Owen McCall

Final Project Outline

Project Summary:

I created a small bank system for my final project using SQLite. The primary function is to “deposit” and “withdraw” money from a password-protected customer account. Money can also be transferred between customers with their userIDs. A history of transactions will be stored and can be viewed by customers upon request. User accounts will be of three types: customer, employee and administrator. Employee accounts will be able to create customer accounts for when a new customer signs up and can also reset those customers passwords upon request. Administrator accounts work like a higher level of account, but only administer employee accounts. This is because these accounts would likely be used for IT personnel who would not be given the duties of supporting customers but would be supporting the employee accounts.

Class Outlines:

1.

Class name: Login

Class variables: none

Class methods:

main(): initializes the bank system, asks for login information, starts one of the three types of account systems based on the account type.

2.

Class name: banksys

Class variables:

String user\_ID

Int user\_type

Long user\_balance

String user\_PIN

Boolean logged

Class methods:

Getters and setters for each variable

Get\_conn(): returns a connection object for working with the database

four\_digits(): returns a string of four digits from user input, adds zeros for full four digits

one\_digit():returns an integer from user input based on a controlled range

many\_digits():returns a long of a controlled number of digits

long\_to\_money(): converts a long of cents to a string of appropriately formatted dollars and cents

3.

Class name: User **extends** banksys

Class variables:

Class methods:

User(): constructor, gets info from banksys

home(): prints a home menu and prompts the user for choice of function, after the function if complete the customer will return here

check\_balance(): displays the customer’s balance

user\_deposit(): deposits a custom amount into customer account

user\_withdrawal(): withdraws a custom amount from the customer account

user\_transfer(): transfers a custom amount to a specified customer account

user\_history():prints an interactive history of the customers interactions

reset\_PIN(): resets customer’s PIN to specified four-digit string

4.

Class name: Employee **extends** banksys

Class variables:

Class methods:

Employee(): constructor, uses info from banksys

home(): prints a home menu and prompts the user for choice of function, after the function is completed the employee with return here

create\_user(): creates a user account of specified IDwith supporting database content

reset\_userPIN(): reset a specified customers PIN to “0000”

reset\_PIN(): resets PIN to specified four-digit string

5.

Class name: Admin **extends** banksys

Class variables:

Class methods:

Admin(): constructor, uses info from banksys

home(): prints home menu of functions, asks for number input, after the choie is completed admin will return here

create\_employee(): creates an employee account with specified userID and supporting database content

reset\_employeePIN(): resets a specified employee PIN to 0000

reset\_PIN(): resets PIN to specified four-digit string

Test Plan:

Before starting, you’re going to have to change the path used in banksys.get\_conn(), its line 48. I had to place my database inside my SQLite directory for some reason, so it’s not simply just the name of the database like other example of java and SQLite.

Testing requires software for viewing SQLite databases (.db files), I used DB Browser.

1. Run java login
2. Attempt to enter “word”, a number below -1, and a number above 9999.
3. Enter 0001
4. Attempt to enter any four-digit number but 0000
5. Enter 0000
6. Enter 2
7. Enter 0002
8. Enter 3
9. Attempt to enter any four-digit number but 0000
10. Enter 0000
11. Enter 1234
12. Enter any four-digit number but 1234
13. Enter 1234
14. Verify the message “Your PIN has been set to 1234.” was printed.
15. Enter 3
16. Enter -1
17. Verify no message was printed
18. Enter 3
19. Enter 1234
20. Enter -1
21. Enter 1234
22. Enter 0000
23. Enter -1
24. Enter 0000 twice
25. Verify “Your PIN has been set to 0000.” was printed
26. Enter 4
27. Enter 0002
28. Enter 0000
29. Enter 3
30. Enter 0000, 1234 then 1234
31. Verify “Your PIN has been set to 1234.” was printed
32. Enter 4
33. Enter 0001 and 0000
34. Enter 2
35. Enter 0002
36. Verify “Employee 0002 has had their PIN set to 0000.” was printed
37. Enter 2
38. Enter 9999
39. Verify “Error: ID could not be found, select another or enter -1 to go back.” was printed
40. Enter 2
41. Enter -1
42. Enter 1
43. Enter -1
44. Enter 4
45. Enter 0002 and 0000
46. Enter 1
47. Enter -1
48. Enter 1
49. Attempt to enter 0002
50. Enter 0003
51. Enter 6
52. Enter 0000, 1234, 1234
53. Enter 7
54. Enter 0003, 1234
55. Enter 7
56. Enter 0002, 0000
57. Enter 2
58. Enter 0003
59. Enter 4
60. Enter 0003, 0000
61. Enter 2
62. Attempt to enter “word” and 999999999999
63. Enter -1
64. Enter 2, 100000000
65. Verify “$1000000.00 has been deposited into your account.” was printed
66. Enter 3, -1
67. Enter 3, 2000000
68. Verify “$20000.00 has been withdrawn from your account.” was printed
69. Enter 1
70. Verify “Your Balance is: $980000.00” was printed
71. Enter 4, -1
72. Enter 4
73. Attempt to enter 0003
74. Attempt to enter 0002
75. Enter 0004, 8000000
76. Verify “$80000.00 has been transferred to 0004.” was printed
77. Enter 2, 10 Eight times
78. Enter 5, Verify five deposits of $.10 appear
79. Enter 2, verify three deposits of $.10 appear, followed by a transfer to 0004 of -$80000.00 and lastly a withdrawal of $20000.00
80. Enter 3, verify a single deposit of $1000000.00
81. Attempt to enter “word”, 4 and -2
82. Enter -1
83. Enter 6, 0000, 1234, 1234
84. Enter 7
85. Enter -1
86. Open bankdb.db, navigate to table “Users2”, verify it appears as below

|  |  |  |  |
| --- | --- | --- | --- |
| **userID** | **PIN** | **type** | **balance** |
| 0001 | 0000 | 3 | 0 |
| 0002 | 0000 | 2 | 0 |
| 0003 | 1234 | 1 | 90000080 |
| 0004 | 0000 | 1 | 800000 |

1. Navigate to table “0003”, verify it appears as below

|  |  |  |
| --- | --- | --- |
| **thirdparty** | **amount** | **type** |
| ATM | 100000000 | 0 |
| ATM | 2000000 | 1 |
| 0004 | 8000000 | 1 |
| ATM | 10 | 0 |
| ATM | 10 | 0 |
| ATM | 10 | 0 |
| ATM | 10 | 0 |
| ATM | 10 | 0 |
| ATM | 10 | 0 |
| ATM | 10 | 0 |
| ATM | 10 | 0 |

1. Navigate to table “0004”, verify it looks like below

|  |  |  |
| --- | --- | --- |
| **thirdparty** | **amount** | **type** |
| 0003 | 8000000 | 0 |