Team 24: Stephen Abrahim, Brian Robbins

ALL GITHUB CODE IS THE CODE USED FOR INSTANCE #2 AND #3 WAR FILE (No https etc)

**Task 1**

* How did you use connection pooling?

Connection pooling was used anytime a connection was made to server especially during searches. Added a resource to content.xml and web.xml

* File name, line numbers as in Github

File: content.xml Lines: 12-16

File: web.xml Lines: 24-33

File: confirmation.jsp Lines 90-111

File: dashboard.jsp Lines: 78-97

File: main.jsp Lines: 92-114

File: movieListBrowse.jsp Lines: 184-203

File: movieListSearch.jsp Lines: 163-182

File singleMovie.jsp Lines: 82-97

File: singleStar Lines 79 -98

File AddMovie.java lines 53 – 72

File AddStar.java Lines 52- 71

File: AndroidLoginServlet.java Lines 50-69

File: AndroidSearch Lines 50-69

File: Login.java Lines 60-79

File: MovieServlet.java Lines 54-75

File: MovieSuggestion.java Lines 58-78

File: checkoutValidation.java Lines 56-76

File: employeeLogin.java Lines 66-87

* Snapshots showing use in your code

MovieSuggestion.java



Login.java



main.jsp



dashboard.jsp



* How did you use Prepared Statements?

Prepared statements were used for almost every query as they sped up searches and browsing because of the cached prepared statements.

* File name, line numbers as in Github

browseByGenre.jsp 83-85

confirmation.jsp 96-116

main.jsp 78 -82, 94-99, 104-109

movieListBrowse.jsp 189-194, 199-207, 217-229, 302-309, 321-333, 410-424, 435-446, 519-525, 535 -538, 542-550, 636 -651, 684 -686

movieListSearch.jsp 190-216, 281-305, 370-399, 475-503, 585 – 603

singleMovie.jsp 81-84, 95- 103

singleStar.jsp 78-93, 102-105

Login.java 70-76

checkoutValidation.java 67-106

employeeLogin.java 75-81

AddMovie.java 61-83

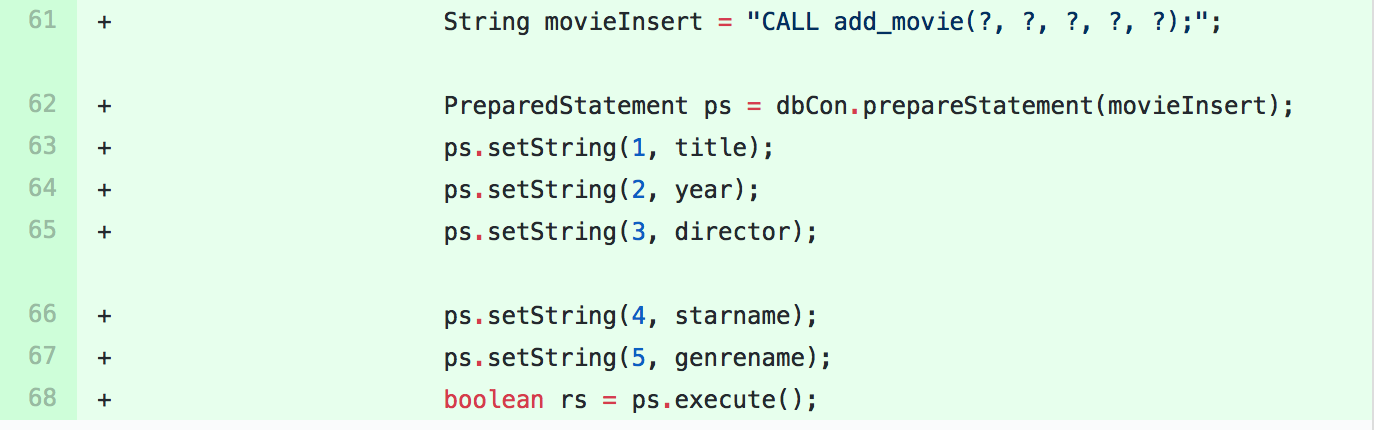
AddStar.java 58-70

AndroidLoginServlet.java 80-85

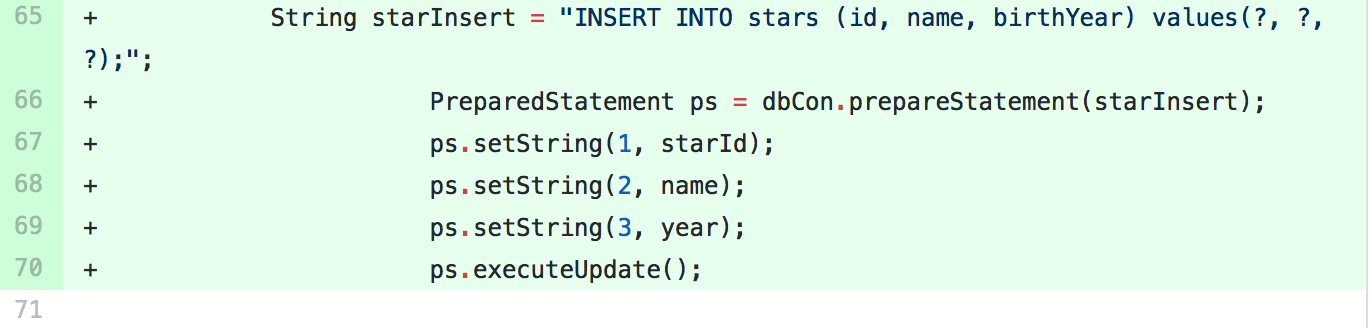
AndroidSearch.java 72-95

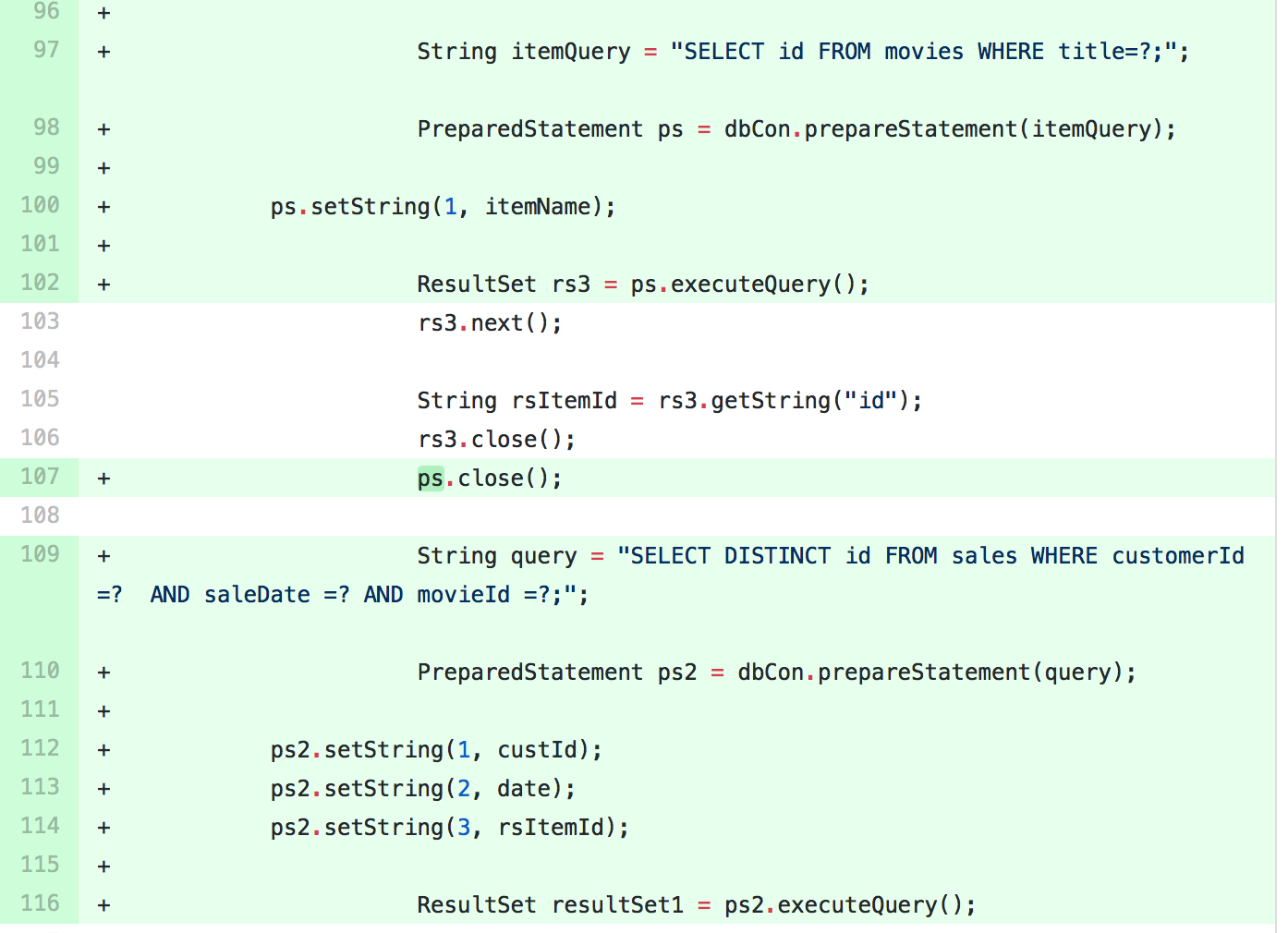
* Snapshots showing use in your code

AddMovie.java

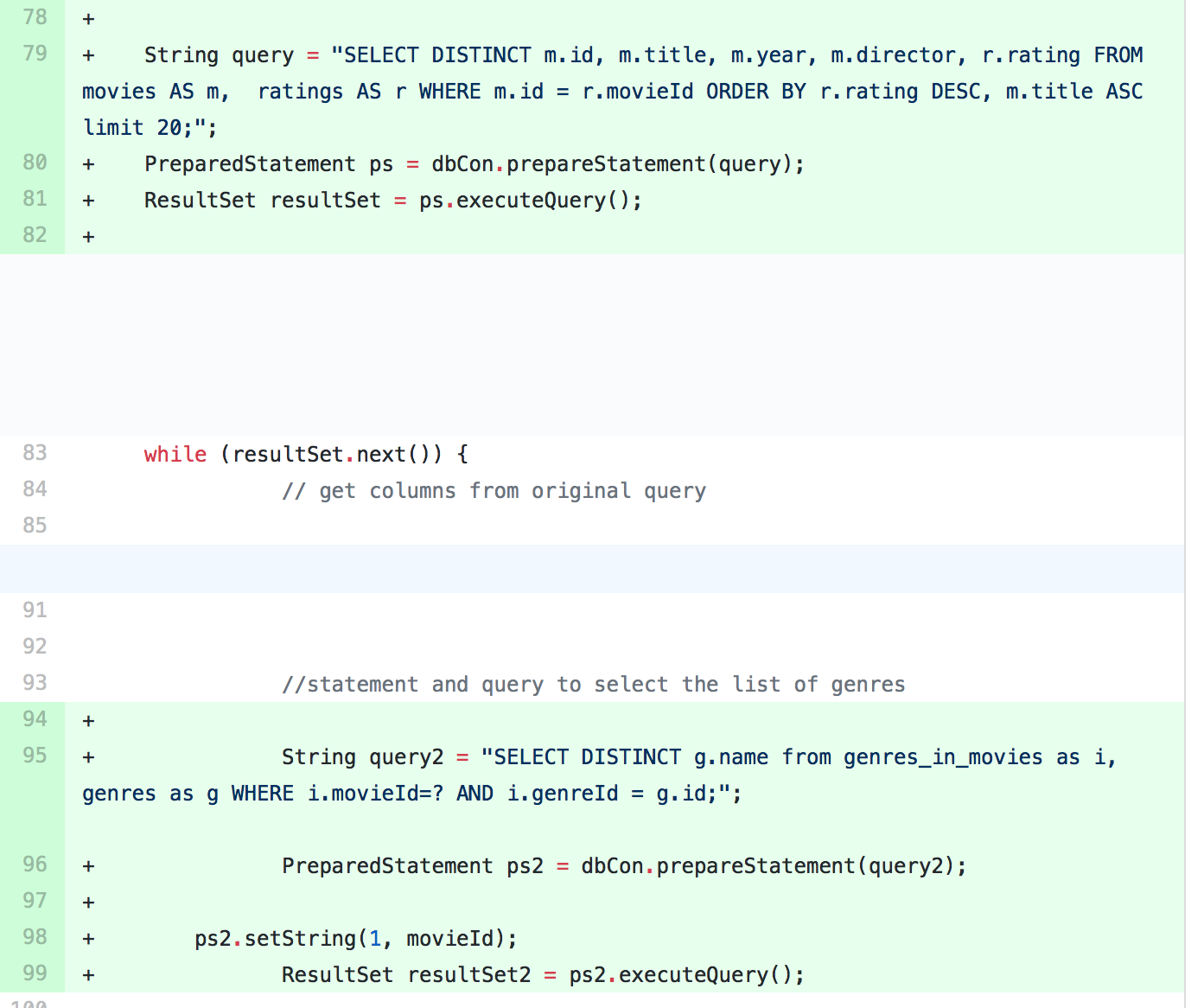


AddStar.java



confirmation.jsp

main.jsp



**Task 2**

* Address of AWS and Google instances

Google: 35.185.216.168

AWS #1: 18.216.78.253

AWS #2: 18.216.86.228

AWS #3: 18.191.128.82

http://IP:port/Fabflix/ (note the slash), for all four IP addresses

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

Yes. HTTPS was disabled for routing but still works on instance #1 for that particular instance.

* Explain how connection pooling works with two backend SQL (in your code)?

Connection pooling was used by the server similar as they were on the original except because of the way read and writes are done, most interactions with backend will stay with its current connection pool unless it is a write and it’s forced to go to the other server. This was also used because the load balancer picked the server to route each connection to.

* + File name, line numbers as in Github

File: content.xml Lines: 12-16

File: web.xml Lines: 24-33

File: AddMovie.java Lines: 73-88

File: AddStar.java Lines: 72-86

File: checkoutValidation.java Lines: 77-88

File: content.xml Lines: 12-16

File: web.xml Lines: 24-33

File: confirmation.jsp Lines 90-111

File: dashboard.jsp Lines: 78-97

File: main.jsp Lines: 92-114

File: movieListBrowse.jsp Lines: 184-203

File: movieListSearch.jsp Lines: 163-182

File singleMovie.jsp Lines: 82-97

File: singleStar Lines 79 -98

File AddMovie.java lines 53 – 72

File AddStar.java Lines 52- 71

File: AndroidLoginServlet.java Lines 50-69

File: AndroidSearch Lines 50-69

File: Login.java Lines 60-79

File: MovieServlet.java Lines 54-75

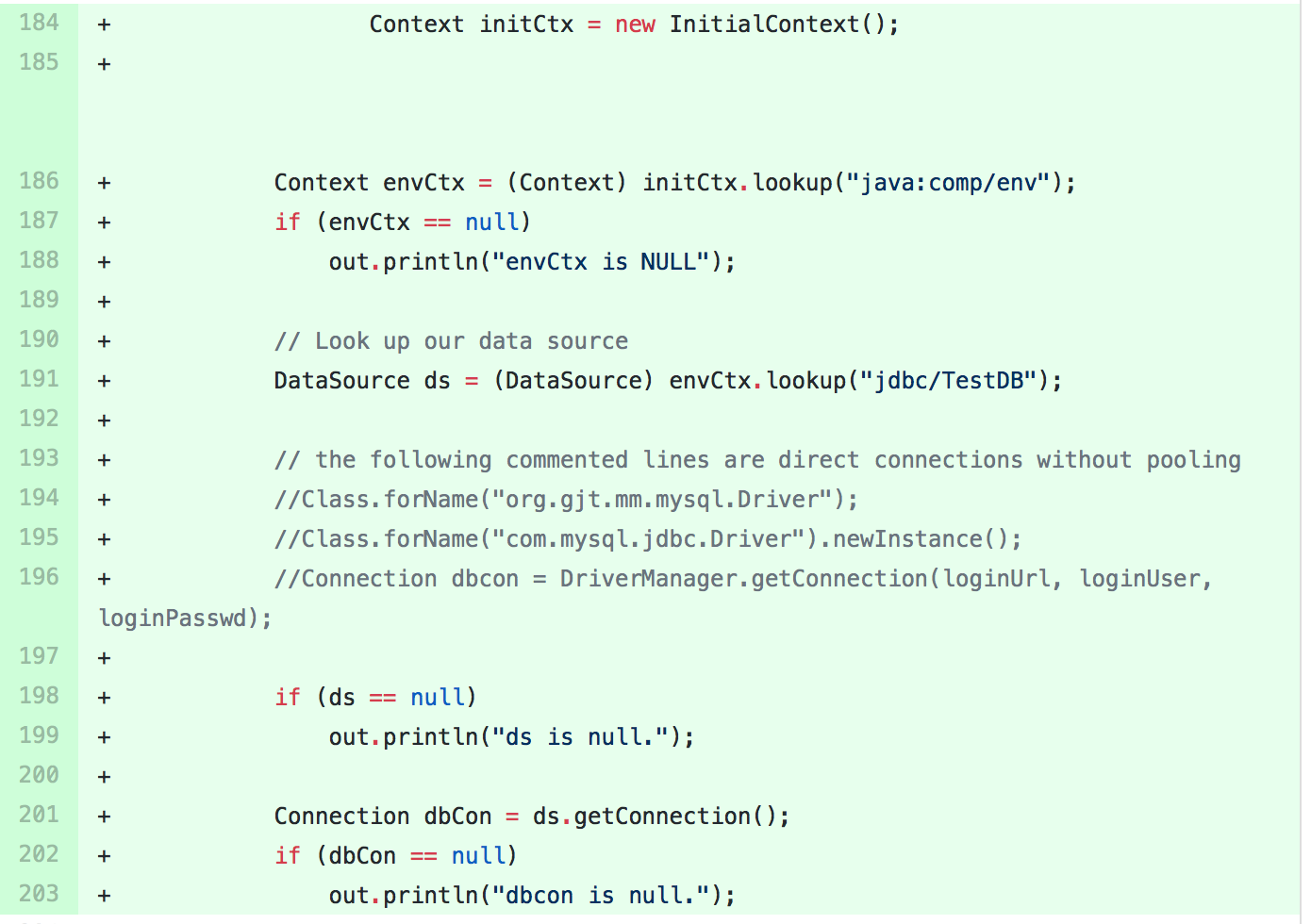
File: MovieSuggestion.java Lines 58-78

File: checkoutValidation.java Lines 56-76

File: employeeLogin.java Lines 66-87

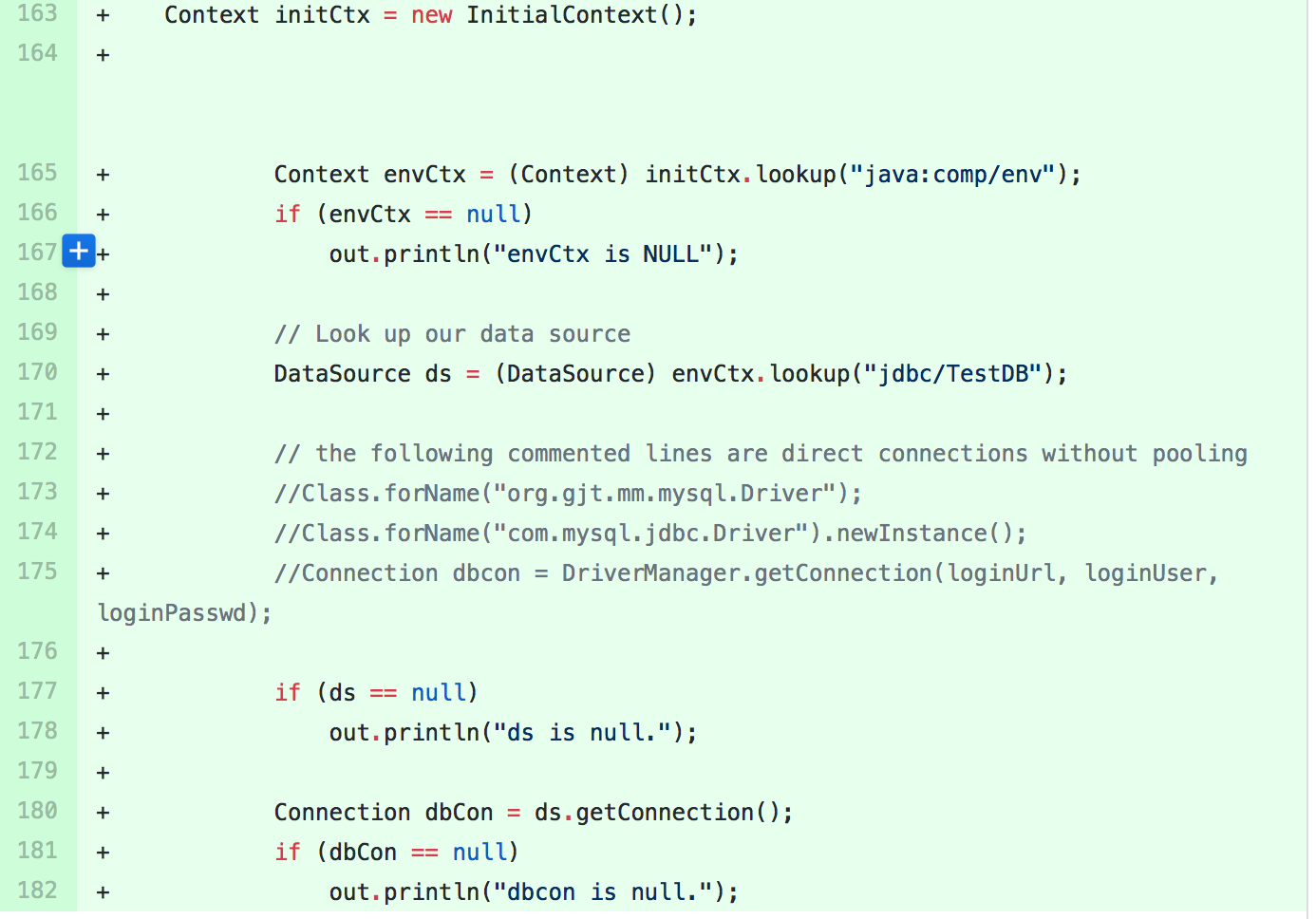
* + Snapshots

movieListBrowse.jsp



confirmation.jsp

movieListSearch.jsp



* How read/write requests were routed?

Added a new resource to content.xml and web.xml. Reads are routed to the server that the client is routed to while all writes were then routed to the master backend. This was done by creating a second resource that pointed strictly at the back end. It didn’t matter which server the client was routed to, all writes go to the master

* + File name, line numbers as in Github

File: content.xml Lines: 17-21

File: web.xml Lines: 34-44

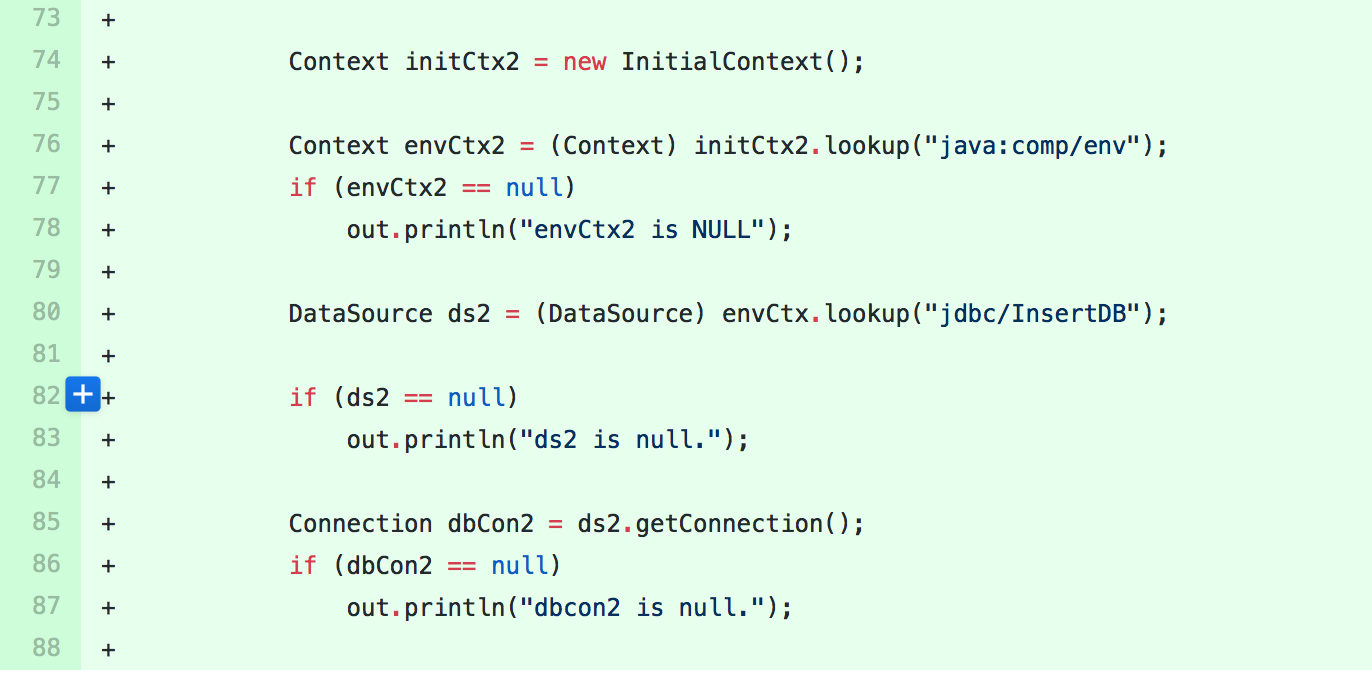
File: AddMovie.java Lines: 73-88

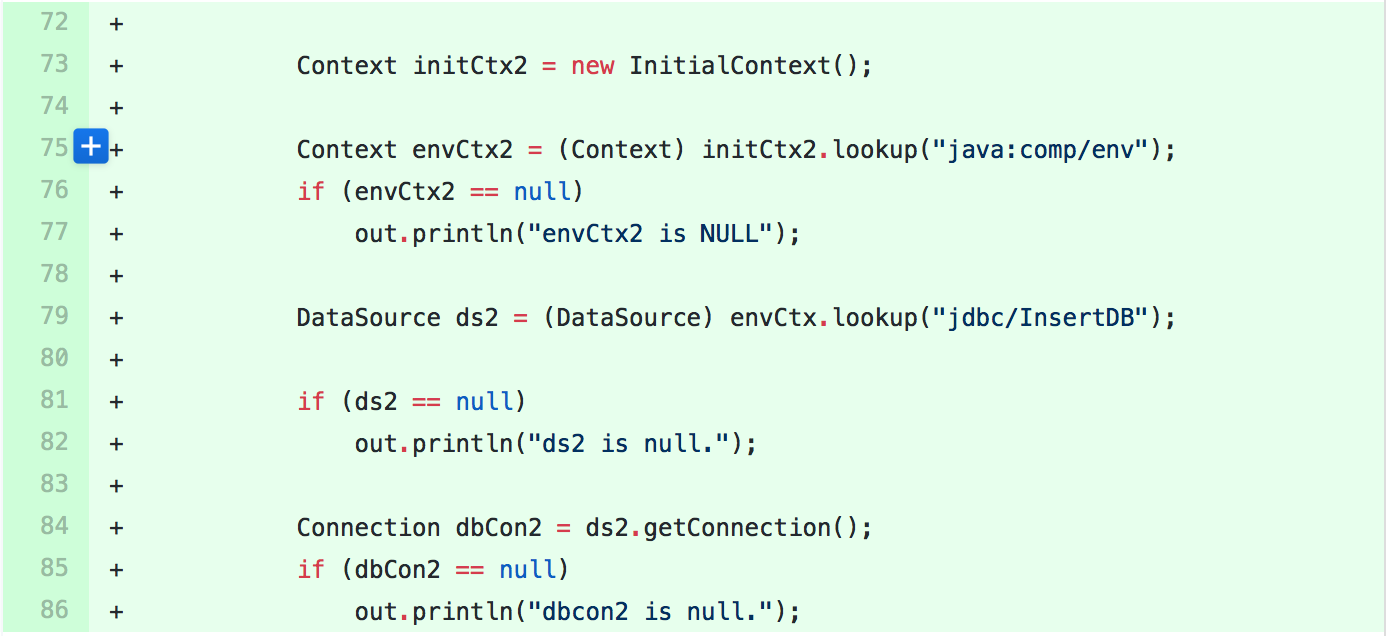
File: AddStar.java Lines: 72-86

File: checkoutValidation.java Lines: 77-90

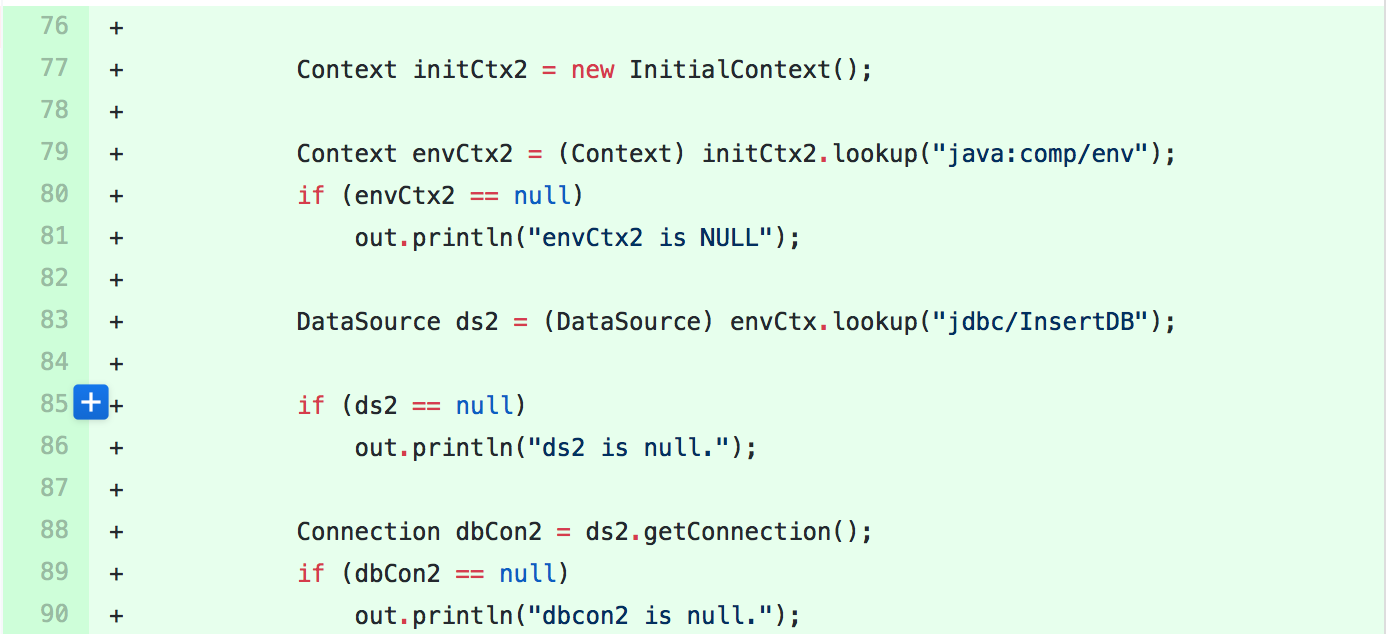
* Snapshots

AddMovie.java



AddStar.java

checkoutValidation.java



**Task 3**

* Have you uploaded the log files to Github? Where is it located?
* Have you uploaded the HTML file (with all sections including analysis, written up) to Github? Where is it located?
* Have you uploaded the script  to Github? Where is it located?

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