JustHealth Test Cases

|  |  |
| --- | --- |
| **Iteration #:** |  |
| **Authored By:** |  |
| **Date Authored:** |  |
| **Version #:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Date**  (DD-MMM-YYYY) | **Signed** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Client Table: Inserting legitimate data into the client table

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.1.1 | 1. Run the SQL found in Appendix 1.1.1. 2. Run the following SQL on the database:   SELECT \* FROM client | There is a record in the database with username = test, email = [test@test.com](mailto:test@test.com) and verified = FALSE. |  |

# Client Table: Inserting data that is too long into the client table

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.2.1 | 1. Run the SQL found in Appendix 1.2.1. | The insert statement should fail to execute. |  |
| 1.2.2 | 1. Run the SQL found in Appendix 1.2.2. | The insert statement should fail to execute. |  |
| 1.2.3 | 1. Run the SQL found in Appendix 1.2.2. | The insert statement should fail to execute. |  |

# Client Table: Executing an insert command with data missing

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.3.1 | 1. Run the SQL found in Appendix 1.3.1. | The insert statement should fail to execute. |  |
| 1.3.2 | 1. Run the SQL found in Appendix 1.3.2. | The insert statement should fail to execute. |  |
| 1.3.3 | 1. Run the SQL found in Appendix 1.3.3. | The insert statement should fail to execute. |  |

# Client Table: Deleting records from the table

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.4.1 | 1. Run DELETE \* FROM client 2. Run the first SQL statement found in Appendix 1.4.1. 3. Run SELECT \* FROM client;   *you should see the record that you just added with the previous statement.*   1. Run the second SQL statement found in Appendix 1.4.1. 2. Run SELECT \* FROM client, password; | The record that was previously added should be deleted. There should be no records in the table. |  |
| 1.4.2 | 1. Run DELETE \* FROM client 2. Run the first SQL statement found in Appendix 1.4.2. 3. Run SELECT \* FROM client;   *you should see the record that you just added with the previous statement.*   1. Run the second SQL statement found in Appendix 1.4.2. 2. Run SELECT \* FROM client, password; | The record that was previously added should be deleted. There should be no records in the table. |  |

# Client Table: Update the data for a given record

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.5.1 | 1. Run DELETE \* FROM client 2. Run the first SQL statement found in Appendix 1.5.1. 3. Run SELECT \* FROM client;   *you should see the record that you just added with the previous statement.*   1. Run the second SQL statement found in Appendix 1.5.1. 2. Run SELECT \* FROM client; | The username field should have updated and now display ‘testingUpdate’. |  |
| 1.5.2 | 1. Run DELETE \* FROM client 2. Run the first SQL statement found in Appendix 1.5.2. 3. Run SELECT \* FROM client;   *you should see the record that you just added with the previous statement.*   1. Run the second SQL statement found in Appendix 1.5.2. 2. Run SELECT \* FROM client; | The email field should have updated and now display ‘testingUpdate@testingUpdate.com’. |  |

# Client Table: Update the data for a given record (Cont’d)

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.5.3 | 1. Run DELETE \* FROM client 2. Run the first SQL statement found in Appendix 1.5.3. 3. Run SELECT \* FROM client;   *you should see the record that you just added with the previous statement. Verified should be set to FALSE.*   1. Run the second SQL statement found in Appendix 1.5.3. 2. Run SELECT \* FROM client; | The verified field should have updated to TRUE. |  |
| 1.5.4 | 1. Run DELETE \* FROM client 2. Run the first SQL statement found in Appendix 1.5.4. 3. Run SELECT \* FROM client;   *you should see the record that you just added with the previous statement. Verified should be set to FALSE.*   1. Run the second SQL statement found in Appendix 1.5.4. 2. Run SELECT \* FROM client; | The accountLocked field should have updated to TRUE. |  |

# Password Table: Inserting legitimate data into the password table

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.6.1 | 1. Run the two SQL statement found in Appendix 1.6.1. 2. Run SELECT \* FROM uq8LnAWi7D | The record that Appendix 1.6.1 inserts should be seen in the table. |  |

# Password Table: Inserting illegitimate data into the password table

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.7.1 | 1. DELETE \* FROM uq8LnAWi7D 2. Run the SQL statement found in Appendix 1.7.1. | The SQL statement should not execute successfully because no username has been provided. |  |
| 1.7.2 | 1. Run the SQL statement found in Appendix 1.7.2. | The SQL statement should not execute successfully as the user ‘notInTable’ does not exist in the client table. |  |
| 1.7.3 | 1. Run the SQL statement found in Appendix 1.7.3. | The SQL statement should not execute successfully because the password field is too long. |  |
| 1.7.4 | 1. Run the SQL statement found in Appendix 1.7.4. | The SQL statement should not execute successfully because the isCurrent value passed is not a legitimate Boolean value. |  |
| 1.7.5 | 1. Run the SQL statement found in Appendix 1.7.5. | The SQL statement should not execute successfully because the expiryDate is not valid. |  |

# Password Table: Deleting records from the password table

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.8.1 | 1. Run the first SQL statement found in Appendix 1.8.1. 2. Run SELECT \* FROM uq8LnAWi7D;   *You should see the record that you just added with the previous statement.*   1. Run the second SQL statement found in Appendix 1.8.1. 2. Run SELECT \* FROM uq8LnAWi7D; | The record that was previously added should be deleted. There should be no records in the table. |  |

# Password Table: Testing the Constraint on Foreign key

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Database |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
| 1.9.1 | 1. Run DELETE \* FROM uq8LnAWi7D; 2. Run DELETE \* FROM client; 3. Run the first SQL statements found in Appendix 1.9.1. 4. Run SELECT \* FROM client;   *You should see the record that you just added with the previous statement.*   1. Run SELECT \* FROM uq8LnAWi7D;   *You should see the record that you just added with the previous statement.*   1. Run the second SQL statement found in Appendix 1.9.1. 2. Run SELECT \* FROM client; 3. Run SELECT \* FROM uq8LnAWi7D; | The records in both the client and the uq8LnAWi7D should have been removed. |  |

# Web Application: User Registration

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Web |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
|  | 1. Navigate to the registration screen. 2. Fill in the username with ‘registrationTest’ 3. Fill in the email address with *tester’s email address* 4. Fill in the two password fields with ‘registrationPassword’ 5. Press Submit. 6. Connect to the database, run SELECT \* FROM client;   **Take a screenshot**   1. Connect to the database, run SELECT \* FROM uq8LnAWi7D;   **Take a screenshot** | The SELECT \* FROM client query should show the record created in the client table with the following attributes:   * Username = registrationTest * Email = *tester’s email address* * Verified = false   The SELECT \* FROM uq8LnAWi7D query should show the record created in the ‘password’ table with the following attributes:   * Username = registrationTest * Password = SHA value * isCurrent = True * expiryDate = 45 days from day of test |  |

# Web Application: User Registration (Cont’d)

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Web |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
|  | 1. Check your email to check for the verification email from JustHealth.   **Take a screenshot.** | Email to have been received asking for email verification after your recent registration. |  |
|  | 1. Click the link in the email. | The link takes you to a page to verify your email address. |  |
|  | 1. Connect to the database and run, SELECT \* FROM client   **Take a screenshot.** | The verified field has switched to TRUE. |  |

# Web Application: Log In Functionality

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Web |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
|  | 1. Navigate to the login screen. 2. Use the username: ‘registrationTest’ and password: ‘registrationPassword’. 3. Press login. | The login is authorised. |  |
|  | 1. Navigate to the login screen. 2. Use the username: ‘registrationTest’ and password: ‘registrationPasswords’. 3. Press login. | The login is not authorised – incorrect password. |  |
|  | 1. Navigate to the login screen. 2. Use the username: ‘registrationTest’ and password: ‘registrationPasswords’. 3. Press Login. 4. Attempt this another 4 times.   **Take a screenshot.** | A message should appear stating that the account is locked. |  |

# Web Application: Log In Functionality (Cont’d)

|  |  |
| --- | --- |
| **Iteration #:** | 1 |
| **Application Type:** | Web |
| **Date/Time**  **(DD-MMM-YYYY / HH:MM):** |  |
| **Tester Name:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Steps** | **Expected Result** | **Pass/Fail (Initials)** |
|  | 1. Check your email to find the email informing you of your account being locked. | Email to have been received informing you that your account is locked and containing a link to reset your password. |  |
|  | 1. Click the link in the email from JustHealth that is informing you that your account is locked. | Directs you to a site that allows you to reset your password. |  |

# Appendix

## 

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

## 1.2.1.

INSERT INTO client (username, email)

VALUES (‘testtesttesttesttesttestte’, ’test@test.com’);

INSERT INTO client (username, email)

VALUES (‘test’, ’testesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttestt@test.com’);

INSERT INTO client (username, email)

VALUES (‘testtesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttestt’, ’testesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttesttestt@test.com’);

INSERT INTO client (username, email)

VALUES (‘’, ’test@test.com’);

INSERT INTO client (username, email)

VALUES (‘test’, ‘’);

INSERT INTO client (username, email)

VALUES (‘’, ‘’);

SQL statement one

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

SQL statement two

DELETE FROM client

WHERE (username = ‘test’);

SQL statement one

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

SQL statement two

DELETE FROM client

WHERE (email = ‘test@test.com’);

SQL statement one

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

SQL statement two

UPDATE client

SET username = ‘testingUpdate’

WHERE username = ’test’;

SQL statement one

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

SQL statement two

UPDATE client

SET email = ‘testingUpdate@testingUpdate.com’

WHERE username = ’test’;

## 

SQL statement one

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

SQL statement two

UPDATE client

SET verified = TRUE

WHERE username = ’test’;

## 1.5.4.

SQL statement one

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

SQL statement two

UPDATE client

SET accountLocked = TRUE

WHERE username = ’test’;

## 1.6.1.

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

INSERT INTO uq8LnAWi7D (username, password, isCurrent, expiryDate)

VALUES (‘test’, crypt(‘password’, gen\_salt(‘md5’)), TRUE, 10/10/2014);

INSERT INTO uq8LnAWi7D (password, isCurrent, expiryDate)

VALUES (crypt(‘password’, gen\_salt(‘md5’)), TRUE, 10/10/2014);

INSERT INTO uq8LnAWi7D (username, password, isCurrent, expiryDate)

VALUES (‘notInTable’, crypt(‘password’, gen\_salt(‘md5’)), TRUE, 10/10/2014);

INSERT INTO uq8LnAWi7D (username, password, isCurrent, expiryDate)

VALUES (‘test’, crypt(‘passwordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpasswordpassword’, gen\_salt(‘md5’)), TRUE, 10/10/2014);

INSERT INTO uq8LnAWi7D (username, password, isCurrent, expiryDate)

VALUES (test, crypt(‘password’, gen\_salt(‘md5’)), ‘test’, 10/10/2014);

INSERT INTO uq8LnAWi7D (username, password, isCurrent, expiryDate)

VALUES (test, crypt(‘password’, gen\_salt(‘md5’)), TRUE, 19/17/1993);

SQL statement one

INSERT INTO uq8LnAWi7D (username, password, isCurrent, expiryDate)

VALUES (‘test’, crypt(‘password’, gen\_salt(‘md5’)), TRUE, 10/10/2014);

SQL statement two

DELETE FROM uq8LnAWi7D

WHERE (username = ‘test’);

SQL statement one

INSERT INTO client (username, email)

VALUES (‘test’, ’test@test.com’);

INSERT INTO uq8LnAWi7D (username, password, isCurrent, expiryDate)

VALUES (‘test’, crypt(‘password’, gen\_salt(‘md5’)), TRUE, 10/10/2014);

SQL statement two

DELETE FROM client

WHERE (username = ‘test’);