Use Case: Read Entered Transactions

Main Success Scenario:

1. The user enters their database credentials
2. The user logs in
3. The user sees the list of entered transactions
4. The user clicks on a transaction
5. The user sees detailed information about chosen transaction

Alternative Scenarios:

1a. If the user has already logged in once, they can skip steps 1-2.

2a. If the user cannot log in into their db, they must reenter the credentials

3a. If the user does not see transactions, they must create at least one first

3b. Alternatively, if the user has opened any other tab or report, they must close it first to access the list of transactions

FROM NOW ON THE PRECONDITION IS THAT THE USER HAS SUCCESSFULLY LOGGED INTO DB

Use Case: Create Transaction

Main Success Scenario:

1. The user sees the list of entered transactions
2. The user opens Create Entry form
3. The user enters date, amount, category, if transaction is recurrent or not, account type and name, and description of the transaction
4. The user confirms creation of transaction.
5. The user ensures the transaction has been created, by reading the list of all transactions

Alternative Scenarios:

1a. If the list is empty, skip the step.

3a. The user cancels entry if they do not have necessary information.

4a. The user cancels entry if they change their mind.

Use Case: Update Transaction

Main Success Scenario:

1. The user sees the list of entered transactions
2. The user opens Update Entry form
3. The user enters transaction id (note: maybe description or name would be better?)
4. The user confirms their input
5. The system populates the fields with values of the transaction
6. The user changes values
7. The user confirms updating information
8. The user ensures the transaction has been updated, by reading the list of all transactions

Alternative Scenarios:

1a. If the list is empty, create the transaction first.

4a. The user cancels entry if they change their mind.

5a. If the system cannot find the transaction, the user repeats step 3 with correct information.

5b. If the system still cannot find the transaction, the user creates it first.

7a. The user cancels entry if they change their mind.

Use Case: Delete Transaction

Main Success Scenario:

1. The user sees the list of entered transactions
2. The user selects the transaction they want to delete
3. The user clicks delete button
4. The user confirms they want to delete the transaction (note: necessary?)
5. The user ensures the transaction has been deleted, by reading the list of all transactions

Alternative Scenarios:

1a. If the list is empty, create a transaction first.

4a. The user cancels deletion if they change their mind.

Use Case: Generate Report

Main Success Scenario:

1. The user proceeds to generate report tab

2. The user selects the report type they want to generate (e.g. General, Income, expenses, savings)

3. The user generates report

4. The user reads the report table

5. The user reads the report graphs

Alternative Scenarios:

2a. If the user does not know what they want, they should choose General Report

2b. The user can filter Transactions for report generation by date and/or category

4a, 5a. If the report is empty, the user must ensure they have entered at least one transaction.

OPEN ISSUES

1. What about budget? Set goals? How should we handle it? Should we handle it at all?
2. Should we implement search?

3. Should we implement settings?

4. Do we really need to CRUD categories, or only transactions for now?

5. Should we fill out the date field automatically?

6. What if the user is broke? Can we have negative balance?