

Test Case ID: TC_001

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Starting the application, including setting up and importing the database structure.

Brief description: This test case verifies the successful execution of the application, including the setup and import of the database structure.

Pre-conditions: Python, MySQL Workbench and MySQL Server installed.

Dependencies and Requirements: The database should be accessible.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Start MySQL Server		Server starts successfully		
2	Start MySQL Workbench		Database connection established		
3	Connect to MySQL server		MySQL Workbench successfully connects to server		
4	Import database structure		Database with tables and test data will be created		
4	Verify database tables and schema		Database will contain all 6 tables, 2 views and test data		
5	Enter project root directory		Project is installed and accessible		
6	Setup config	File: config/config.ini	config.ini is properly set up to be able to connect to database		
7	Enter src folder and start cmd/terminal		Project source code is installed and accessible		
8	Run command <code>python main.py</code> to open the application		Program successfully starts and <code>Console:~\$</code> will be shown		
8	Verify connection, run command <code>connect</code> in program		Program successfully informs you about connection to database		

Test Case ID: TC_002

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Importing Data into the database

Brief description: This test case verifies the successful import of data into the database.

Pre-conditions: The application and database environment should be properly configured. Data to be imported are available in folder 'data' in file 'import.xml' in format suitable for import.

Dependencies and Requirements: The application should be installed and configured properly. The database should be accessible. Data to be imported should be available.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Run command <code>python main.py</code> to open the application		Program successfully starts and <code>Console:~\$</code> will be shown		
2	Verify connection, run command <code>connect</code> in program		Program successfully informs you about connection to database		
3	Enter <code>manager mode</code> , type <code>manager</code>		Message indicating success will be shown and console will become <code>Console:~(DATABASE NAME)\$</code>		
4	Prepare the data to be imported	File: data/import.xml	Data are prepared in file '/data/import.xml'		
5	Run command <code>import</code> to import the data	File: data/import.xml	Program informs about successful operation		
6	Verify imported data in the database		Imported data are visible in the database		

0	Verify imported data in the database	Imported data are visible in the database		
---	--------------------------------------	---	--	--

Test Case ID: TC_003

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Inserting Data into the Database

Brief description: This test case verifies the successful insertion of data into the database.

Pre-conditions: The application and database environment should be properly configured. Default data are stored in database.

Dependencies and Requirements: The application should be installed and configured properly. The database should be accessible.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Open the application		Application starts successfully		
2	Ensure database connection is established		Database connection established		
3	Enter manager mode		Console:~(DATABASE NAME)\$ is shown		
4	Run command insert		Console:~(DATABASE NAME-SELECT TABLE)\$ is shown		
5	Enter table named owner		You can add values to be inserted into table row		
6	Enter all the data with correct type, which is shown in format <DATA TYPE>	Strings/Integers: 2024-12-12 , 2024-12-13 , PUBLIC , Test , 1	Data passed successfully		
7	Fill the last value and click enter		Message indicating success is shown		
8	Verify inserted data in the database		Inserted data visible are shown in the database		

Test Case ID: TC_004

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Getting database report

Brief description: This test case verifies the function of database report.

Pre-conditions: The application and database environment should be properly configured. Default data are stored in database.

Dependencies and Requirements: The application should be installed and configured properly. The database with data should be accessible.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Open the application		Application starts successfully		
2	Ensure database connection is established		Database connection established		
3	Enter manager mode		Console:~(DATABASE NAME)\$ is shown		
4	Run command report		Multiple options of possible reports are shown		
5	Enter choice number 1		Text table including data in database from view are shown		

Test Case ID: TC_005

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Selecting all rows of table

Brief description: This test case verifies that program can obtain all the data of any table

Pre-conditions: The application and database environment should be properly configured. Default data are stored in database.

Dependencies and Requirements: The application should be installed and configured properly. The database with data should be accessible.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Open the application		Application starts successfully		
2	Ensure database connection is established		Database connection established		
3	Enter manager mode		Console:~(DATABASE NAME)\$ is shown		
4	Run command select		Console:~(DATABASE NAME-SELECT TABLE)\$ is shown		
5	Enter table named auction		You can add values by which program will filter		
6	Enter all attributes blank		You are always be redirected to next attribute		
7	Click enter once you reached last attribute		All the rows of table auction are shown in text table		
8	Verify retrieved rows match all the rows in database table auction		All data are shown and match all rows of table auction		

Test Case ID: TC_006

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Updating table row

Brief description: This test case verifies that program can modify any row of any table

Pre-conditions: The application and database environment should be properly configured. Default data are stored in database.

Dependencies and Requirements: The application should be installed and configured properly. The database with data should be accessible.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Open the application		Application starts successfully		
2	Ensure database connection is established		Database connection established		
3	Enter manager mode		Console:~(DATABASE NAME)\$ is shown		
4	Run command update		Console:~(DATABASE NAME-SELECT TABLE)\$ is shown		
5	Enter table named bidder		You can add values by which program will filter		
6	Enter ID value to 1	Integer: 1	You are redirected to next attributes		
7	Enter SURNAME to value Test	String: Test	You will always be redirected to next attribute		
8	Leave all other attributes blank		You are always be redirected to next attribute		
9	After all attributes are skipped, click enter on the last one		Message indicating seccussful modification is shown		
10	Verify updated SURNAME in the database		Updated SURNAME matches value Test in table bidder		

Test Case ID: TC_007

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Deleting row of table

Brief description: This test case verifies that program can delete any row of any table

Pre-conditions: The application and database environment should be properly configured. Default data are stored in database.

Dependencies and Requirements: The application should be installed and configured properly. The database with data should be accessible.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Open the application		Application starts successfully		
2	Ensure database connection is established		Database connection established		
3	Enter manager mode		Console:~(DATABASE NAME)\$ is shown		
4	Run command delete		Console:~(DATABASE NAME-SELECT TABLE)\$ is shown		
5	Enter table named bidder		You can add values by which program will filter		
6	Enter ID value to 1	Integer: 1	You are redirected to next attributes		
7	Enter SURNAME to value Smith	String: Smith	You will always be redirected to next attribute		
8	Leave all other attributes blank		You are always be redirected to next attribute		
9	After all attributes are skipped, click enter on the last one		Message indicating seccussful modification is shown		
10	Verify deleted SURNAME is not in the database		Database table bidder does not contain row with ID 1 and with SURNAME Smith		

Test Case ID: TC_008

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Using invalid command

Brief description: This test case verifies that the program can handle invalid commands

Pre-conditions: The application is installed.

Dependencies and Requirements: The application should be installed.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Open the application		Application starts successfully		
2	Enter command test		Message indicating Error is shown		
3	Run command help		Program works and list of available commands is shown		

Test Case ID: TC_009

Test Designed by: Erik Boháč

Test Designed date: 04.02.2024

Test Title: Invalid data passed to CRUD operation

Brief description: This test case verifies that the program can handle invalid user data inputs

Pre-conditions: The application is installed.

Dependencies and Requirements: The application should be installed.

Step	Test Steps	Test Data	Expected Result	Passed	Notes
1	Open the application		Application starts successfully		
2	Ensure database connection is established		Database connection established		
3	Enter manager mode		Console:~(DATABASE NAME)\$ is shown		
4	Run command delete		Console:~(DATABASE NAME-SELECT TABLE)\$ is shown		

5	Enter table named <code>bidder</code>		You can add values by which program will filter		
6	Enter ID value to <code>abc</code>		Operation is interrupted and <code>Console:~(DATABASE NAME)\$</code> is shown again		