Mobile Storage Analytics

Test Plan

Team CodeStars

Project Members

• Hisham Alhussain • Matthew LeBlanc

• Joy Liu • Yueying Liu

• Paul Maurais • Aisiri Murulidhar

• Siddharth Naidu • Varun Sharma

• Andrew Sharp • Quyen Tran

• Cody Dyl (manager)

## 

## Table of Contents

[Test Case ID: #1 (Login Functionality)](#_wsutaz8tn4a3) 3

[Test Case ID: #2 (Aggregated Info For Users)](#_hfpmfqh5k87s) 4

[Test Case ID: #3 (Data Visualization)](#_rwyu0v7layc6) 5

[Test Case ID: #4 (Filter)](#_t6gfkbk8z2z8) 6

[Test Case ID: #5 (Storage List View)](#_gfz0up9pexo0) 7

[Test Case ID: #6 (Disk Info)](#_y7xsy8x1rzkk) 8

[Test case ID: #7 (Alerts)](#_va4msa1kc979) 9

[Test case ID: #8 (Unit Test 1 - Toggle Display)](#_qs1nl1bscr4u) 10

[Test Case ID: # 9 (Unit Case 2 – Search Storage Name)](#_93uclzsc9yca) 11

[Test Case ID #10 (Integration Test)](#_e8cnapuh44r7) 12

## Test Case ID: #1 (Login Functionality)

**Description:** The login functionality allows user access control and encapsulation of individual accounts. Users will be asked create a unique password that will be hashed and mapped to an account identification number. The map of IDs and hashes will be stored on a database.

**Objective:** To ensure that a user's login attempt can be properly verified (correct passwords trigger a login, incorrect passwords trigger a failure)

**Testing approach:** Black-box

**Preconditions:**

1. The user is not logged
2. At least one properly configured account exists in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | No ID and no password are entered | No data | Log in failure |
| 2. | No ID and a correct password are entered | A correct password corresponding to a hash in the database | Log in failure |
| 3. | No ID and an incorrect password are entered | An incorrect password | Log in failure |
| 4. | A correct ID and no password are entered | An ID in the database | Log in failure |
| 5. | An incorrect ID and no password are entered | An ID not in the database | Log in failure |
| 6. | An incorrect ID and a correct password are entered | An ID not in the database and a password corresponding to a hash in the database | Log in failure |
| 7. | An incorrect ID and an incorrect password are entered | An ID not in the database and a password that does not correspond to a hash in the database | Log in failure |
| 8. | A correct ID and an incorrect password are entered | An ID in the database and a password that does not correspond to a hash in the database | Log in failure |
| 9. | A correct ID and a correct password (Noncorresponding) are entered | An ID in the database and a password that corresponds to a hash in the database but do not map to each other | Log in failure |
| 10. | A corresponding ID and password are entered | An ID in the database and the password for the corresponding hash | Log in success |

**Post Condition**: The user is logged in

## Test Case ID: #2 (Aggregated Info For Users)

**Description:** Individual clients of the product will have convenient access to details across all owned storage spaces in the service. Clients have the ability to access specific information by including their search specifications into the aggregated information function. HP employees will have universal access to storage space information across all users and can access any specific information at will.

**Objective:** To make sure that the system displays a storage list of view for the user that is accurate.

**Testing approach:** Black-box

**Preconditions:**

1. Actor is logged in
2. Actor is on the dashboard view

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | System is unable to connect to database | No data | “Please check connection” message is displayed |
| 2. | System connects to database and collects list of user’s disks | Database of different users’ storage disks | System displays storage list view of user’s disks |

**Post conditions:** System displays storage list view.

## 

## 

## 

## 

## 

## 

## Test Case ID: #3 (Data Visualization)

**Description:** Our data visualization will generate state of the art GUI representing the aggregated information specified by the clients. Visualization provides a comprehensive high level understanding for both the client and provider on the status and details on each data storage unit.

**Objective:** To make sure that the system displays the intended dashlets for the current data.

**Testing approach:** Black-box

**Preconditions:**

1. Actor is logged in
2. System has fetched storage list information from database

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | System is unable to connect to database | No data | “Please check connection” message is displayed |
| 2. | System passes information to visualization tool | Database of different users’ storage disks | System displays dashlets |

**Post conditions:** System displays dashlets.

## 

## 

## 

## 

## 

## 

## 

## Test Case ID: #4 (Filter)

**Description**: The filter functionality allows the user to refine the list of disks being displayed in the storage view. The user can filter the storage list based on parameters such as health, type of device, etc.

**Objective**: To make sure the filter functionality of the mobile app works as expected (for filtering by health, device type, and storage capacity).

**Testing approach**: Black-box

**Preconditions**:

1) User must be logged in

2) User must be connected to the database

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | User selects filter option | A list of devices with different attributes | System matches selected filters with entries in database and compiles a list based on relevant entries |

**Postconditions**: The app accurately filters out the devices based on the filter option chosen

## Test Case ID: #5 (Storage List View)

**Description**: Storage list view displays the integrated data of all the storages that the user owns. All the information is displayed graphically in a storage system.

**Objective**: To make sure the storage list displays the integrated data correctly.

**Testing approach**: Black-box

**Preconditions**:

1. Actor is logged in
2. Actor is in storage list view

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | Click on the Storage List toggle | Any user with at least one disk | The screen with all the disks displays |
| 2. | Click on one of the folded disks | Same as above | The disks with the correct content displays |

**Postconditions**: System displays the storage list.

## Test Case ID: #6 (Disk Info)

**Description**: When the user tap on a specific disk, the detailed disk information should show.

**Objective**: The test is to make sure the system can retrieve the detailed disk data from the database and display them.

**Testing approach**: Black-box

**Preconditions**:

1. Actor must be logged in
2. Actor must own at least one disk
3. Actor is in storage list view

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | Click on an individual disk in the storage list | Any of the disks in the storage list view | The content of the main part of the app changes to the detailed information of the disk |

**Postconditions**: The screen should display the disk information. If the back button is pressed or swipe the screen to the left, the app should go back to the storage list view.

**Exception**: If the app is not connected to the database, an error message “please check connection” is displayed.

## Test case ID: #7 (Alerts)

**Description**: The user should be able to see a notification that appears as soon as a disk’s storage is 10% or less and will stay there until storage is more than 10%.

**Objective**: The test ensures that once the free storage space falls under a certain amount, a message will appear notifying the user that disk storage is low.

**Testing approach**: Black-box

**Preconditions**:

1. The actor is logged in.
2. Actor owns at least one disk.
3. Actor is in storage list view.

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | User checks storage capacity left in specific disks in storage list view | The space left is above 10% in a specific disk | No alerts displayed in alerts, or problematic disk in storage list |
| 2. | User checks storage capacity left in specific disks in storage list view | The space left is 10% or below in a specific disk | System displays an alert next to problematic disk in storage list and a message, “storage low”, percentage left, and name of disk appears in alerts page |

**Postconditions**: the system displays an alert on the dashboard and alerts page if storage falls under 10%, and nothing if the condition is not true.

## Test case ID: #8 (Unit Test 1 - Toggle Display)

**Description**: The user should be able to toggle back and forth between the dashboard and storage list views using a button on the top menu. After entering valid credentials, the user should be able to go to the dashboard from the log in screen.

**Objective**: When on the log in or storage list view, toggling the display should switch to the dashboard. When on the dashboard view, toggling the display should switch to the storage list.

**Testing approach**: White-box

**Preconditions**:

1. The user has been successfully authenticated.

|  |  |  |  |
| --- | --- | --- | --- |
| Step number | Test case steps | Test data | Expected result |
| 1. | Call toggle() while on the log in screen. | No data | The view switches to the dashboard screen. |
| 2. | Call toggle() while on the dashboard screen. | No data | The view switches to the storage list screen. |
| 3. | Call toggle() while on the storage list screen. | No data | The view switches to the dashboard screen. |

**Post condition**s: Expected results match the results of calling toggle() from each view.

## Test Case ID: # 9 (Unit Case 2 – Search Storage Name)

**Description**: The User should be able to search for a storage system by name and get the correct result.

**Objective**: When the storage system name is enter into the search bar the correct storage system should be displayed.

**Testing approach**: White-box

**Preconditions**:

1. The user has been successfully authenticated and has at least one storage system.

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1. | Enter storage name into search bar | Search text | Show storage device |

**Postconditions**: The correct storage device is displayed.

## Test Case ID #10 (Integration Test)

**Description**: Generating a dashlet visualization based on the data in the database.

**Objective**: When the dashboard view is entered, the app generates a total capacity graph that match the data in the database.

**Testing approach**: White-box

**Preconditions**: The user is moving into the dashboard view and the test data exists in the database

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Test Case Steps | Test Data | Expected Result |
| 1 | The user clicks on the dashboard button | No data | The user is brought to the dashboard |
| 2 | The system fetches the required data from the database | System capacity test data | The data exists on the device to be used in the dashlet generation |
| 3 | The system uses the information fetched from the database to generate a dashlet | System capacity test data | The dashlet is generated to be displayed on the dashboard |
| 4 | The dashlet is displayed on the dashboard | System capacity dashlet | A graph is displayed that correctly represents the data fetched from the database |

**Postconditions**: There is a correctly generated dashlet displayed on the screen based on the information stored in the database. Correctly generating the dashlet means that the information displayed matches the information stored in the database