



TEST PLAN

RaceTrac mobile automation

Legal Notice: This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM®.

CONFIDENTIAL

Related Artifacts	
Ref.	Name

Abbreviations and Acronyms	

CONTENTS

1	INTRODUCTION	4
2	SCOPE OF WORK.....	4
2.1	COMPONENTS AND FUNCTIONS NOT TO BE TESTED	4
2.2	THIRD-PARTY COMPONENTS.....	4
3	QUALITY AND ACCEPTANCE CRITERIA	4
4	CRITICAL SUCCESS FACTORS	5
5	RISK ASSESSMENT	5
6	RESOURCES	5
6.1	KEY PROJECT RESOURCES	5
6.2	TEST TEAM	6
6.3	TEST ENVIRONMENT	6
6.4	TEST TOOLS.....	6
7	TEST DOCUMENTATION AND DELIVERABLES	6
8	TEST STRATEGY.....	7
8.1	ENTRY CRITERIA.....	7
8.2	TEST METHODS	7
8.3	TEST TYPES	7
8.3.1	Function Testing.....	7
8.3.2	GUI Testing.....	8
8.3.3	Regression testing.....	8
8.3.4	Compatibility testing	8
8.3.5	API testing.....	8
8.3.6	Integration testing	8
8.3.7	Performance testing	9
8.4	TEST LEVELS	9
8.4.1	Smoke Test	9
8.4.2	Critical Path Test.....	9
8.4.3	Extended Test	9
8.5	BUG AND DOCUMENTATION TRACKING	9
8.5.1	Bug Severity Definitions	9
9	TESTING SCHEDULE	9

1 INTRODUCTION

This document describes the approach and methodologies used by the testing team to plan, organize and perform the testing applications of the *RaceTrac company*.

2 SCOPE OF WORK

Components and Functions to be Tested#	Application/component name	Function name	Reference
	Native Android Application	According to backlog and user stories	https://tfsappprd01.corp.racetrac.com/RaceTrac/RaceTracPortfolio/_backlogs/backlog/EPA M Mobile Team/Stories
	Native iOS Application	The same as previous	

2.1 COMPONENTS AND FUNCTIONS NOT TO BE TESTED

#	Application/component name	Function name	Reference/Comment
	Native Android Application	TBU	
	Native iOS Application	TBU	

2.2 THIRD-PARTY COMPONENTS

#	Component name	Component role	Reference/Comment
1	Facebook auth	Provides access to application through itself API	Not applicable for automation
2	AppleID auth	Provides access to application via Apple API	Not applicable for automation, hard to implement
3	Twitter auth	Removed from scope	

3 QUALITY AND ACCEPTANCE CRITERIA

- The product should work according to the requirements and functional specification listed at sections [Scope of work](#), [References](#);

- The product bug level should reach the acceptance criteria defined in Contract/Requirements, or if other is not specified, the product should not have bugs with severity Critical and Major to be released for production.

4 CRITICAL SUCCESS FACTORS

- Meet a schedule and complete development and testing of all functionality in term.
- Support multi-language interface.
- Application shouldn't have known bugs with severity Critical and Major at the time of Final Release.
- Functional requirements do not have last minute changes.

5 RISK ASSESSMENT

All project risks including testing ones are tracked and monitored in *DHM*, module Risks.]

#	Risk	Probability	Impact	Mitigation / Contingency Plans
1	Tight schedule Due to the tight schedule the risk of lacking features and/or quality is remarkably high.	High	High	Contingency plan: <ul style="list-style-type: none">- Add resources (only limited possibility)- Extend schedule- Prioritize features and bug tickets
2	Disruption through other projects If dedicated resources are forced to focus on other tasks and projects, due to ad-hoc changes in prioritization, this will have a major impact on time and quality of the product.	Medium	High	Contingency plan: <ul style="list-style-type: none">- Avoid any 'disturbing' tasks (A strict prioritization should be done beforehand)
3	Attrition of key team members	Medium	Medium	Prepare detailed knowledge transfer documents. Developers should work in teams of two, e.g. for code reviews, paired testing, etc.
4	Change in scope and frequent change in requirements	High	High	Structured change management process to be laid down to ensure accurate impact assessment and planning
5	Availability of test environments	Medium	High	Ensure the availability of all needed test environments throughout the project. Planned downtime and maintenance should be communicated with the project team in order to find a solution with the least impact.

6 RESOURCES

6.1 KEY PROJECT RESOURCES

#	Project Role	Name, e-mail, location
	Delivery Manager	Uladzislau Zhurkevich Office: +375 17 389 0100x51928

		Email: Uladzislau_Zhurkevich@epam.com
	Account Manager	Tamajit Sarkar Office: +375 17 389 0100x32931 Email: Tamajit_Sarkar@epam.com

6.2 TEST TEAM

#	Project Role	Name	Location	Responsibilities
	Key Tester	Serhii Suzanskyi	Ukraine	Responsible for creating and maintain Test Automation framework, test artifacts, CI/CD processes
	Tester			

6.3 TEST ENVIRONMENT

Environment name	Environment type (Hardware/Software, etc.)	Owner (EPAM/Client)	Application (for which purposes is it used)
QA environment	TBU		
Dev environment	TBU		

6.4 TEST TOOLS

#	Tool	Comment	Link
	Azure DevOps	Test Management tool	TBU
	Azure DevOps	Bug tracking tool	TBU
	Asure DevOps	Test documentation storage	TBU

7 TEST DOCUMENTATION AND DELIVERABLES

#	Title	Responsible person(s)	Frequency (delivery time)	Method of delivery
	<RaceTrac> TestPlan	Serhii Suzanskyi	Once before the testing start	

	<RaceTrac> TestCases	Serhii Suzanskyi	Before the testing start	
	Bug reports	Serhii Suzanskyi	Upon finding a bug	
	Test Result Reports	Serhii Suzanskyi	Weekly	

8 TEST STRATEGY

The **RaceTrac** applications will be tested using a “black box” approach, which is based on the requirements and functional specification without knowledge of the internal structure or program source code.

8.1 ENTRY CRITERIA

The Testing Team may suspend partial or full-testing activities on a given build if any of the following occurs:

- It is impossible to install successfully the new build following the readme.
- There is a fault with a feature that prevents its testing.
- Application does not contain the specified change(s).
- New claimed functionality doesn't work or works improperly.
- A severe problem has occurred that does not allow testing to continue.
- Development has not corrected the problem(s) that previously suspended testing.
- A new version of the software is available to test.

8.2 TEST METHODS

Testing is the process of attempting to find discrepancies between the program and its functional specification/ requirements. The goal is to make sure that all functions of the **RaceTrac** applications work correctly.

- Manual functional testing - is considered as the main method of the application testing.
- Automated functional testing - smoke test will be performed manually/ automatically using tools described in section [Test Tools](#).
- Automated Performance/Load testing will be performed upon request according to a separate plan specified in section [Test documentation and deliverables](#).

8.3 TEST TYPES

This is a sample list of testing types, to be adapted to the specific project needs and requirements.

8.3.1 Function Testing

Function (feature) testing will be performed to verify that the product functions meet the functional requirement. This testing will be performed on test environments (Dev, QA1) and finally after it is live on production function testing will be performed on Production as well if no constrains are applicable (like necessity to use payment etc).

8.3.2 GUI Testing

The Graphic User Interface testing covers testing of design e.g. labels, text boxes, text, captions, buttons, lists, icons, links and content to correspond mock ups provided by UX team. QA provides both functional testing combined with UI testing. But final story acceptance (of stories that include UI work) is done by UX team member assigned to specific team.

8.3.3 Regression testing

The objective of tests executed in this level is to make sure that migration components did not affect any of the existing functionality that has not been updated or changed. At this level test team will provide quality assessment of X application as whole functional unit. Regression testing will be performed on DEV environment.

8.3.4 Compatibility testing

Compatibility Testing on RaceTrac projects determines whether the product operates correctly with various devices. Testing on small screen resolutions is performed as well as a part of responsive design implementation. Real device testing is not required, can be performed in cases when QA engineer has device available and time permits to perform the testing on real device.

The list of devices that is based on current usage stats (requirements) is below. The device can be added to the list or excluded according to statistics or PO requirement. The highest priority to the most popular devices according to the market statistics for Client apps.

The list of screen resolutions that should be used for testing based on agreement with Key tester, Delivery Manager and UX team (that creates mock-ups for the user stories):

#	Screen resolutions	When to test	Where to test
1	1080 × 1920	Stories testing, bugs verification, regression testing	Emulator Pixel 4 XL
2	1792 × 828	Regression (layout check, perform Smoke test if time permits)	Emulated iPhone 11

8.3.5 API testing

API testing includes verification of work of new and existing endpoints (*list of endpoints*) in three directions: verification of documentation for the endpoints, validation of work of the endpoints against business rules, verification of the endpoints against API version. API testing presupposes checking API returns values based on the input condition, verifying that API doesn't return anything at all or the wrong results, verifying if the API triggers some other event or calls another API, verifying if the API is updating any data structures.

Testing is performed based on Swagger documentation or stories acceptance criteria mostly using the Postman tool. Developer tools in browser are also used for requests/responses review.

Note API testing is not in scope for mobile automation team.

8.3.6 Integration testing

Integration testing is performed to test the modules/components integrated work as expected i.e. to test the modules that are working fine individually do not have issues when integrated. Integration testing on RaceTrac project is performed for

- API integration with front end (modules also interact with some third-party tools or APIs that also need to be tested, so that the data accepted by that API / tool is correct and that the response generated is also as expected).

- All the components implemented by each team all together as a system, going through all the scenarios.

8.3.7 Performance testing

Performance testing is performed by Automation team and manual QA. New tests are created only when requested and specific scenarios are defined.

8.4 TEST LEVELS

8.4.1 Smoke Test

Smoke Test is performed to quickly assess the readiness of the product for further more deep and thorough testing. It includes testing *RaceTrac* applications major functions on the one most often used and consequently most important server/ client configuration.

Smoke Test is performed under *iOs*, *Android* client platforms at *AsureDevOps Node*. Typical Smoke Test Scenario would include the steps described in the “Smoke Test” section of the *RaceTrac* TestCases for each application which are referred in section [Test documentation and deliverables](#).

If Smoke Test failed, Testing Team sends notification and suspends testing until corrected version of the product is available.

8.4.2 Critical Path Test

Critical Path Test will be performed after Smoke Test is passed. The goal of the Critical Path Test is to find bugs that could affect the major functionality of the application that is most important for the product users. Critical Path Test will be performed manually according to *RaceTrac* Test Cases document on all platforms to be certified.

8.4.3 Extended Test

The *Extended Test*’s goal to find bugs related to the non-typical but still possible and likely usage scenarios (e.g. entering the incorrect data into the fields, boundary testing and so on). Extended Test will be performed both according to test cases and using ad hoc testing scenarios.

- Detailed Test Results Report containing testing process description and results summary will be issued as it specified in section [Test documentation and deliverables](#).

8.5 BUG AND DOCUMENTATION TRACKING

Tools described in the section [Test Tools](#) will be used for bug reporting and documentation tracking. The bug metrics and statistics will be included in the test results reports.

8.5.1 Bug Severity Definitions

Critical - Application, component or module crash or are not accessible

Major - Data corruption/loss, a problem in major functionality, no workaround is known.

Medium - A problem with workaround, secondary features do not work properly

Minor - Cosmetic flaw

9 TESTING SCHEDULE

#	Activity	Begin Date	End Date	Assignment	Location	Work content
	Test plan creation	20 March 2021		Serhii Suzanskyi	Ukraine	Up to 5 days

	Test cases creation	TBU	TBU			
	Build installation	TBU	TBU			
	Smoke Test execution	TBU	TBU			
	Critical path Test execution	TBU	TBU			

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
0.1	Prepared common documentation, artifacts and dedline	Serhii Suzanskyi	23 March 2021		