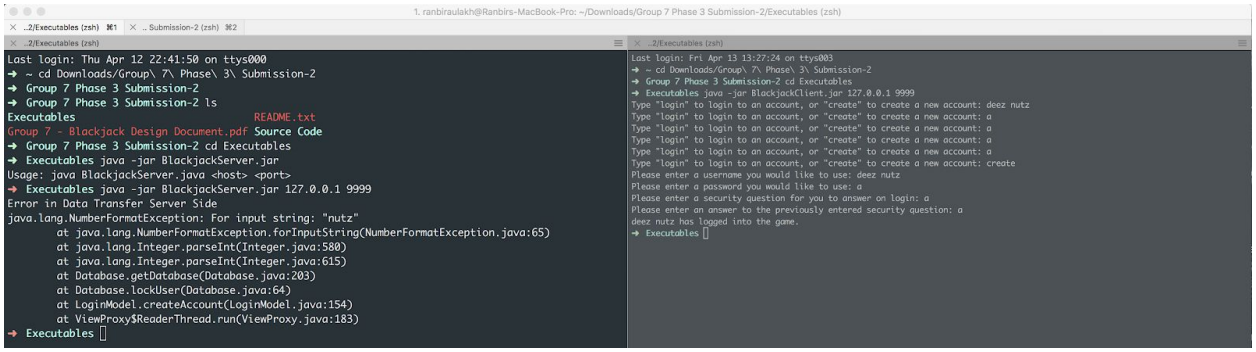
Ranbir Aulakh, Kemoy Campbell, Jason Durek - Investigated by Group 6

Group 7 Investigations

Setting up the Blackjack projects was not difficult. Below are the list of problems we discovered when using your software.

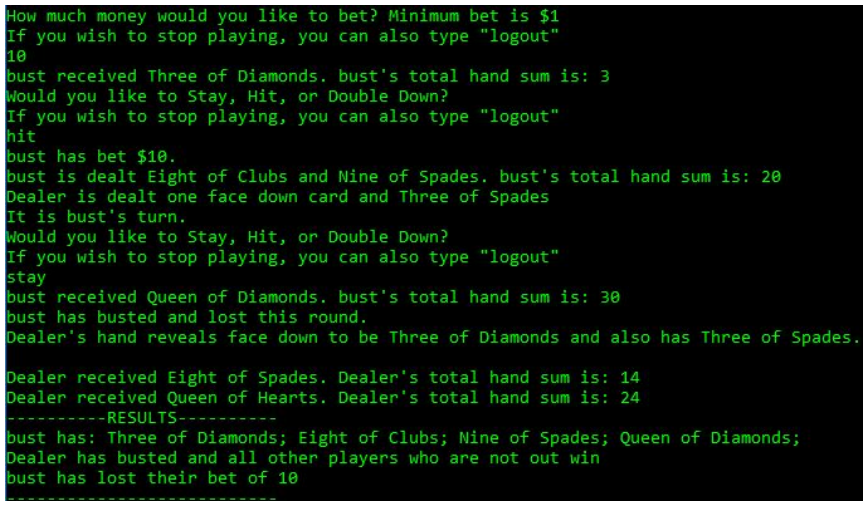
1. Logging in with a “space” in the username crashes the program. To elaborate, it crashes upon trying to read the username from the ‘database.txt’, as their method of reading from it uses spaces to split elements in the ‘table’. This means that the rest of the name falls under a different column, which expects integers only. We haven’t tested passwords or security questions for the space issue, but we suspect the same issue would arise.

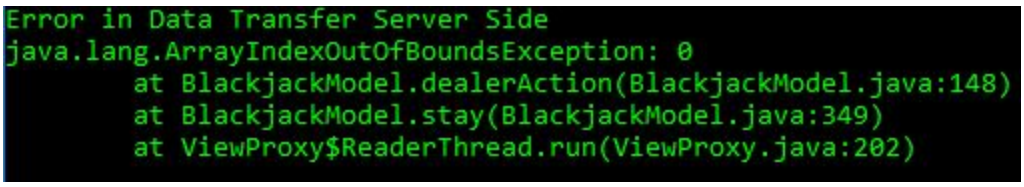


2. Everything is stored in a text file, instead of a proper database; Changing to a proper database would fix a large number of the issues with storage and retrieval of information.
3. Creating an account with existing username (ie: john) and paste “790976699 -1360666020 a 3104 100 false” as password then “space” for security question and security answer. Then it will say username exist. Then again with different username (ie: doe) and repeat the same process as above, it will pass and logged me in as john.
    a. Creating a new account with an existing username seems to result in the system thinking that user is logged in when it rejects the username application, locking that user out remotely.
    b. A countermeasure for this would be to first check file corruption prior to progress with the login in function. In other word, if the file exist, first check the corruption of the file.
4. After monitoring the database.txt file, it can be seen that the 3rd element is the hash that is used to determine whether the file is corrupted or not. Let us say if I win a 300 in total, I can just copy the entirely file content, store it somewhere else, if I get broke, I just copy the old content I will start at 300 again without having to start from 10
    a. Also using the fact that only the 3rd row is changing, it didn’t take long to figure out that the hashcode is combination of concating the username + passcode + money, with that being said, I can create my own username + passcode + 1M dollar and post those new changes to the file with correct hash hence it will read

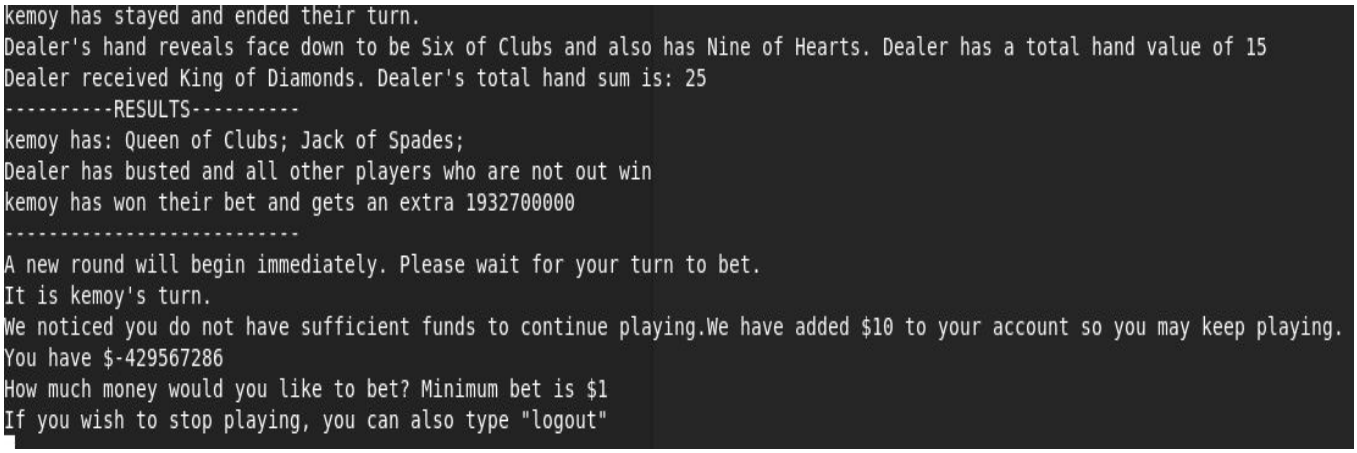
as valid file instead of a corrupt one and I will be starting the game at 1 million dollars

5. I successfully logged out an account. But logging back in the same account creates an error “User r has been logged in on another client. If you wish to log in on this client”.
6. Double Down is extremely glitchy, and not very indicative of how it works. Some results of double-down issues include having a hand that should be 16 reporting as 35 (Added result of previous game to current one), Causing only 1 card to be dealt before the start of a round, Crashing the central server, the dealer (Unsure what condition is, seems to be if the dealer busts under certain conditions)
    a. One card being dealt, Hit causes the match to act like it started, with the 3rd card already in play, Stay adds another card, causing player to bust. Dealer manages to bust as well, and ends up crashing. Player doesn’t crash until next input is provided





7. The addition of $10 if the user reaches $0 for testing doesn’t check to see if the value is negative. This isn’t too bad of an issue, since reaching negative values is normally impossible, but due to the integer overflow, it will just add $10 every time a user logs in until they reach $0 or better. With the overflow, this can result in a user being locked out due to the sheer number of logins needed.
    a. Current method to trigger the overflow: Login as non-corrupted user, then modify the database.txt file’s money column for this user to the max integer value for Java (2147483647). Save the changes to the database file, then attempt to win a round. The resulting gain of money from winning that round will result in an overflow.



8. In addMoney in Database.java there is this line int corruptCheck = (user.getUsername() + user.getPassCode() + user.getMoney()).hashCode(); which does not seem to be used.
9. It is possible to manipulate values in database.txt and bypass the corruption checks if the user is currently logged in. As far as we can tell, the corruption checks only trigger on user login, not during gameplay.