Project Test Plan & ICD

CMSC 495 6380

6/18/2021

Group 5

Summer Smith, Sean Dwyer, Keith Combs

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Name** | **Description** |
| 6/18 | Summer | Created document with cover page, revision history table, and outline. |
| 6/21 | Sean | Added Authentication module related Test Cases and ICD |
| 6/21 | Keith | Added Transaction module test cases and ICD |
| 6/22 | Summer | Added Interest Accrual test cases and ICD. Combined formatting for test cases. |
| 6/22 | Keith | Adjusted formatting for transaction subsystem and misspelled methods |
| 6/22 | Summer, Sean , Keith | Final review |

**Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case #** | **Requirement Number** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/ Fail** |
| 1 | 1, 3 | User attempts to register with valid username, password, and email | Registration success message, database file updated with new record |  |  |
| 2 | 1, 3 | User attempts to register with invalid username | Registration username exists failure message |  |  |
| 3 | 1, 3 | User attempts to register with valid username, invalid password | Registration password complexity failure message |  |  |
| 4 | 1, 3 | User attempts to register with valid username and password, invalid email | Registration email exists failure message |  |  |
| 5 | 1, 2, 4 | User attempts to login with valid username and password | Login success message, log file success record added |  |  |
| 6 | 1, 2, 4 | User attempts to login with invalid username | Login username does not exist failure message, log file failure record added |  |  |
| 7 | 1, 2, 4 | User attempts to login with valid username, invalid password | Login password does not match failure message, log file failure record added |  |  |
| 8 | 1, 2 | Logged in user attempts to logout | Logout success message |  |  |
| 9 | 1, 2 | Non-logged in user attempts to logout | Cannot access webpage message |  |  |
| 10 | 5, 11 | Logged in user chooses checking and deposit. User is prompted for amount. User enters $20 dollars and $20 dollars is added to the balance. The user then is shown the balance of their checkings and savings. | Web page refreshes and balance of checkings and savings shows. $20 is added to checkings. |  |  |
| 11 | 5, 11 | Logged in user chooses savings and deposit. User is prompted for amount. User enters $20 dollars and $20 dollars is added to the balance. The user then is shown the balance of their checkings and savings. | Web page refreshes and balance of checkings and savings shows. $20 is added to savings. |  |  |
| 12 | 5, 11 | Logged in user chooses checkings and deposit, user enters “cat”. Then chooses savings and deposit and user enters “dog”. Then chooses savings and deposit and enters -100. | Web page catches wrong input and shows invalid input |  |  |
| 13 | 6, 11, 12, 13, 14 | Logged in user chooses checkings and withdrawal. Balance of 100 shows and user enters 105. The web page prompts to show there is an overdraft fee of $20. User selects yes, 120 is subtracted from the balance and a negative balance shows. | A negative balance shows after withdrawal is complete |  |  |
| 14 | 6, 11, 12, 13, 14 | Logged in user chooses savings and withdrawal. Balance of 100 shows and user enters 105. The web page prompts to show there is an overdraft fee of $20. User selects yes, 120 is subtracted from the balance and a negative balance shows. | A negative balance shows after withdrawal is complete |  |  |
| 15 | 6, 11 | Logged in user chooses checkings and withdrawal. Balance of 100 shows and user enters 20. User selects yes, 120 is subtracted from the balance and a positive balance shows. | A positive balance of 120 shows after withdrawal is complete |  |  |
| 16 | 6, 11 | Logged in user chooses checkings and withdrawal. Balance of 100 shows and user enters 20. User selects yes, 120 is subtracted from the balance and a positive balance shows. | A positive balance of 120 shows after withdrawal is complete |  |  |
| 17 | 7, 11 | Logged in user chooses checkings and then chooses show balance | Balance is shown for checking |  |  |
| 18 | 7, 11 | Logged in user chooses savings and then chooses show balance | Balance is shown for savings |  |  |
| 19 | 8 | There is no add account option | No add account options is on web page |  |  |
| 20 | 9, 11 | Logged in user selects transfer from checkings to savings. Balance of checkings and savings shows. User enters $20 to transfer from checkings to savings | $20 dollars is subtracted from checkings and added to savings. New balances are shown. |  |  |
| 21 | 9, 11 | Logged in user selects transfer from checkings to savings. Balance of checkings and savings shows. User enters $20 to transfer from checkings to savings | $20 dollars is subtracted from savings and added to checkings. New balances are shown |  |  |
| 22 | 9, 11, 12, 13, 14 | Logged in user selects transfer from checkings to savings. Balance of checkings and savings shows. User enters $20 more than balance from checkings to transfer from checkings to savings | Overdraft alert shows. User selects yes, withdrawal amount + 20 dollars is subtracted from checkings and added to savings |  |  |
| 23 | 10 | User logs out on February 1, 2021 and does not log back in till February 10, 2021. Balance of interest is updated to reflect interest earned over past 9 days. APY for account is 0.05%. | Savings is $100 on February 1 and interest is $0. On February 10, savings is $100, interest is $0.01. (0.005 / 365 \* 100 \* 9 = 0.012319) |  |  |
| 24 | 10 | User logs out on February 1, 2021 and logs back in later that same day on February 1, 2021. Balance of interest does not change because it is only updated once a day. | Savings is $100 and interest is $10 after first log in. User logs out and back in, balances do not change. |  |  |

**ICD**

Data between user and Registration Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Values | Default |
| Registration webpage username input form | Registration Subsystem | request.form(‘username’); NOTE: framework feature | String | Any String | N/A; NOTE: HTML form forces user input |
| Registration webpage password input form | Registration Subsystem | request.form(‘password’); NOTE: framework feature | String | Any String | N/A; NOTE: HTML form forces user input |
| Registration webpage email input form | Registration Subsystem | request.form(‘email’); NOTE: framework feature | String | Any String | N/A; NOTE: HTML form forces user input |
| Registration Subsystem | Registration webpage | flash(message); NOTE: framework feature | String | Username exists failure message | N/A |
| Registration Subsystem | Registration webpage | flash(message); NOTE: framework feature | String | Password complexity failure message | N/A |
| Registration Subsystem | Registration webpage | flash(message); NOTE: framework feature | String | Email exists failure message | N/A |
| Registration Subsystem | Registration webpage | flash(message); NOTE: framework feature | String | Registration success message | N/A |

Data between database and Registration Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Database file | Registration Subsystem | database.fetchall(SQL query); NOTE: Database API | List | String List | Empty if no results |
| Registration Subsystem | Database file | database.execute(SQL record statement); NOTE: Database API | String | SQL statement String | N/A |

Data between user and Logon Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Values | Default |
| Login webpage username input form | Logon Subsystem | request.form(‘username’); NOTE: framework feature | String | Any String | N/A; NOTE: HTML form forces user input |
| Login webpage username input form | Logon Subsystem | request.form(‘password’); NOTE: framework feature | String | Any String | N/A; NOTE: HTML form forces user input |
| Logon Subsystem | Login webpage | flash(message); NOTE: framework feature | String | Username does exist failure message | N/A |
| Logon Subsystem | Login Webpage | flash(message); NOTE: framework feature | String | Password does not match failure message | N/A |
| Logon Subsystem | Login Webpage | flash(message); NOTE: framework feature | String | Login success message | N/A |
| Logout Webpage | Logon Subsystem | HTML GET request; NOTE: framework implementation | N/A; NOTE: GET request alone is sufficient data | N/A | N/A |
| Logon Subsystem | Logout Webpage | flash(message); NOTE: framework feature | String | Logout success message | N/A |

Data between database and Logon Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Database file | Logon Subsystem | database.fetchall(SQL query); NOTE: Database API | List | String List of query results | Empty List (if no results) |
| Database file | Logon Subsystem | database.fetchone(SQL query); NOTE: Database API | Sequence | Single sequence of query results | None (if no results) |
| Logon Subsystem | Database | database.execute(SQL query); NOTE: Database API | Datetime | Datetime of login | Current datetime (YYYY-MM-DD HH:MM:SS) |

Data between Interest Accrual Subsystem and Logon Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Logon Subsystem | Interest Accrual Subsystem | session.get(‘user\_id’); NOTE: framework feature global | String | User ID string for authenticated user session | None (if no authenticated user session) |

Data between Interest Accrual Subsystem and database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Database | Interest Accrual Subsystem | database.fetchall(SQL query); NOTE: Database API | Datetime | Datetime of last login | Current datetime (YYYY-MM-DD HH:MM:SS) |
| Database | Interest Accrual Subsystem | database.fetchall(SQL query); NOTE: Database API | float | Balance of account’s interest | 0.0 |
| Interest Accrual Subsystem | Database | database.execute(SQL query); NOTE: Database API | float | New balance of account’s interest | 0.0 |

Data between log file and Logon Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Logon Subsystem | Log file | File.write(line to be written); NOTE: Python standard library File API | String | String details of login attempt | N/A |

Data between session global and Logon Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Logon Subsystem | Session global | session[‘user\_id’]; NOTE: framework feature global assignment | String | User ID string for authenticated user logged in | N/A |
| Logon Subsystem | Session global | Session.clear(); NOTE: framework feature | N/A; clear request alone is sufficient data | N/A | N/A |

Data between session global and Session Authentication Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Session global | Session Authentication Subsystem | session.get(‘user\_id’); NOTE: framework feature global | String | User ID string for authenticated user session | None (if no authenticated user session) |

Data between Transaction Subsystem and Session Authentication Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Session Authentication Subsystem | Transaction Subsystem | Session\_Authenication.verify() | String | User ID string for authenticated user session | None (if no authenticated user session) |

Data between User and Transaction Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Web Page | Transaction Subsystem | Transaction.chooseAcctType() | String | [“Checkings”, “Savings”] | “Checkings” |
| Web Page | Transaction Subsystem Show balance | Transaction.showBalance() | String,  integer | [“Checkings”, “Savings”], 7 digit | “Checkings”,  0000000 |
| Web Page | Transaction Subsystem Withdrawal | Transaction.withdrawal() | float | 0-9999 | 0 |
| Web Page | Transaction Subsystem Deposit | Transaction.deposit() | float | 0-9999 | 0 |
| Web Page | Transaction Subsystem Transfer | Transaction.transfer() | Float, String, String | 0-9999,  [“Checkings”, “Savings”], [“Checkings”, “Savings”] | 0, “Checkings”, “Savings” |

Data between Transaction Subsystem and database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Transaction Subsystem Show balance | Database | database.fetchall(SQL query); NOTE: Database API | String,  integer | [“Checkings”, “Savings”], 7 digit | “Checkings”,  0000000 |
| Transaction Subsystem | database | database.execute(SQL query); NOTE: Database API | float | 0-9999 | 0 |

Data between Transaction Subsystem and User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction Subsystem - Alerts | Web Page | flash(message); NOTE: framework feature | String | Withdrawal Amount over Balance | N/A |
| Transaction  Subsystem - Alerts | Web Page | flash(message); NOTE: framework feature | String | Transaction Failed | N/A |
| Transaction  Subsystem - Alerts | Web Page | flash(message); NOTE: framework feature | String | Confirm transaction | N/A |