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 |

**Test Cases**

* Make sure all functional reqs are covered
* Cover all normal operations and error-handling scenarios

*Test Case 1*

Description: User attempts to register with valid username, password, and email

Expected Results: Registration success message, database file updated with new record

Functional Requirement(s): 1, 3

*Test Case 2*

Description: User attempts to register with invalid username

Expected Results: Registration username exists failure message

Functional Requirement(s): 1, 3

*Test Case 3*

Description: User attempts to register with valid username, invalid password

Expected Results: Registration password complexity failure message

Functional Requirement(s): 1, 3

*Test Case 4*

Description: User attempts to register with valid username and password, invalid email

Expected Results: Registration email exists failure message

Functional Requirement(s): 1, 3

*Test Case 5*

Description: User attempts to login with valid username and password

Expected Results: Login success message, log file success record added

Functional Requirement(s): 1, 2, 4

*Test Case 6*

Description: User attempts to login with invalid username

Expected Results: Login username does not exist failure message, log file failure record added

Functional Requirement(s): 1, 2, 4

*Test Case 7*

Description: User attempts to login with valid username, invalid password

Expected Results: Login password does not match failure message, log file failure record added

Functional Requirement(s): 1, 2, 4

*Test Case 8*

Description: Logged in user attempts to logout

Expected Results: Logout success message

Functional Requirement(s): 1, 2

*Test Case 9*

Description: Non-logged in user attempts to logout

Expected Results: Cannot access webpage message

Functional Requirement(s): 1, 2

**ICD**

* Any 2 communicating classes must have an ICD
* Must list data types, range of values, and default values

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) |

Data between Interest Accrual Subsystem and Logon Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| From | To | Name | Type | Value | Default |
| Logon Subsystem | Interest Accrual Subsystem | ? | Datetime | Datetime of login | N/A |

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) |