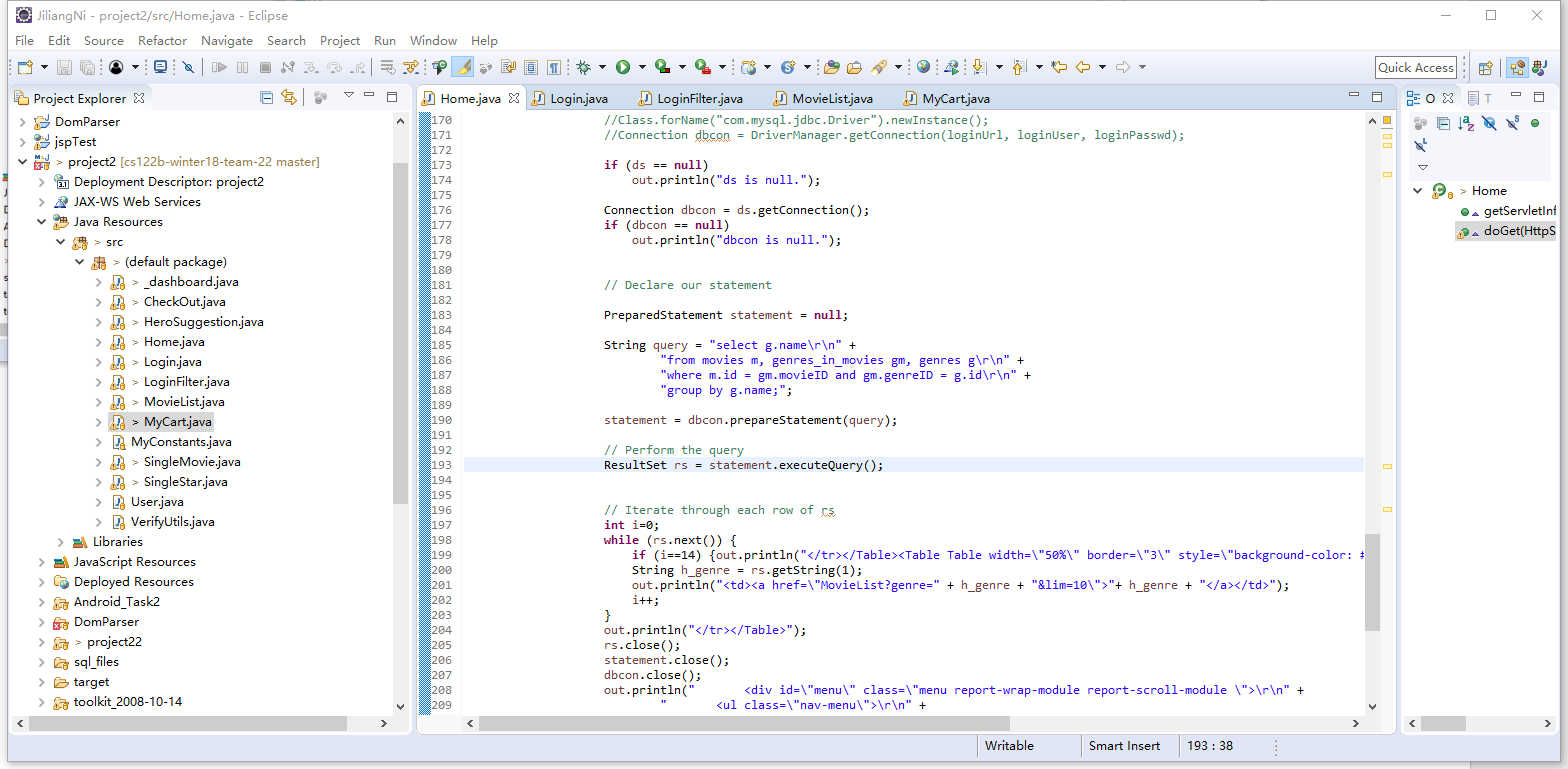
2.



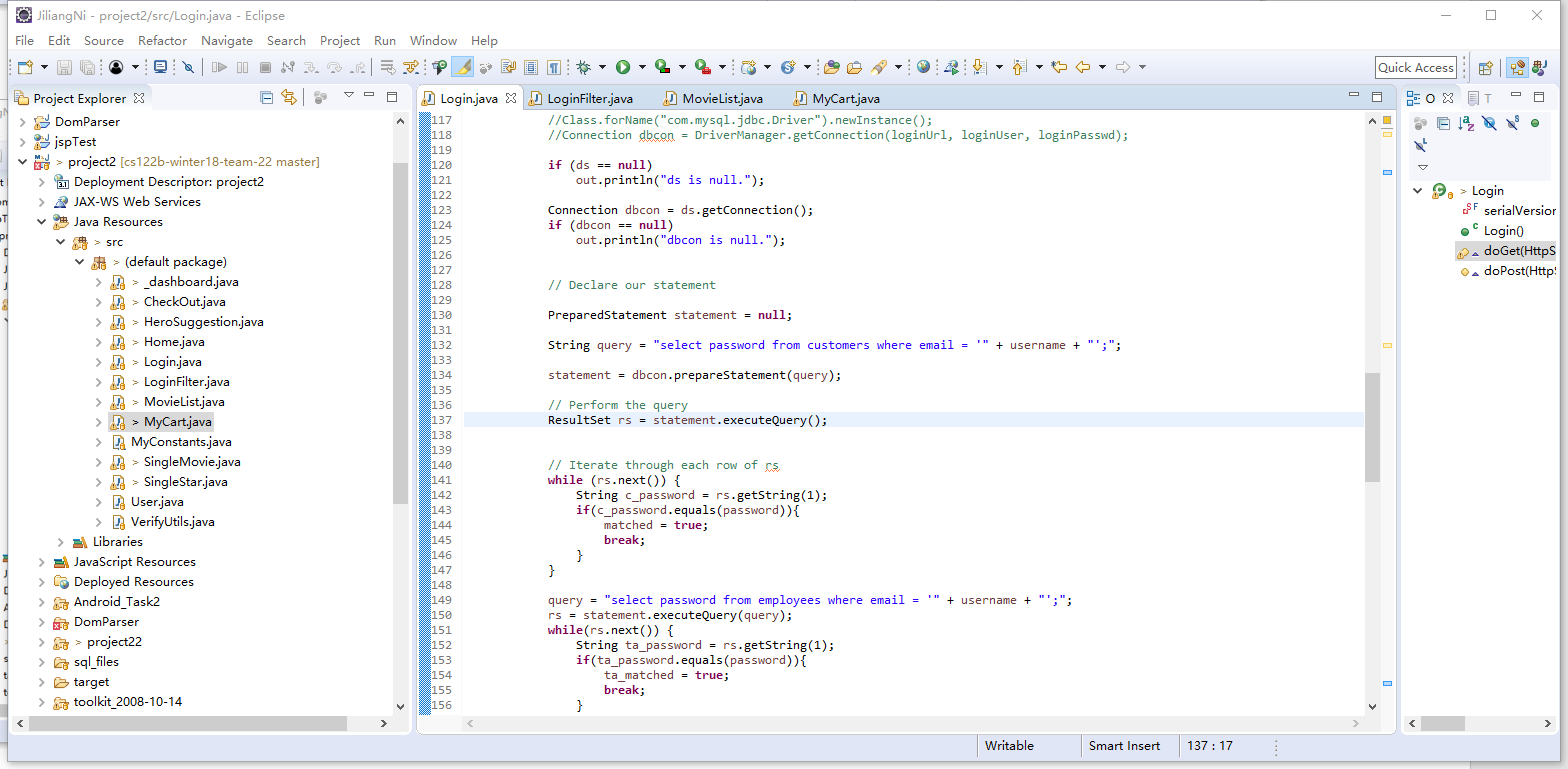
3.



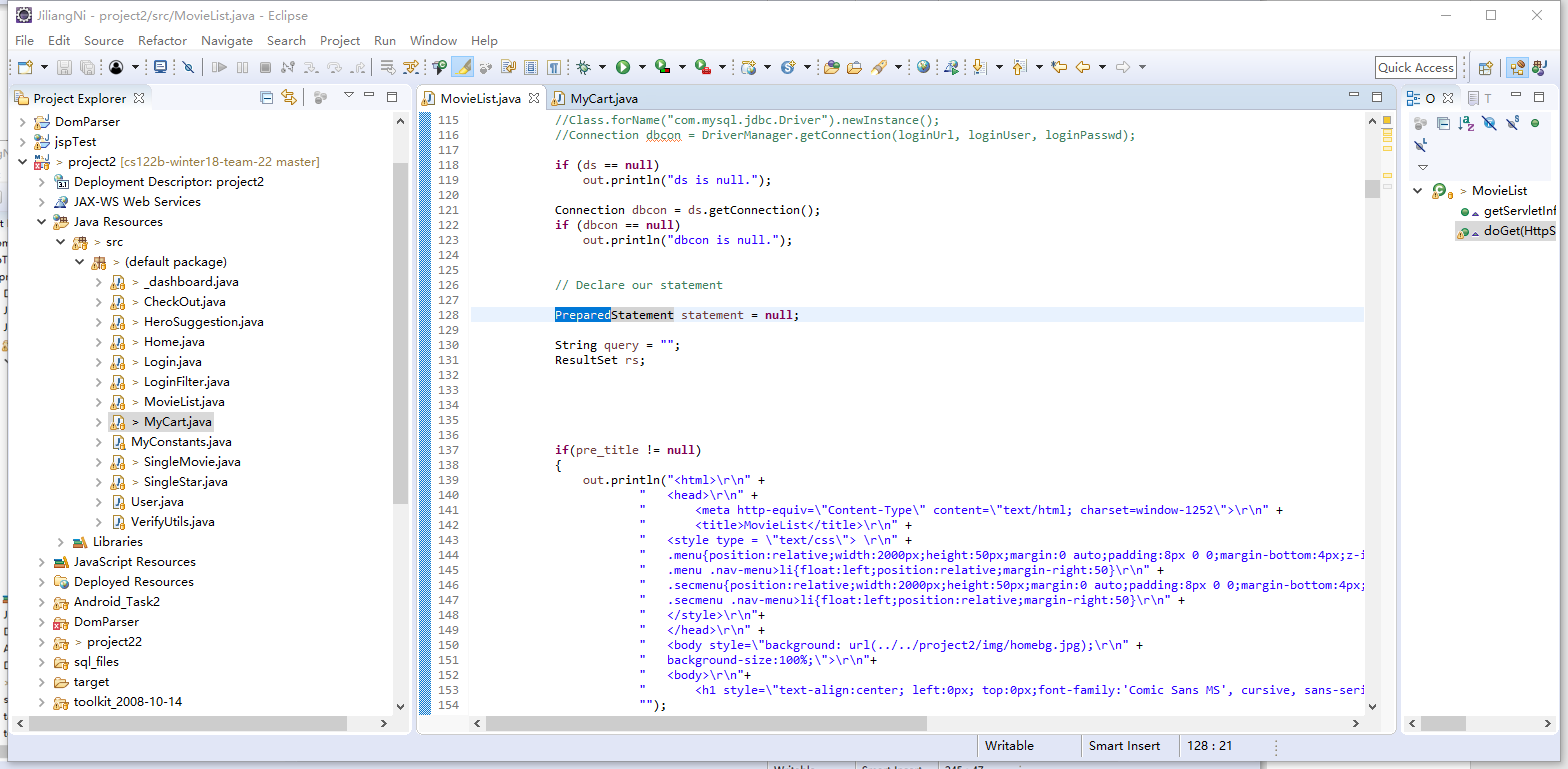
4.

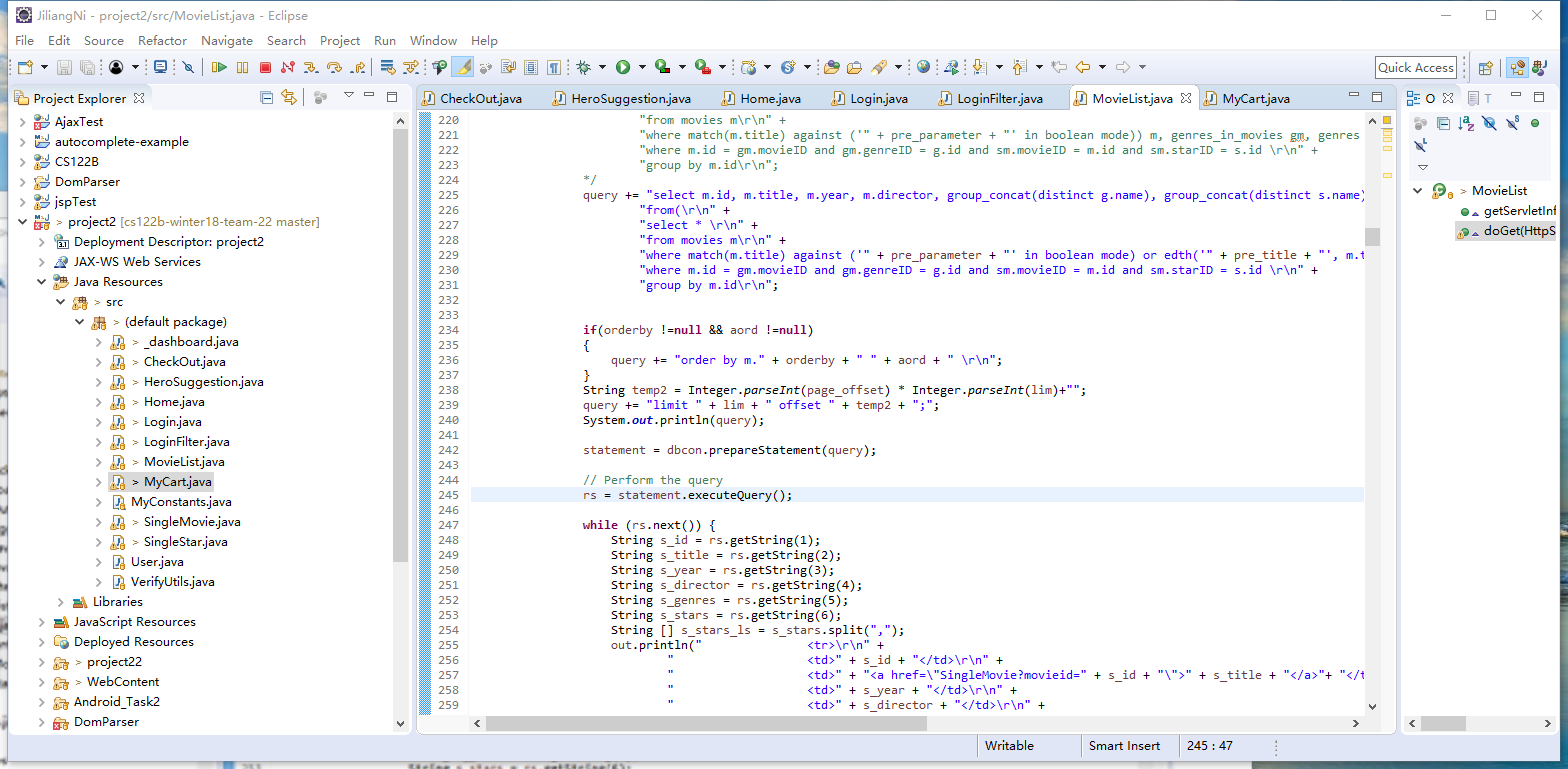


5.

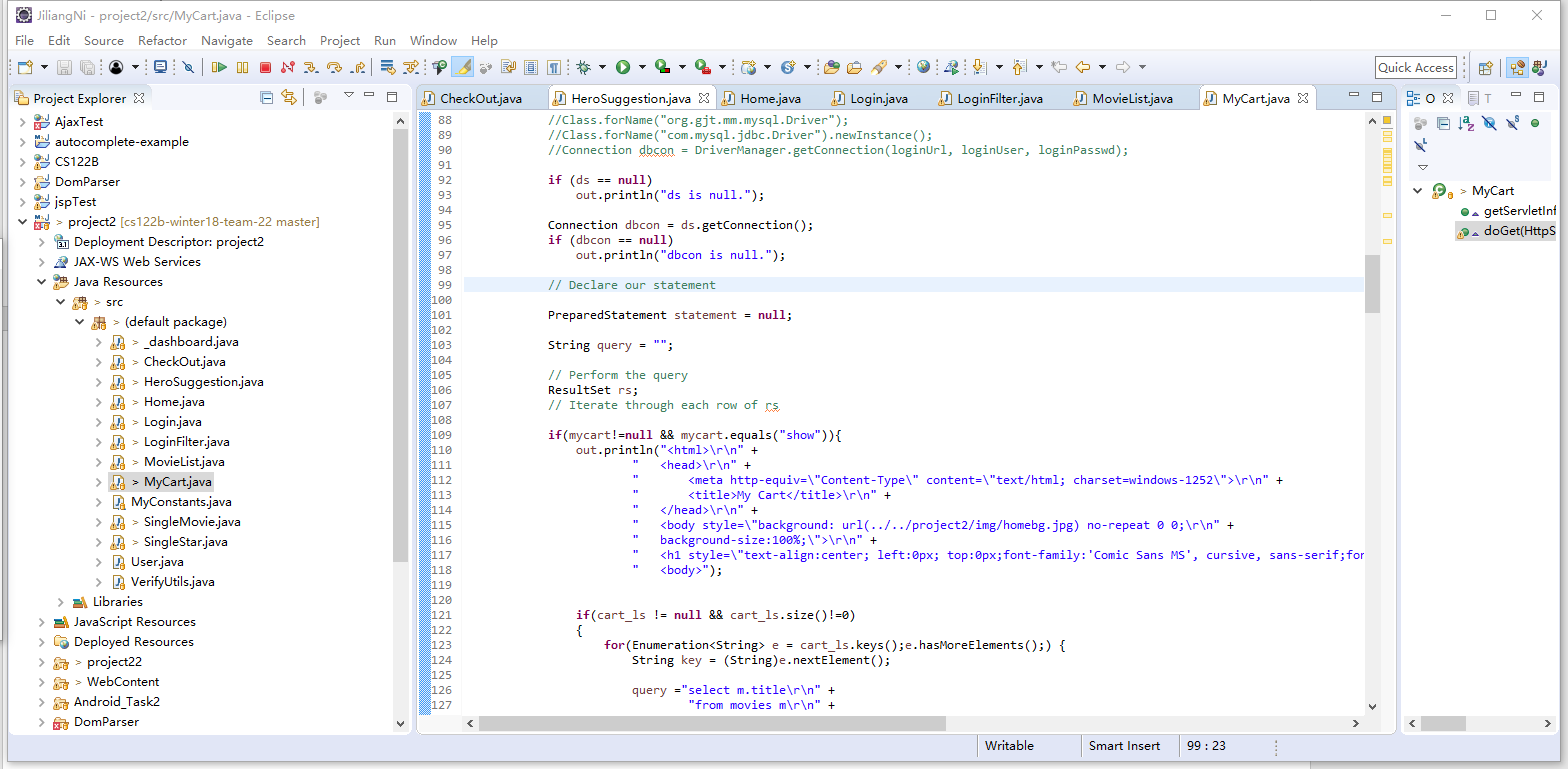


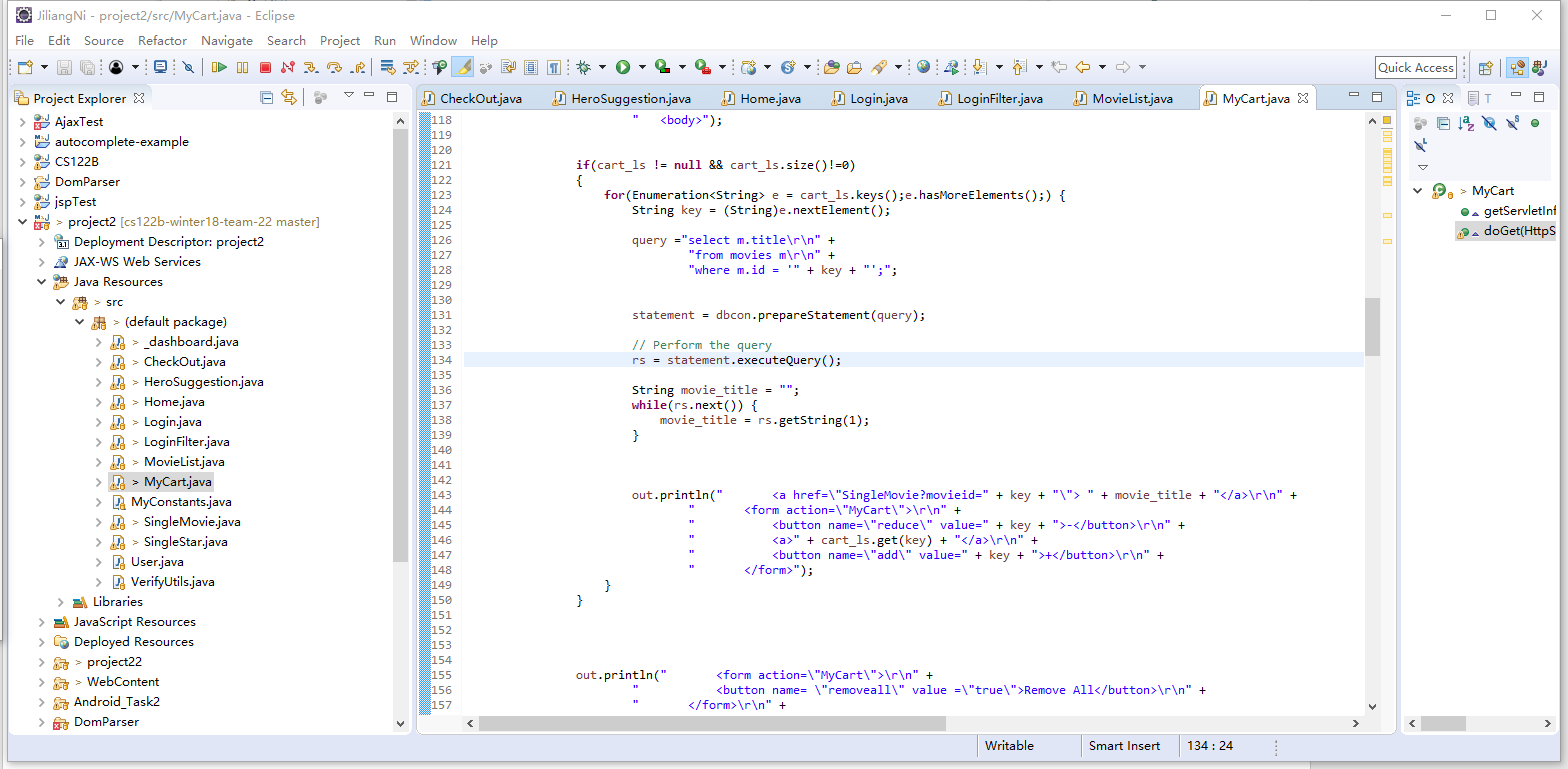
6.



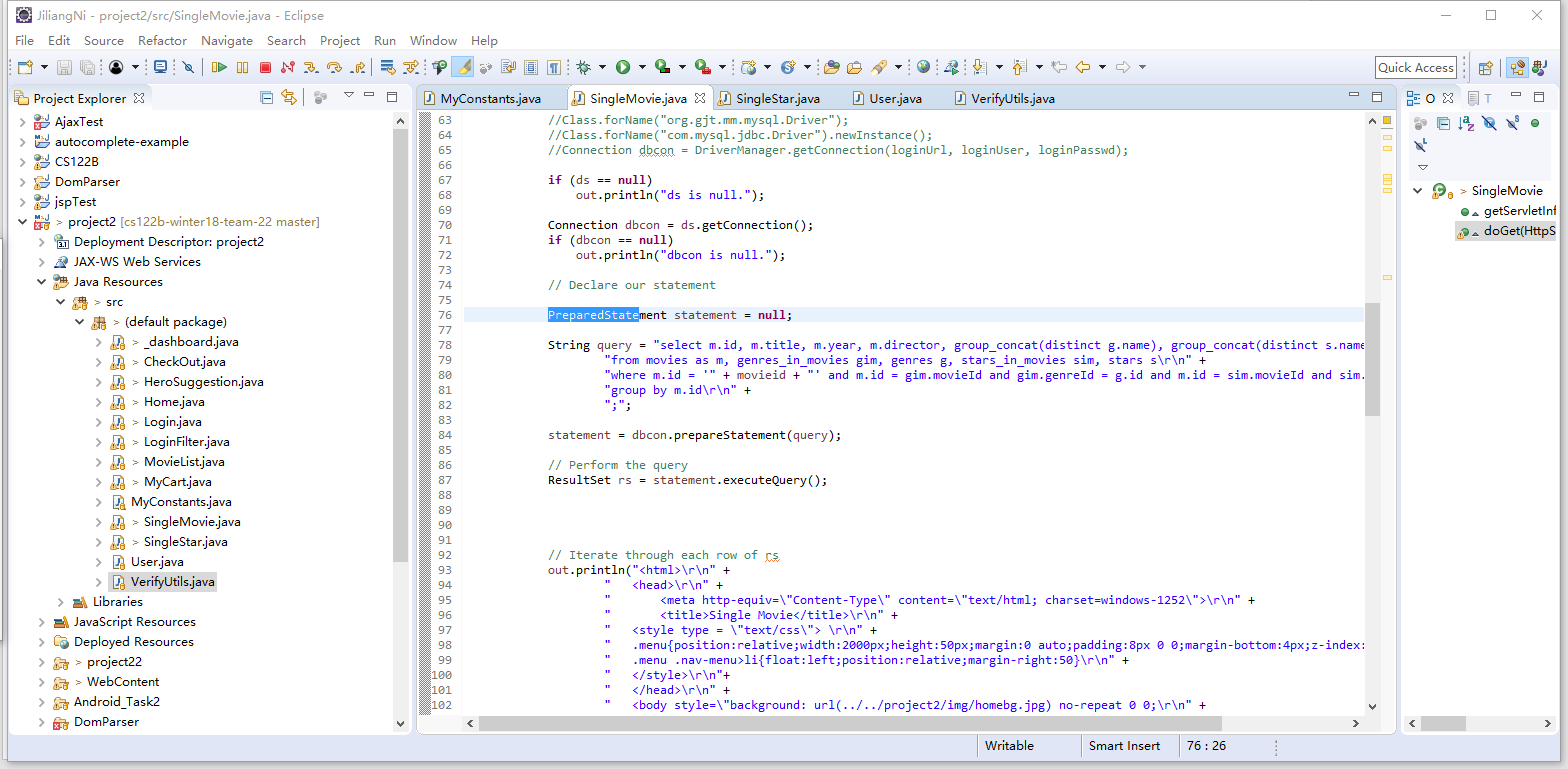


7.

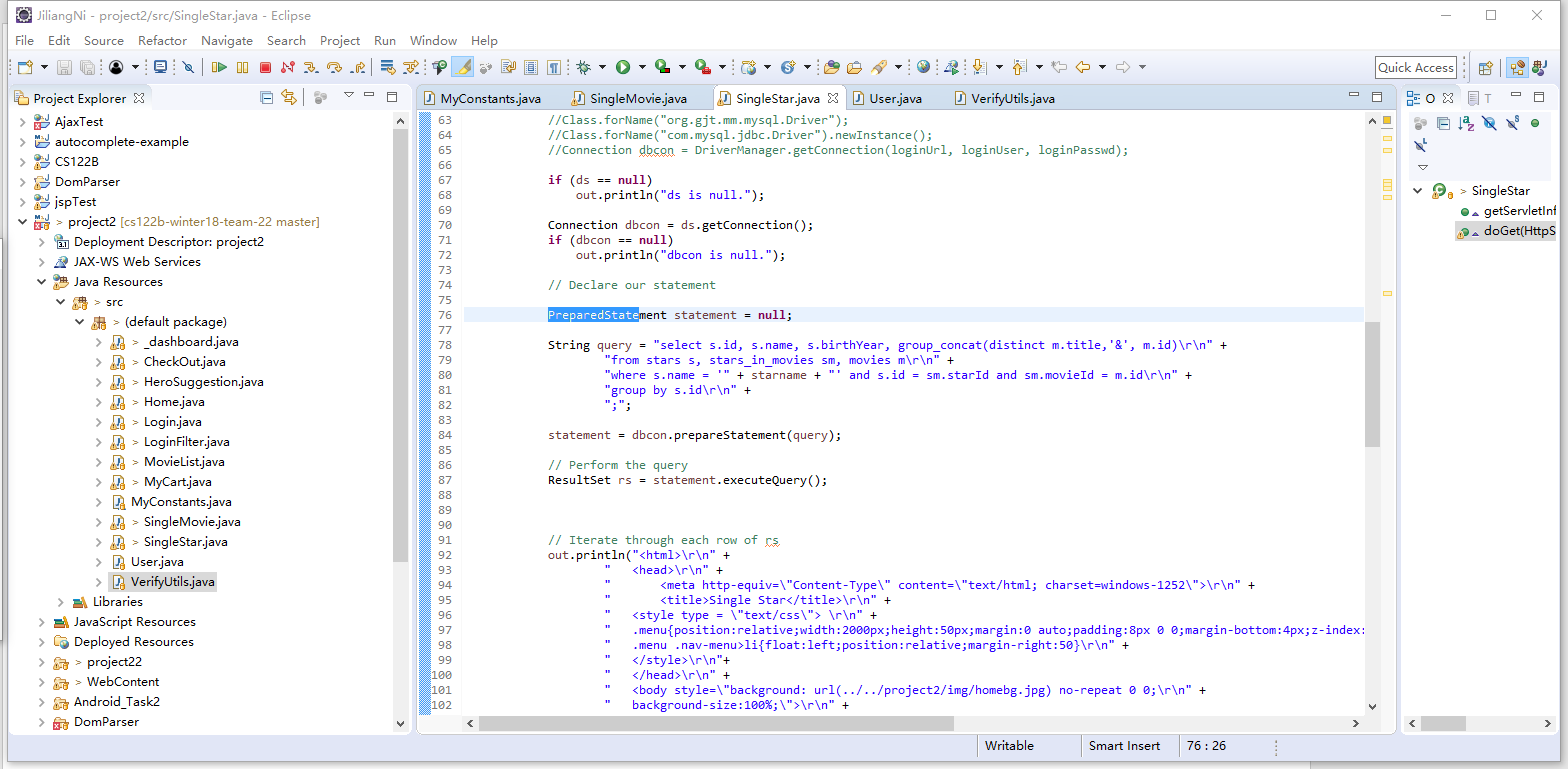




8.



9.



**Task 2**

* Address of AWS and Google instances

Address of load balancer:

Public IP: 54.153.49.96

Private IP: 172.31.7.154

Address of instance 1 master:

Public IP: 13.57.239.137

Private IP: 172.31.21.85

Address of instance 2 slave:

Public IP: 13.57.218.115

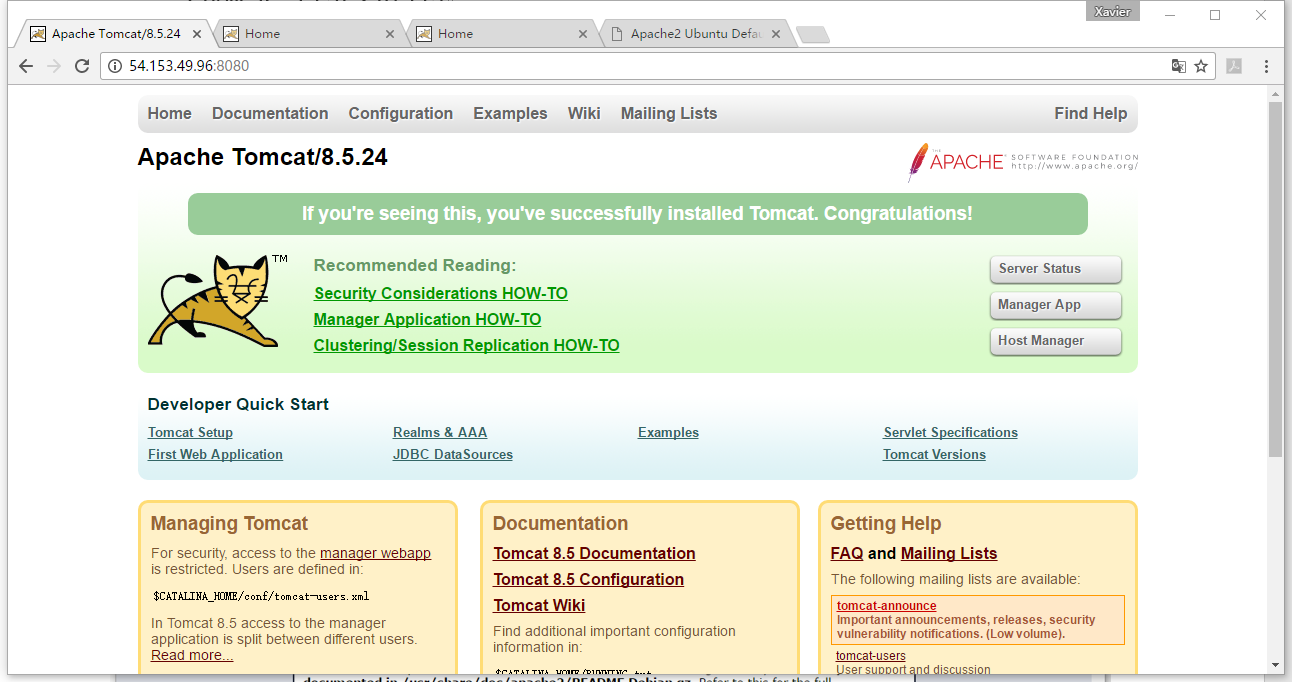
Private IP: 172.31.19.105

Address of Google instance:

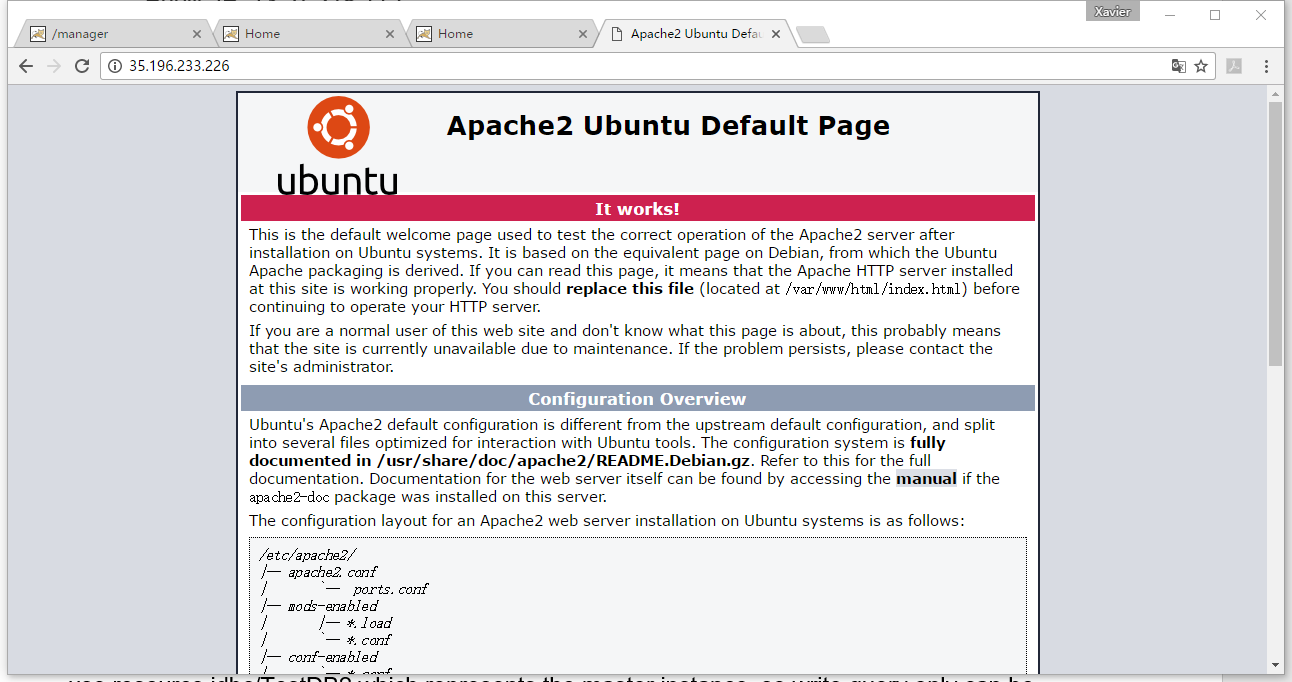
Public IP: 35.196.233.226

* Have you verified that they are accessible? Does Fablix site get opened both on Google’s 80 port and AWS’ 8080 port?

Yes, I have verified that they are accessible.



Yes my Fabilx site opens both on Google’s 80 port and AWS’ 8080 port.



* How connection pooling works with two backend SQL?

1. The load balancer will use apache2 first distribute queries to master or slave instance. Then, master or slave will use connection pooling, which I write the code in /rpoject2/WebContent/META-INF/context.xml

Each resource represents one connection. Jdbc/TestDB represents localhost’s mysql connection, while jdbc/TestDB2 represents the master instance’s mysql connection.

And the instance will determine which type of each query is. If it is read query, the instance will use jdbc/Test and just do the query in mysql of localhost. But if it is write query, the instance will use resource jdbc/TestDB2 which represents the master instance, so write query only can be run in master instance’s mysql database.

2. This is how I let the servlet to send write query to master instance’s mysql database.

3. This is how I let the servlet to send read query to its own mysql database.

4. This is how I let the load balancer to distribute request to master or slave.

* + File name, line numbers as in Github

1. /project2/WebContent/META-INF/context.xml

All lines in this file.

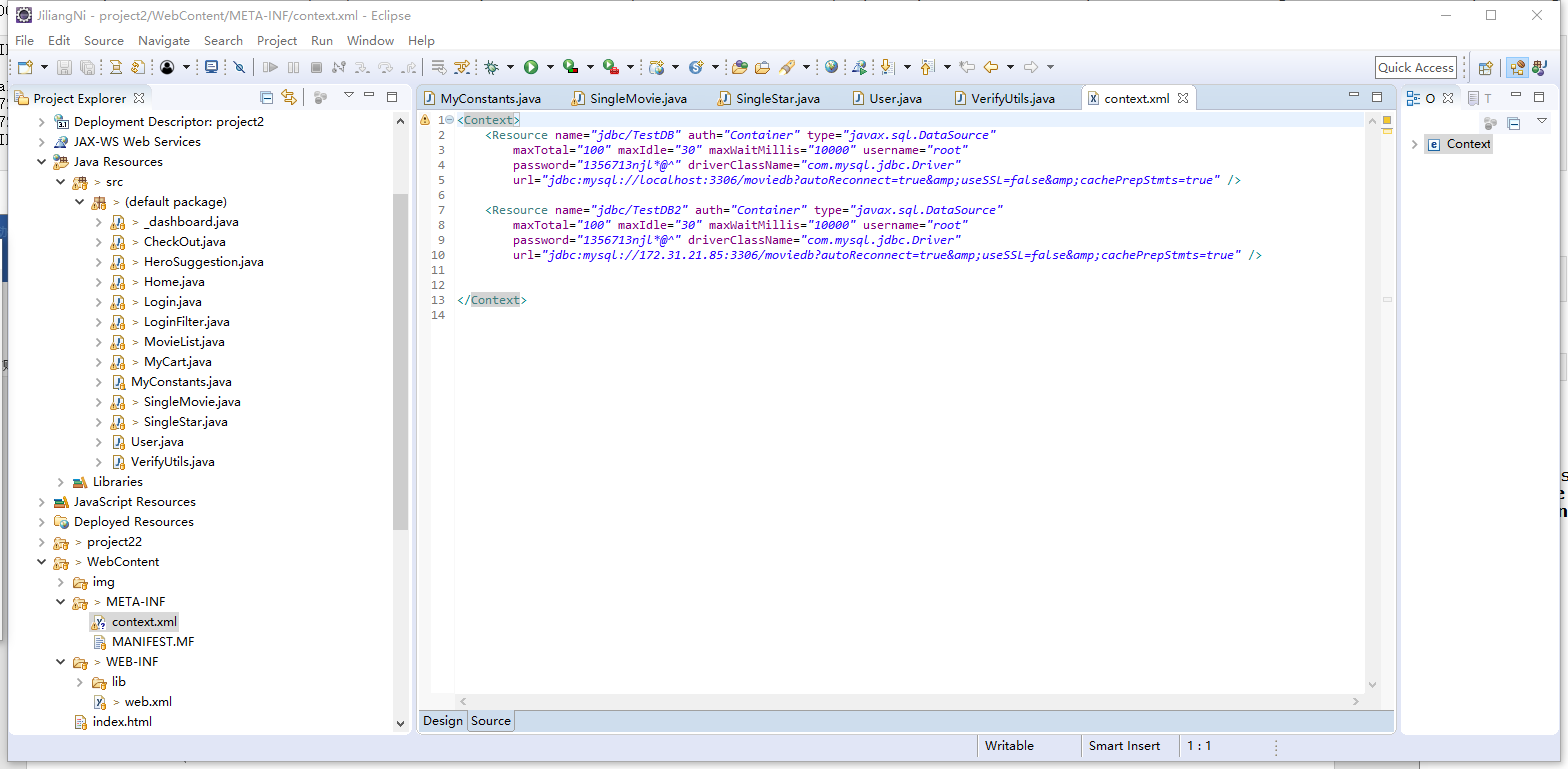
2. /project2/src/\_dashboard.java

From line 425 to line 464.

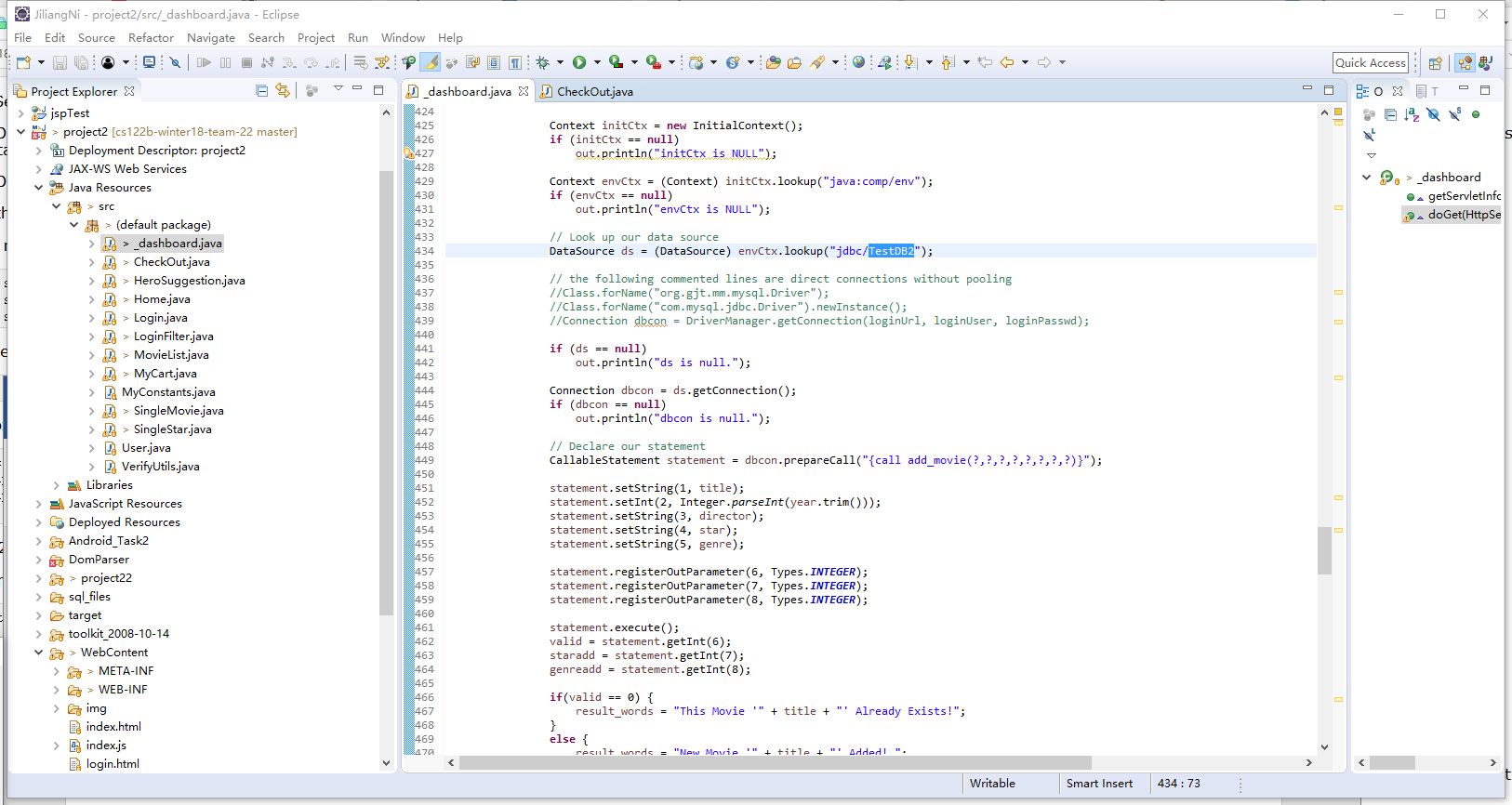
3. /project2/src/SingleMovie.java

From line 51 to line 87.

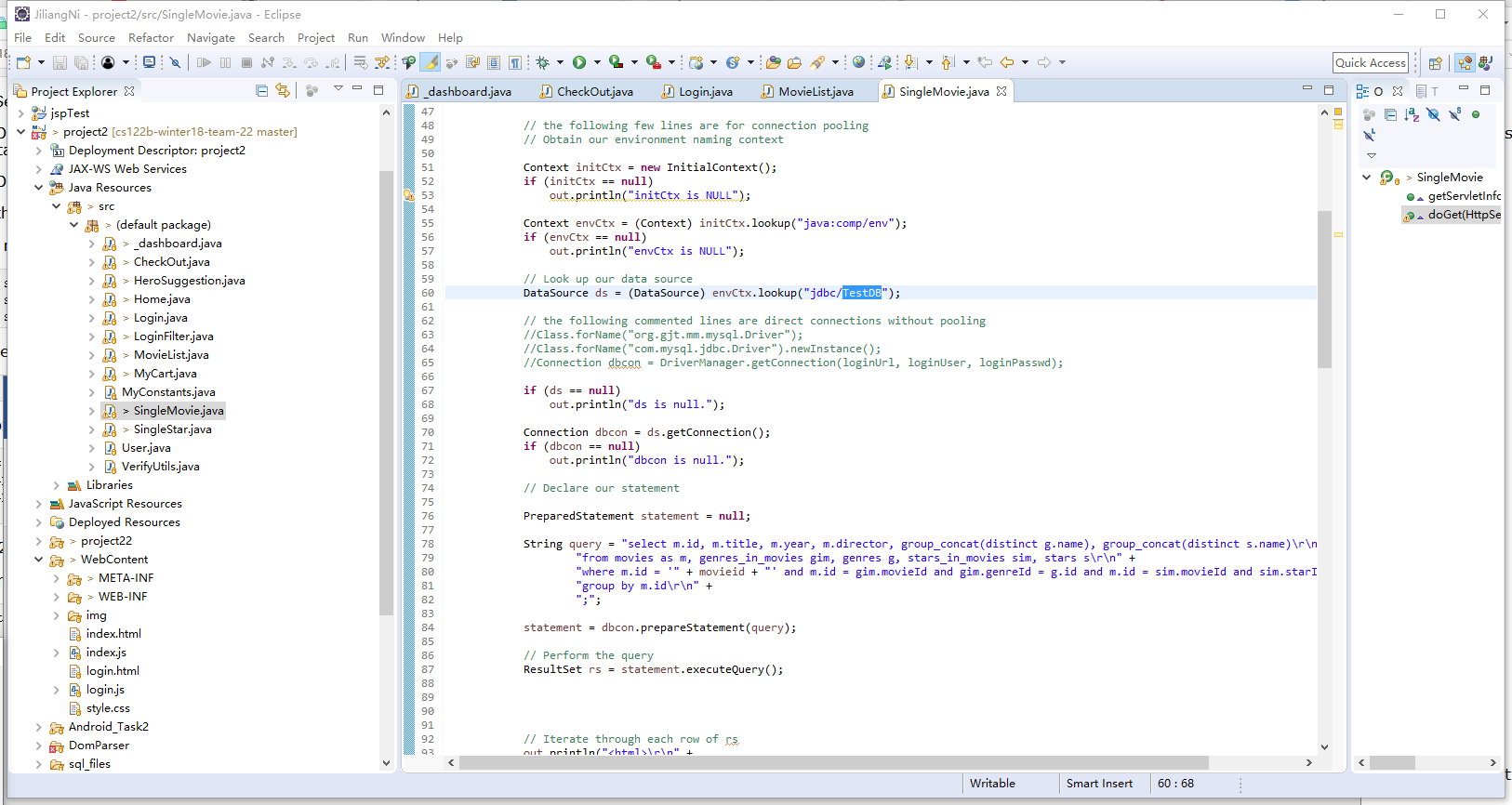
* + Snapshots

1.  


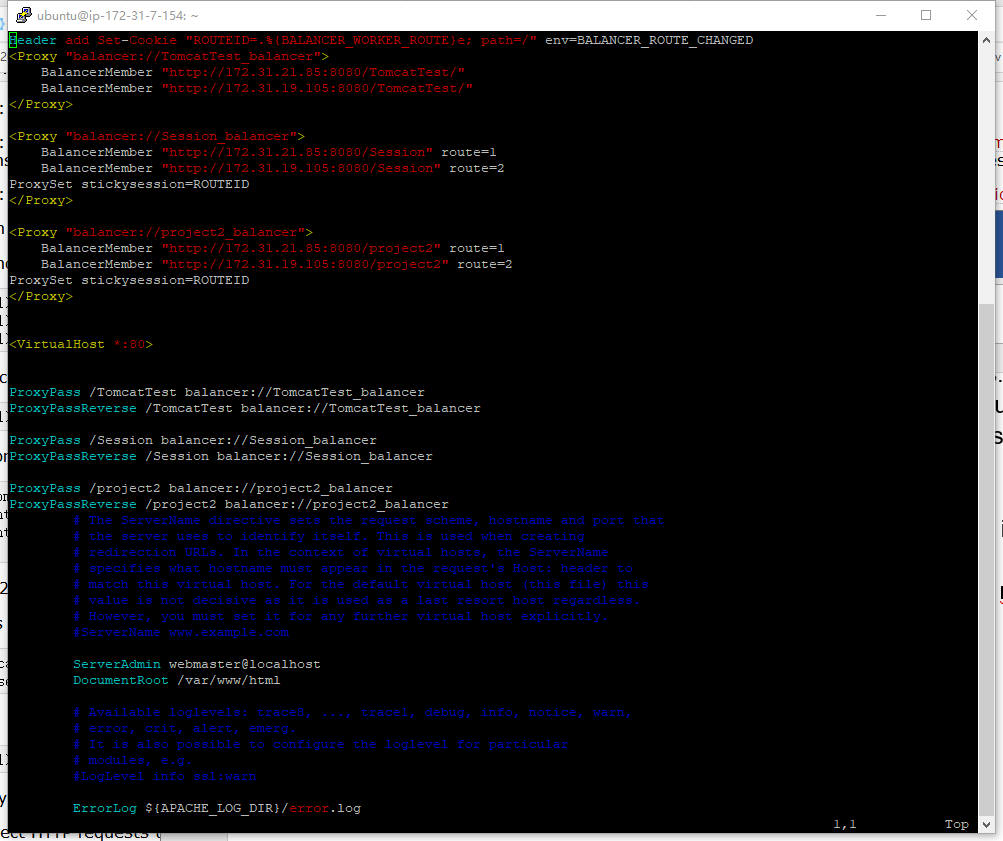
2.



3.



4.



* How read/write requests were routed?

1. The load balancer will use apache2 first distribute queries to master or slave instance. Then, master or slave will use connection pooling, which I write the code in /rpoject2/WebContent/META-INF/context.xml

Each resource represents one connection. Jdbc/TestDB represents localhost’s mysql connection, while jdbc/TestDB2 represents the master instance’s mysql connection.

And the instance will determine which type of each query is. If it is read query, the instance will use jdbc/Test and just do the query in mysql of localhost. But if it is write query, the instance will use resource jdbc/TestDB2 which represents the master instance, so write query only can be run in master instance’s mysql database.

2. This is how I let the servlet to send write query to master instance’s mysql database.

3. This is how I let the servlet to send read query to its own mysql database.

* + File name, line numbers as in Github

1. /project2/WebContent/META-INF/context.xml

All lines in this file.

2. /project2/src/\_dashboard.java

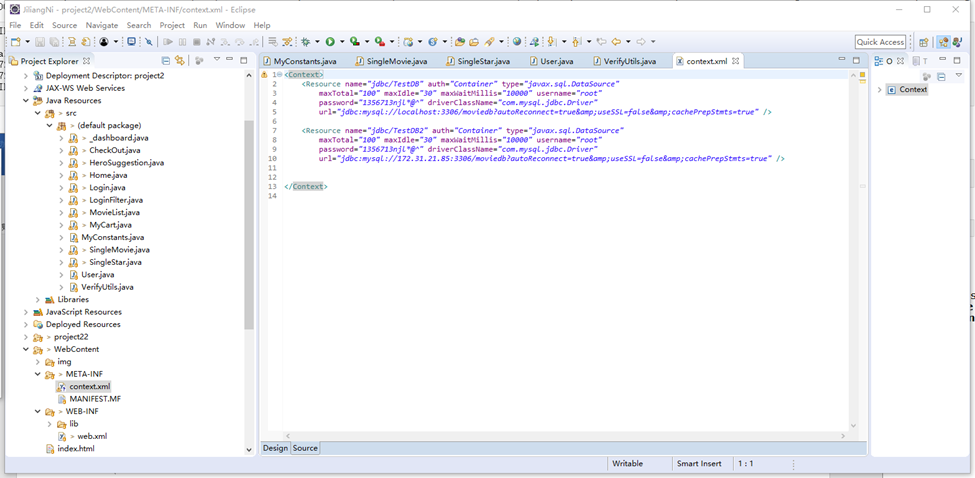
From line 425 to line 464.

3. /project2/src/SingleMovie.java

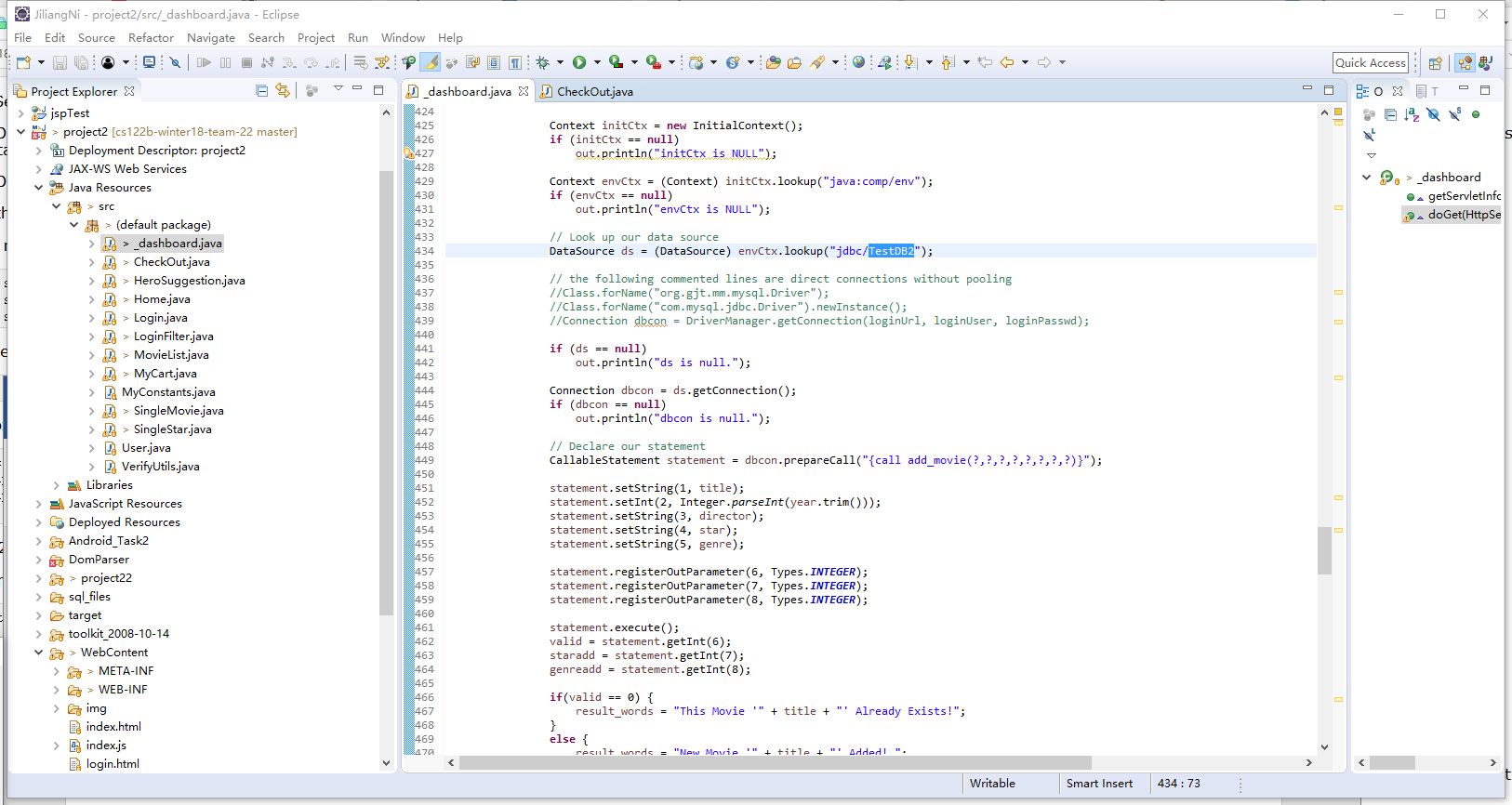
From line 51 to line 87.

* Snapshots

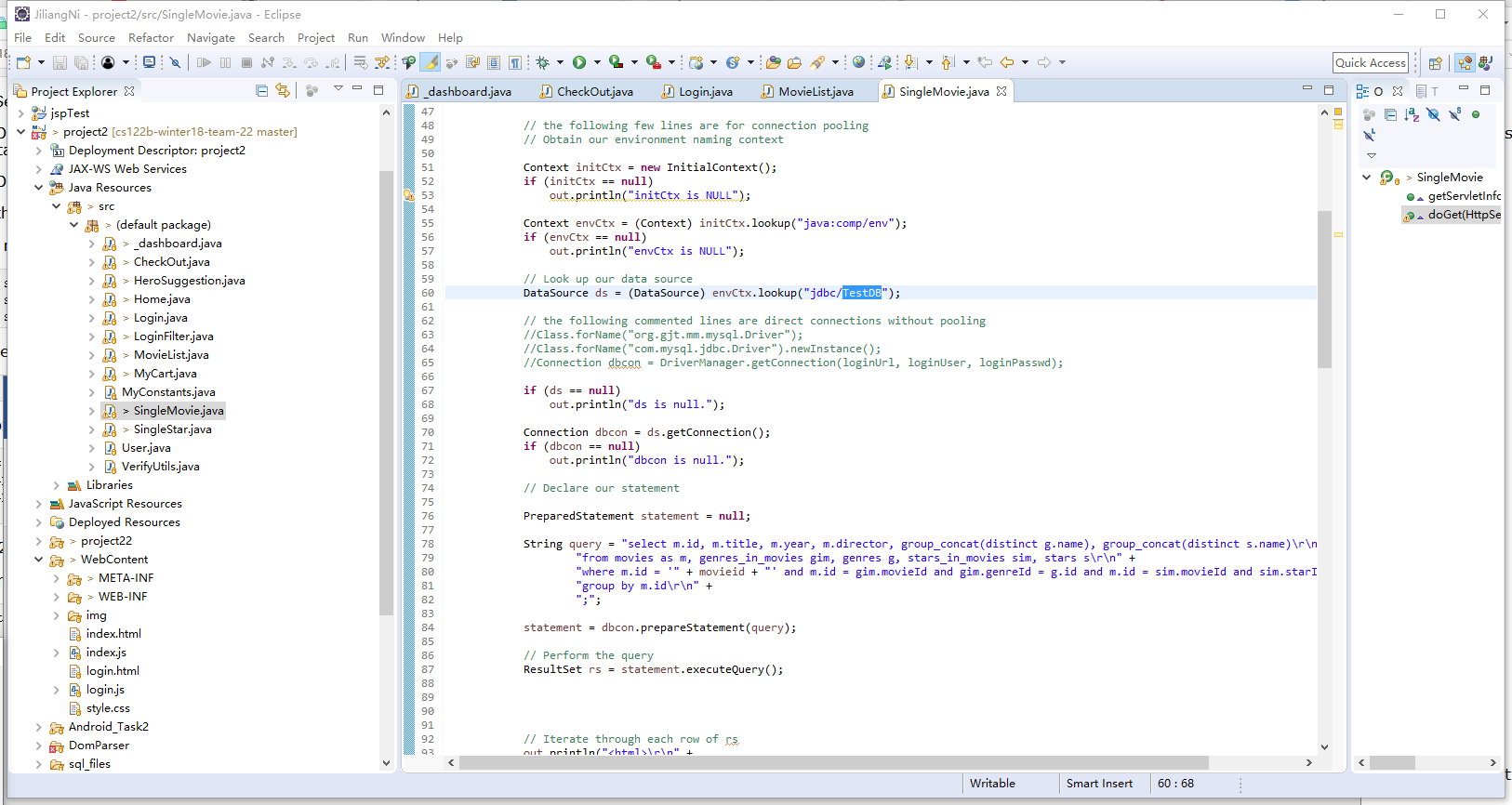
1.



2.



3.



**Task 3**

* Have you uploaded the log file to Github? Where is it located?

Yes. It is located at \*\cs122b-winter18-team-22\project5\_files\Task3.3 final report

And each TJ and TS log files are in each directory for each case like case1, case2 and so on.

* Have you uploaded the HTML file to Github? Where is it located?

Yes. It is located at \*\cs122b-winter18-team-22\project5\_files\Task3.3 final report\jmeter\_report.html

And the graph of results are also attached in same directory with jmeter\_report.html file.

* Have you uploaded the script to Github? Where is it located?

Yes. It is located at \*\cs122b-winter18-team-22\project5\_files\Task3.3 final report\get.py

* Have you uploaded the WAR file and README to Github? Where is it located?

Yes.

The WAR file is located at \*\cs122b-winter18-team-22\project5\_files\Task3.3 final report\project2.war

The README.txt file is located at \*\cs122b-winter18-team-22\project5\_files\Task3.3 final report\README.txt