REVISION HISTORY

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 30.12.2023 | 1.0 | Tests are divided into groups and Account & Login tests completed and reported. | Sevval Kurt  Azra Acil |
| 31.12.2023 | 1.1 | Vehicle Rent Requirement, User Information & Filter Process tests completed and reported. | Kerem Bekmez  Ahmet Pehlivanoglu  Mehmet Kutlutan |
| 31.12.2023 | 1.2 | The remaining 6 tests completed and added to the Software Test Report document. | Azra Acil,  Mehmet Kutlutan,  Kerem Bekmez, Ahmet Pehlivanoglu,  Sevval Kurt |
| 1.01.2024 | 1.3 | Final Revision is done, and Overall Assessment of Tests is filled in. | Azra Acil,  Mehmet Kutlutan,  Kerem Bekmez, Ahmet Pehlivanoglu,  Sevval Kurt |

**TABLE OF CONTENTS**

[**1 Introduction 3**](#_heading=)

[1.1 Document overview 3](#_heading=h.1fob9te)

[1.2 References 3](#_heading=)

[1.2.1 Project References 3](#_heading=h.2et92p0)

[1.3 Conventions 3](#_heading=h.tyjcwt)

[**2 Overview of Tests Results 4**](#_heading=)

[2.1 Tests log 4](#_heading=h.1t3h5sf)

[2.2 Rationale for decision 4](#_heading=)

[2.3 Overall assessment of tests 4](#_heading=)

[2.4 Impact of test environment 5](#_heading=)

[**3 Detailed Tests Results 6**](#_heading=)

[3.1 Account Test 6](#_heading=)

[3.2 Login Tests 7](#_heading=)

[3.3 User Information Test 9](#_heading=)

[3.4 Filter Process Test 10](#_heading=)

[3.5 Rental Process Test 12](#_heading=)

[3.6 Payment Test 12](#_heading=h.qr5mrn1a5xq6)

[3.7 Car Rent Requirements Tests 14](#_heading=h.q5ktghejt1jw)

[3.8 User Rental History Test 16](#_heading=h.z152dh6qiiqf)

[3.9 Admin Features Test 17](#_heading=h.aeh35nh8dycp)

[3.10 Admin Rental History Test 18](#_heading=h.vlitxm7rumah)

[3.11 Filter Removal Test 18](#_heading=h.4wg01k3d65qh)

# Introduction

## Document overview

This document is the software test report of the testing phase of the VRS software development project. It contains the results of tests performed on the developed system.

## References

### Project References

| # | Document Identifier | Document Title |
| --- | --- | --- |
| [STP] | VRS-STP-1 | VRS Software Test Plan |
| [SRS] | VRS-SRS-1 | VRS Software Requirements Specifications |

## Conventions

Each test case will be shown in the following table format:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | <ID of the test> |  | <Decisions> |
| Test description | <A brief description of the test> |  |  |
| Verified Requirement | <ID numbers of requirements that are to be verified with this test.> | Verification method: <I,A,D,T> |  |
| Initial conditions | <The state of software before test> |  |  |
| Tests inputs | <Input data from any test tool, input files name and location, if any > |  |  |
| Data collection actions | <Recording and post processing of output data, if any > |  |  |
| Tests outputs | <Output data files, if any> |  |  |
| Assumptions and constraints | If any, may be limited access to a tool, license … |  |  |
| Expected results and criteria | <list of expected results> | <special criteria to be met by the result, if any> |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** |  |
| 1 | <actions listed row by row> | <expectation after each action if any> |  |
| 2 | … | … |  |

# Overview of Tests Results

## Tests log

The VRS software (version 0.10.1) was tested on the Microsoft Windows test platform located in Istanbul, from the 2023/12/30 to the 2023/12/31. The tests of the test phase (ref. software test plan) where executed.

Testers where:

* Sevval Kurt,
* Ahmet Pehlivanoglu,
* Mehmet Kutlutan
* Azra Acil
* Kerem Bekmez

## Rationale for decision

After executing a test, the decision is defined according to the following rules:

* **OK:** The test sheet is set to "OK" state when all steps are in "OK" state. The real result is compliant to the expected result.
* **NOK:** The test sheet is set to "NOK" state when all steps of the test are set to "NOK" state or when the result of a step differs from the expected result.
* **NOT RUN:** Default state of a test sheet not yet executed.
* **NOT COMPLETED:** The test sheet is set to "Not Completed" state when at least one step of the test is set "Not Run" state.

Test results are listed in §3.

## Overall assessment of tests

* Website works as expected in the testing, but it may slow down due to the number of users.
* Website functions correctly, but the design could be updated for a more modern, appealing appearance.
* 2 test cases in particular, even though they are functioning as expected, the UI messages are not as specified in the Software Test Plan document.
* All of the remaining tests are working as specified in the Software Test Plan document.

Statistics about tests:

* 100% of tests OK,
* 0% of tests NOK
* 0% of tests POK
* 0% of tests NR
* 0% of tests NC

Give also statistics about bugs and enhancements:

* Total number: 0
* Number of Critical: 0
* Number of Major: 0
* Number of minor: 0
* Number of enhancements: 20

## Impact of test environment

None.

# Detailed Tests Results

For each executed test, this document contains:

* Test identification;
* Test title;
* Test decision;
* A comment containing additional information or problems encountered during execution and differences with the test procedure.

For the problems leading to a bug, the bug ID is reported in the result of the step where the problem was encountered.

## Account Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-001** | **Comment** | **Decision** |
| Test description | Test to confirm users can create new accounts within 2 minutes and they get a pop-up window validating that the account has been created successfully. |  | **OK** |
| Verified Requirement | SRS-REQ-001 | D |  |
| Initial conditions | N/A | N/A |  |
| Tests inputs | Username and Password | Filling the username and passwords correctly. |  |
| Data collection actions | Adding the user’s data to the database | Assigning an ID to the user and adding the information to the database. |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | The UI shall return a message indicating the user has been created successfully. |  |  |
| Expected results and criteria | The user should be created and added to the database within 3 minutes. | In the database, the total user count shall increase by one. |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Fill in the username | N/A | **OK** |
| 2 | Fill in the password | N/A | **OK** |
| 3 | Click the sign-up button and start the timer. | The user gets saved to DB and the user count is increased by one. | **OK** |
| 4 | Once the “Successful sign-up” pop-up message is received, the operator will stop the timer and check if it is under 3 minutes. | The UI returns a message indicating the result of the sign-up attempt. | **OK** |

## 3.2 Login Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-002.1** | **Comment** | **Decision** |
| Test description | Test to confirm users can login to their existing accounts. |  | **OK** |
| Verified Requirement | SRS-REQ-002.1 | D |  |
| Initial conditions | The user shall be already present in the database | Adding the demonstration user to the database. |  |
| Tests inputs | Username and Password | Filling the username and passwords correctly |  |
| Data collection actions | Checking the user’s existence in the database | Returning a Boolean if the user exists. |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | The UI shall let the user to the renting page. |  |  |
| Expected results and criteria | The user shall successfully log-in within a minute and see its balance, past rentals, etc. | The UI should change the login screen to the rent screen. |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Fill in the correct username | N/A | **OK** |
| 2 | Fill in the correct password | N/A | **OK** |
| 3 | Click the log-in button and start the timer. | The user is searched with a database query and returns true. | **OK** |
| 4 | Once the user gets the log-in message, the timer should stop and check if it was under a minute. | The UI changes the window to the renting page if successful, otherwise returns a pop-up message saying username or password was wrong. | **OK** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-002.2** | **Comment** | **Decision** |
| Test description | Test to confirm users cannot login with wrong credentials. |  | **OK** |
| Verified Requirement | SRS-REQ-002.2 | D |  |
| Initial conditions | The user shall be already present in the database | Adding the demonstration user to the database. |  |
| Tests inputs | Username and Password | Filling the wrong username and passwords. |  |
| Data collection actions | Checking the user’s existence in the database | Returning a Boolean if the user exists. |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | The UI shall not let the user to the renting page. |  |  |
| Expected results and criteria | The user shall get a pop-up message indicating that the user’s credentials are wrong. | The UI shall stay in the log-in page. |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Fill in the wrong username | N/A | **OK** |
| 2 | Fill in the wrong password | N/A | **OK** |
| 3 | Click the log-in button. | The user is searched with a database query and returns false. | **OK** |
| 4 | The user shall get a pop-up window indicating the user credentials are wrong. | The UI displays a pop-up window with the wrong credentials message. | **OK** |

## 3.3 User Information Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-003** | **Comment** | **Decision** |
| Test description | Test to confirm that users can see their balance. |  | **OK** |
| Verified Requirement | SRS-REQ-003 | I |  |
| Initial conditions | Users should already be logged in and be on the renting page. | Successfully log-in with an already existing user. |  |
| Tests inputs | N/A | N/A |  |
| Data collection actions | N/A | N/A |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | On the renting page UI, balance shall be seen in a green box. |  |  |
| Expected results and criteria | The tester shall see the balance of the user in the renting page. | Controlling if the amount matches with the user’s amount in the DB. |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Fill in the username | N/A | **OK** |
| 2 | Fill in the password | N/A | **OK** |
| 3 | Click the log-in button | The tester logs into a user’s account, and UI changes to the rent screen. | **OK** |
| 4 | The operator shall be able to spot it in the rental page without pressing anywhere. | The UI shall show the balance in a green box. | **OK** |

## 3.4 Filter Process Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-004** | **Comment** | **Decision** |
| Test description | Test to confirm if the filter operation is working correctly |  | **OK** |
| Verified Requirement | SRS-REQ-004 | D |  |
| Initial conditions | DB shall have enough vehicles with different attributes so that the filter operation can be tested using demonstration correctly. | Adding vehicles with different attributes that will be used on the demonstration filtering. |  |
| Tests inputs | Different filters selected from the UI | N/A |  |
| Data collection actions | N/A | The database shall apply the requested filters using a query. |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | If there are any matches with the filter, the UI will show them. |  |  |
| Expected results and criteria | Giving different filters will change the UI, and regardless of the filter number, the query time won’t exceed 5 seconds. | N/A |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Apply vehicle filter to car, start the timer and hit the search button. | Database filters cars from all vehicles and passes the data to the business layer, which then updates the UI accordingly. | **OK** |
| 2 | Once the UI changes, stop the timer, verify that it was under 5 seconds, restart the timer, apply vehicle color and year filters, and start the timer while hitting the search button again. | Database adds vehicle color and year to the query and passes the data to the business layer, which then displays it on the screen. | **OK** |
| 3 | Once the UI changes, the operator shall stop the timer and verify that even with more filters applied, the time is under 5 seconds. |  | **OK** |

## 3.5 Rental Process Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-005.1** | **Comment** | **Decision** |
| Test description | Test to confirm user cannot click the rent button if the dates are not selected. |  | **OK** |
| Verified Requirement | SRS-REQ-005.1 | D |  |
| Initial conditions | Not selecting a date for the rental. | N/A |  |
| Tests inputs | N/A | N/A |  |
| Data collection actions | N/A | N/A |  |
| Tests outputs | Verify that the rent button is not clickable if date information is not given. | N/A |  |
| Assumptions and constraints | The user selects a vehicle, does not give a rent date and hits the rent button. | N/A |  |
| Expected results and criteria | The rent button shall not be responsive if the date information is not given. | Nothing happens in the background. |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Select a vehicle | List of available vehicles | **OK** |
| 2 | Hit the rent button. | Check if the dates are given and if not, do not return anything. | **OK** |
| 3 | Operator verifies that the button is unresponsive. | N/A | **OK** |
| Test description | Test to confirm user cannot click the rent button if the dates are not selected. |  | **OK** |

## 3.6 Payment Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-006** | **Comment** | **Decision** |
| Test description | Test to confirm the user gets the correct total payment amount for the selected dates and the vehicle. |  | **OK** |
| Verified Requirement | SRS-REQ-006 | D |  |
| Initial conditions | A vehicle that is available for the dates to be tested. | N/A |  |
| Tests inputs | Vehicle, Number of Rental Days | Selected vehicles must be available. |  |
| Data collection actions | N/A | N/A |  |
| Tests outputs | Verify that the total payment amount is correct | N/A |  |
| Assumptions and constraints | The user selects a valid vehicle and enters a valid number of rental days. The UI can calculate the total payment amount correctly. | N/A |  |
| Expected results and criteria | The total payment amount displayed is correct. | In the UI, the total payment calculated correctly |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Select a vehicle | List of available vehicles | **OK** |
| 2 | Enter the number of days you want to rent the vehicle for | Total Payment is calculated and displayed to the screen | **OK** |
| 3 | Operator verifies the payment amount is correctly calculated. | N/A | **OK** |

## 3.7 Car Rent Requirements Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-007.1** | **Comment** | **Decision** |
| Test description | Test to confirm that users can rent the available vehicles if their amount is sufficient. |  | **OK** |
| Verified Requirement | SRS-REQ-007.1 | D |  |
| Initial conditions | Car availability and user’s money amount | Selected cars must be available, and the user amount is sufficient. |  |
| Tests inputs | Car and user’s money amount | Select the available car and have enough money. |  |
| Data collection actions | Updating the vehicle database | Renting operation will be added to vehicle renting schema |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | The UI shall return a message indicating the user has rented the selected car successfully. |  |  |
| Expected results and criteria | The renting should be done and added to the database within 10 seconds, the amount will be withdrawn from the user’s balance. | In the database, the total renting count shall increase by one and the rent shall be added to the rental histories. |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Filter the desired car attributes | Available cars get listed | **OK** |
| 2 | Select the desired car. | N/A | **OK** |
| 3 | Click rent button | The renting operation gets saved to DB and the renting count is increased by one. The user’s balance is reduced as much as the payment amount, and the vehicle’s availability converts to busy. | **OK** |
| 4 | If the rental is successful, the operator shall see a decrease in the balance box of the UI, get a verification pop-up with the correct rental information and the vehicle’s status shall return to busy for the rented dates. | The UI returns a message indicating the result of the renting attempt. | **OK** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-007.5** | **Comment** | **Decision** |
| Test description | Test to confirm users cannot rent if their amount is lower than payment or vehicle is not available. |  | **OK** |
| Verified Requirement | SRS-REQ-007.5 | D |  |
| Initial conditions | Car availability, user’s balance | Controlling selected vehicles must be busy or the user must have insufficient amount. |  |
| Tests inputs | User balance, car availability |  |  |
| Data collection actions | N/A | N/A |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | The UI shall return a message indicating the user has not rented the selected car. |  |  |
| Expected results and criteria | The user should not be renting a vehicle. |  |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Select a car that has a total payment higher than the user amount. | N/A | **OK** |
| 2 | Press the rent button | The system should check the total payment amount and user amount and return negative as the rent result | **OK** |
| 3 | The UI shall give a message indicating the rent failed. | N/A | **OK** |

## 3.8 User Rental History Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-008** | **Comment** | **Decision** |
| Test description | Test to confirm users to see their rental history. |  | **OK** |
| Verified Requirement | SRS-REQ-008 | D |  |
| Initial conditions | Selected users must have a rental saved to the DB beforehand. | N/A |  |
| Tests inputs | N/A | N/A |  |
| Data collection actions | N/A | N/A. |  |
| Tests outputs | N/A | Database returns the rental history. |  |
| Assumptions and constraints | The UI shall return a pop-up showing the previous rentals of the user. |  |  |
| Expected results and criteria | The user shall see the previous rentals with their rental information correctly. | N/A |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Click the see history button | History is listed in the database and returned to the business layer. | **OK** |
| 2 | N/A | The business layer passes the data to the UI, and it is shown to the user. | **OK** |
| 3 | Operator checks if the rental history is matching with the user’s rentals. |  | **OK** |

## 3.9 Admin Features Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-009** | **Comment** | **Decision** |
| Test description | Test to confirm admin can add, delete, and modify vehicles. |  | **OK** |
| Verified Requirement | SRS-REQ-009 | D |  |
| Initial conditions | Admin Login | N/A |  |
| Tests inputs | N/A | N/A |  |
| Data collection actions | Adding, deleting, modifying vehicle data to the database | Adding the information modified to the database. |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | Modifications must be seen in the vehicles page. |  |  |
| Expected results and criteria | The admin should create or modify a vehicle within 3 minutes and the updates must be seen visually. | In the database, changes must be updated. |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Admin log-in | N/A | **OK** |
| 2 | Add Vehicle button pressed | N/A | **OK** |
| 3 | The necessary information is added and the create button is clicked. | The new device should be added to the DB. | **OK** |
| 4 | Once the message is received, the operator shall check the added vehicle and see if it matches with the requested vehicle. | The UI returns a message indicating the result of the car addition operation. | **OK** |

## 3.10 Admin Rental History Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-011** | **Comment** | **Decision** |
| Test description | Test to confirm admin can see the rental history. |  | **OK** |
| Verified Requirement | SRS-REQ-011 | D |  |
| Initial conditions | VRS Software is running, and an admin is logged in already. | VRS Software has been used and all the rental operations have been saved to DB. |  |
| Tests inputs | N/A | N/A |  |
| Data collection actions | N/A | N/A |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | N/A | N/A |  |
| Expected results and criteria | Rental History of all the cars should appear on the UI. | N/A |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Log-in to admin | N/A | **OK** |
| 2 | Click the rental history button from the UI. | Database operations in the background should be done within 3 minutes. | **OK** |
| 3 | Operator shall see all the rentals with their correct, relevant information. | A pop-up shall return all the rentals with their relevant information. | **OK** |

## 3.11 Filter Removal Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test-012** | **Comment** | **Decision** |
| Test description | Test to confirm users can remove all filters applied. |  | **OK** |
| Verified Requirement | SRS-REQ-012 | D |  |
| Initial conditions | VRS software is running, and the user is in the view vehicles page and some filters are applied to the vehicles. | N/A |  |
| Tests inputs | N/A | N/A |  |
| Data collection actions | N/A | N/A |  |
| Tests outputs | N/A | N/A |  |
| Assumptions and constraints | N/A |  |  |
| Expected results and criteria | All applied filters should be removed and all vehicle data must be shown in the UI. | N/A |  |
| **Test procedure** |  |  |  |
| **Step number** | **Operator Actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Click to the remove filters button. | All applied filters should be removed | **OK** |
| 2 | The operator shall be able to see the starting page (the page that comes when the user first log’s in to its account). | UI should show every vehicle from the DB. | **OK** |