**Bank account Functional Test Doc**

1. **Equivalence partitioning**

|  |  |  |  |
| --- | --- | --- | --- |
| Partition ID | Input variable | Valid partition | Invalid partition |
| 1 | Card number | Length=16 | Length>16 |
| 2 | choice | 1<choice<4 | Choice =5 |
| 3 | CVV | Length=3 | Length>3 |
| 4 | Amount to withdraw | 1<amount<amount in bank | Amount>amount available |
| 5 | Name | Name=string | Name=number |
| 6 | Initial balance | Balance=number | Negative balance |
| 7 | Balance | Balance =number | Balance= string |

1. **Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Test inputs | Expected output | Partition ID covered |
| 1 | Card number | Card number=4532978826600708 | Length=16 |
| 2 | Name | Name= Max | Name=string |
| 3 | Initial balance | Balance=12.0 | Balance=number |
| 4 | CVV | CVV=763 | Length=3 |
| 5 | choice | Choice=3 | 1<choice<4 |
| 6 | balance | Balance= aaaa | Balance= string |
| 7 | Amount to withdraw | Amount to withdraw= 100 | 1<Amount to withdraw<amount in bank |
|  |  |  |  |
|  |  |  |  |

1. **Not able to test using Junit because User input is needed from the console… and that is not possible with unit testing.**

Manual testing with user input

A screenshot of a computer

Description automatically generated