MediStore Manager Test Plan and Results

Jackson Rodgers and Kevin Sherman

Overall Test Plan

Our testing consists of two primary areas. First, we will create automated tests to test the create, read, update, and delete operations for all relevant tables in the database. This will ensure the code interacts properly with the existing database. Second, we will test the functionality of the application itself. This will involve interacting with the application and verifying the proper create, read, update, and delete operations are performed on the database and updated in the application. This will test the communication between the software and the database, along with overall functionality.

Test Case Descriptions

DB1.1 **Database Test 1**
DB1.2 This test will test the application's ability to create records in the database.
DB1.3 For this test we will run the database and feed it new data from classes created to store data before sending it to the database. The test will be repeated for each table in the database.
DB1.4 Inputs: Entries into the database just like the ones users will create.
DB1.5 Outputs: Newly created records in the database matching the data entered by the user.
DB1.6 Normal
DB1.7 Whitebox
DB1.8 Functional
DB1.9 Unit
DB1.10 Results:

DB2.1 **Database Test 2**
DB2.2 This test will test the application's ability to read data from the database.
DB2.3 For this test we will run the database, then send a read query from the application and temporarily store the retrieved data in the application. The test will be repeated for each table in the database.
DB2.4 Inputs: Existing database records.
DB2.5 Outputs: The retrieved data should match what is present in the database.
DB2.6 Normal
DB2.7 Whitebox
DB2.8 Functional
DB2.9 Unit
DB2.10 Results:

DB3.1 **Database Test 3**

DB3.2 This test will test the application’s ability to update records in the database.

DB3.3 For this test we will run the database, then retrieve some data from the database. Then we will change some of the data and send an update query back to the database. The test will be repeated for each table in the database.

DB3.4 Inputs: Edited database records.

DB3.5 Outputs: The updated information should be reflected in the database.

DB3.6 Normal

DB3.7 Whitebox

DB3.8 Functional

DB3.9 Unit

DB3.10 Results:

DB4.1 **Database Test 4**

DB4.2 This test will test the application’s ability to remove records from the database.

DB4.3 For this test we will run the database, then send a delete query from the application along with the unique identifiers for specific records in the database. The test will be repeated for each table in the database.

DB4.4 Inputs: Unique identifiers for database records.

DB4.5 Outputs: The specified records should no longer be present in the database.

DB4.6 Normal

DB4.7 Whitebox

DB4.8 Functional

DB4.9 Unit

DB4.10 Results:

DB5.1 **Database Test 5**

DB5.2 This test will measure the application’s ability to handle large amounts of data.

DB5.3 For this test we will populate the database with a large amount of data. Then we will create a series of queries that should touch every table in the database (excluding the users table). We will run these queries alongside a timer to measure how long it takes to retrieve the requested information.

DB5.4 Inputs: Queries to search through the entire database.

DB5.5 Outputs: The results of the queries along with the time taken to complete them.

DB5.6 Normal

DB5.7 Whitebox

DB5.8 Performance

DB5.9 Unit

DB5.10 Results:

A1.1 **Application Test 1**

A1.2 This test will test the application's ability to interact with the database to create, read, update, and delete records in the database relating to Patient Information.

A1.3 For this test, we will interact with the application by selecting the Add Patient button and filling in necessary information. This patient will then be selected in the list and the information will be viewed on the right side. Then, the Edit Patient Information button will be selected and the values will be changed. Verify the information is updated, then select Edit Patient Information again and select delete. Finally, verify the patient no longer appears in the list.

A1.4 Inputs: Button clicks and test information

A1.5 Outputs: Newly created records in the database matching the data entered by the user, edited information, verified deletion.

A1.6 Normal

A1.7 Blackbox

A1.8 Functional

A1.9 Integration

A1.10 Results:

A2.1 **Application Test 2**

A2.2 This test will test the application's ability to interact with the database to create, read, update, and delete records in the database relating to Inventory Information.

A2.3 For this test, we will interact with the application by selecting the Add Inventory button and filling in necessary information. This item will then be selected in the list and the information will be viewed on the right side. Then, the Edit Item Information button will be selected and the values will be changed. Verify the information is updated, then select Edit Item Information again and select delete. Finally, verify the item no longer appears in the list.

A2.4 Inputs: Button clicks and test information

A2.5 Outputs: Newly created records in the database matching the data entered by the user, edited information, verified deletion.

A2.6 Normal

A2.7 Blackbox

A2.8 Functional

A2.9 Integration

A2.10 Results:

A3.1 **Application Test 3**

A3.2 This test will test the application's ability to interact with the database to create, read, update, and delete records in the database relating to Supplier Information.

A3.3 For this test, we will interact with the application by selecting the Add Supplier button and filling in necessary information. This supplier will then be selected in the list and the information will be viewed on the right side. Then, the Edit Supplier Information button will be selected and the values will be changed. Verify the information is updated, then select Edit Supplier Information again and select delete. Finally, verify the supplier no longer appears in the list.

A3.4 Inputs: Button clicks and test information

A3.5 Outputs: Newly created records in the database matching the data entered by the user, edited information, verified deletion.

A3.6 Normal

A3.7 Blackbox

A3.8 Functional

A3.9 Integration

A3.10 Results:

A4.1 **Application Test 4**

A4.2 This test will test the application’s search bar for the listed items.

A4.3 This test will be performed on all main tabs. At the top left of the page there is a search bar. Click into it and start entering the name of the patient/item/supplier/ticket used for the test. Verify that the proper listing appears.

A4.4 Inputs: Keyboard entries and existing database information.

A4.5 Outputs: Proper existing entry appears in the list.

A4.6 Normal

A4.7 Blackbox

A4.8 Functional

A4.9 Integration

A4.10 Results:

A5.1 **Application Test 5**

A5.2 This test will test the application’s ability to create work order tickets.

A5.3 From the Patients or Inventory tab, select Create Work Order. Fill in necessary information on the pop-up window and select okay. Go to Order Tickets tab and verify the work order was created.

A5.4 Inputs: Button clicks and test data.

A5.5 Outputs: New work order ticket created.

A5.6 Normal

A5.7 Blackbox

A5.8 Functional

A5.9 Integration

A5.10 Results:

A6.1 **Application Test 6**

A6.2 This test will test the application’s ability to create supply order tickets.

A6.3 From the Suppliers tab, select Create Supply Order. Fill in necessary information on the pop-up window and select okay. Go to Order Tickets tab and verify the supply order was created.

A6.4 Inputs: Button clicks and test data.

A6.5 Outputs: New supply order ticket created.

A6.6 Normal

A6.7 Blackbox

A6.8 Functional

A6.9 Integration

A6.10 Results:

A7.1 **Application Test 7**

A7.2 This test will test the application’s login functionality.

A7.3 Launch the application and verify that a login pop-up window appears. Fill in the fields with login information and select okay. Verify login is successful and application fully launches.

A7.4 Inputs: Login information

A7.5 Outputs: Login was successful and application has launched.

A7.6 Normal

A7.7 Blackbox

A7.8 Functional

A7.9 Integration

A7.10 Results:

A8.1 **Application Test 8**

A8.2 This test will test the application’s login functionality.

A8.3 Launch the application and verify that a login pop-up window appears. Fill in the fields with incorrect login information and select okay. Verify login is not successful and a message appears saying that.

A8.4 Inputs: Incorrect login information

A8.5 Outputs: Login was unsuccessful and application does not launch.

A8.6 Abnormal

A8.7 Blackbox

A8.8 Functional

A8.9 Integration

A8.10 Results:

A9.1 **Application Test 9**

A9.2 This test will test the application’s ticket history tab.

A9.3 Select the Order Tickets tab. Verify that previous work and supply order tickets are listed in their respective tabs.

A9.4 Inputs: Button clicks

A9.5 Outputs: History of work and supply order tickets.

A9.6 Normal

A9.7 Blackbox

A9.8 Functional

A9.9 Integration

A9.10 Results:

Test Case Matrix

	Normal/ Abnormal	**Blackbox/ Whitebox**	**Functional/ Performance**	**Unit/ Integration**
DB1	Normal	Whitebox	Functional	Unit
DB2	Normal	Whitebox	Functional	Unit
DB3	Normal	Whitebox	Functional	Unit
DB4	Normal	Whitebox	Functional	Unit
DB5	Normal	Whitebox	Performance	Unit
A1	Normal	Blackbox	Functional	Integration
A2	Normal	Blackbox	Functional	Integration
A3	Normal	Blackbox	Functional	Integration
A4	Normal	Blackbox	Functional	Integration
A5	Normal	Blackbox	Functional	Integration
A6	Normal	Blackbox	Functional	Integration
A7	Normal	Blackbox	Functional	Integration
A8	Abnormal	Blackbox	Functional	Integration
A9	Normal	Blackbox	Functional	Integration