| Test Plan | **“qaAuto”** application. |
| --- | --- |
| Prepared by: | Tkachenko Sofiia  13.11.2023 |

**Test plan:**

1. **Introduction**

The “qaAuto” application helps the user keep track of fuel consumption and keep track of money spent on fuel. The application makes it easy to control, analyze and track expenses, and can also help you find the necessary instructions for your car.

The Test Plan has been created to describe approaches and methodologies that will apply to the unit, integration and system testing of the “qaAuto”. It includes the test items, approach, entry and exit criteria, environment needs, test responsibilities, risk and contingencies.

1. **Test items**

In this project, all functionality is subject to testing:

* [AutoList](https://confluence.ithillel.com/display/HA/AutoList)
* [Main page](https://confluence.ithillel.com/display/HA/Main+page)
* [Profiles](https://confluence.ithillel.com/display/HA/Profiles)

[Guest profile](https://confluence.ithillel.com/display/HA/Guest+profile)

[Guest fuel expenses](https://confluence.ithillel.com/display/HA/Guest+fuel+expenses)

[Guest garage](https://confluence.ithillel.com/display/HA/Guest+garage)

[Guest header](https://confluence.ithillel.com/display/HA/Guest+header)

[Guest instructions](https://confluence.ithillel.com/display/HA/Guest+instructions)

* [User profile](https://confluence.ithillel.com/display/HA/User+profile)

[Header](https://confluence.ithillel.com/display/HA/Header)

[Profile](https://confluence.ithillel.com/display/HA/Profile)

[User fuel expenses](https://confluence.ithillel.com/display/HA/User+fuel+expenses)

[User garage](https://confluence.ithillel.com/display/HA/User+garage)

[User instructions](https://confluence.ithillel.com/display/HA/User+instructions)

[User settings](https://confluence.ithillel.com/display/HA/User+settings)

**3.Approach**

**3.1 Techniques and types of testing that will be useful to us during testing**:

By levels:

1. Unit testing
2. Integration testing
3. System testing
4. Acceptance testing.

Purpose: check the compliance of the application functionality with the stated requirement.

*Functional testing* - testing of functional requirements of a system:

Sanity testing - testing specific functionality.

*Regression testing -* testing after changes in code, upgrades etc to check the new code has not affected the existing code.

*Non functional testing* - testing of non-functional requirements such as:

Load testing - testing the response time of the program to requests of various types.

Usability testing - testing the usability and attractiveness of the application.

Scalability testing - checks the flexibility of systems.

Testing techniques that will be used:

Equivalence partitioning

Boundary values

Pairwise technique.

**3.2 Metrics that will be collected during testing:**

Process metrics:

* Test Case Preparation Productivity = (No of Test Case) / (Effort spent for Test Case Preparation)
* Test Design Coverage = ((Total number of requirements mapped to test cases) / (Total number of requirements)\*100
* Test Execution Productivity = (No of Test cases executed) / (Effort spent for execution of test cases)
* Test Cases Pass = (Total no. of test cases passed) / (Total no. of test cases executed) \* 100
* Test Cases Failed = (Total no. of test cases failed) / (Total no. of test cases executed) \* 100
* Test Cases Blocked = (Total no. of test cases blocked) / (Total no. of test cases executed) \* 100

Product metrics:

* Error Discovery Rate = (Total number of defects found /Total no. of test cases executed)\*100
* Defect Leakage = ((Total no. of defects found in UAT)/(Total no. of defects found before UAT)) \* 100

Requirements Coverage - assessment of test coverage of functional and non-functional requirements for a product by constructing traceability matrices.

**3.3 Configurations**

Testing should be done on the following platforms:

1. iPhone with iOS 16 or newer
2. Android v14 or newer
3. Windows with 11 OS or newer:
   1. Web Browser Google Chrome v 118.0.5993.118 or newer
   2. Web Browser Opera v 104.0.4944.54 or newer
   3. Web Browser Edge v 119.0.2151.58 or newer.
4. MacOS v11.7.10 or newer:
   1. Web Browser Google Chrome v 118.0.5993.118 or newer
   2. Web Browser Safari v16.5.2 or newer
   3. Web Browser Edge v 119.0.2151.58 or newer.

**4.Entry criteria**

* All requirements are approved and defined
* Availability of sufficient and desired test data
* Complete or partially testable code is available.
* Test environment has been set-up and all other necessary resources such as tools and devices are available.
* Test cases are developed and ready.

**4.1 Exit criteria**

* Execution of all test cases.
* All test cases and most of them are PASS
* No Critical or Blocker defects outstanding.
* All necessary requirements are covered
* All the identified defects are corrected and closed.

**5. Environment needs**

1. qaAuto.com
2. Integration Testing Environment.
3. Performance Testing Environment
4. Mobile Testing Environment

**6.Responsibilities**

| Acceptance testing | Savin Roman |
| --- | --- |
| Integration testing | Fedosha Anastasia |
| Writing a Test Cases | Dovgan Olga |
| Unit testing | Perepelitsa Sofia |
| System testing | Tkachenko Denis |
| Preparation of reporting documentation | Whoever draws the short straw |

**7.Risk and contingencies**

1. Lack of personnel resources at the start of testing.
2. Lack of required hardware, software, data or tools.
3. Delays in training on the application and/or tools
4. Changes to the original requirements or designs.
5. Complexities involved in testing the applications
6. The test team will work overtime (this could affect team morale).
7. The scope of the plan may be changed.