Test plan - savings app

Requirements:

\* view a balance for a user

\* view the balances for all users

\* add funds for a user

\* remove funds for a user (funds must not go below 0)

\* add a new user with a balance of 0

\* remove a user

\* exit the app

Complete the table to plan how you will test each of the requirements. Each will need one or more tests to ensure that all possible cases have been tested.

You should test with valid users, non-valid users, valid and non-valid data types, etc.

TRY TO FILL THIS TABLE IN FIRST, then run the app, create 10 users with various balances and modify the data column to match the data you have in the app.

REMEMBER - the objective is to get the app to fail (it should do this quite easily if you have tested it fully).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test no.** | **What is being tested?** | **Data (user\_id, amount) used?** | **Expected outcome** | **Actual outcome** | **Notes** |
| 1 | Views all records in the savings table  get\_data(conn) | get\_data(db\_conn) | user\_id balance  --------- ---------  1 44  2 5  3 42  4 32  5 25  6 5  7 26  8 39  9 45  10 0 | user\_id balance  --------- ---------  1 44  2 5  3 42  4 32  5 25  6 5  7 26  8 39  9 45  10 0 | This test works |
| 2 | View the balance for a valid user  get\_user\_data(user\_id, conn) | get\_user\_data(1, db\_conn) | 44 | 44.0 | This test works |
| 3 | View the balance for a non-valid user  get\_user\_data(user\_id, conn) | get\_user\_data(15, db\_conn) | User does not exist | 44.0 | This test works |
| 4 | View the balance with an incorrect data type – one-digit string  get\_user\_data(user\_id, conn) | get\_user\_data(“1”, db\_conn) | Invalid user\_id data type |  | This test doesn’t work as the string data type is accepted. Need to add data type validation |
| 5 | Add new user with id that already exists and correct data type - integer add\_new\_user(user\_id, conn) | add\_new\_user(1, db\_conn) | User already exists |  | Need to add validation of user\_id to check if already exists |
| 6 | Add new user with id that does not exist and correct data type - integer  add\_new\_user(user\_id, conn) | add\_new\_user(11, db\_conn) | 1) “User 11 record added”  2) There should be a new entry in the sqlite3 database  3) The balance should be 0 | “User 11 record added”  True | This test works |
| 7 | Add new user with id that already exists and incorrect data type – one-digit string  add\_new\_user(user\_id, conn) | add\_new\_user(”1”, db\_conn) | 1) “Invalid user\_id data type”  2) no change in the database – a record was not added |  | This test doesn’t work as the invalid data type – string is accepted, even though unique constraint failed. Need to add data type validation |
| 8 | Add new user with id that does not exist and incorrect data type – two-digit string  add\_new\_user(user\_id, conn) | add\_new\_user(“12”, db\_conn) | 1) “Invalid user\_id data type”  2) no change in the database – a record was not added |  | This test doesn’t work as the invalid data type – string is accepted and a record was added to the database. Need to add data type validation |
| 9 | Add new user with id that already exists and incorrect data type – float  add\_new\_user(user\_id, conn) | add\_new\_user(1.0, db\_conn) | 1) “Invalid user\_id data type”  2) no change in the database – a record was not added |  | This test doesn’t work as the invalid data type – float is accepted, even though unique constraint failed. Need to add data type validation |
| 10 | Add new user with id that does not exist and incorrect data type – float  add\_new\_user(user\_id, conn) | add\_new\_user(13.0, db\_conn) | 1) “Invalid user\_id data type”  2) no change in the database – a record was not added |  | This test doesn’t work as the invalid data type – float is accepted and a record was added to the database. Need to add data type validation |
| 11 | Add new user with id that does not exist and incorrect data type – non-digit string  add\_new\_user(user\_id, conn) | add\_new\_user(“p”, db\_conn) | 1) “Invalid user\_id data type”  2) no change in the database – a record was not added |  | This test doesn’t work as the invalid data type – non-digit string is accepted and a record was added to the database. Need to add data type validation |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |