



RIDERTRACK

Acceptance test plan Document

Index

Revision History	4
1. Introduction	5
1.1 Purpose of the document	5
1.2 Document Organization	5
1.3 Intended Audience	5
1.4 Definition and Acronyms	5
1.4.1 Definitions	5
1.4.2 Acronyms and Abbreviations	6
1.5 References	7
2. Testing processes	8
2.1 Automated tests	8
2.2 Manual tests	8
3. Test cases	9
3.1 Registration	9
3.1.1 Classic registration	9
3.1.2 Classic registration with already used data	9
3.1.3 Social registration	10
3.1. Social registration without permission	10
3.2 Login	11
3.2.1 Classic login	11
3.2.2 Login with wrong credentials	11
3.2.2 Social login	12
3.3 Logout	12
3.4 Event View	13
3.4.1 Event list and Event Details	13
3.4.2 Organized/Enrolled events view	13
3.5 Progress Tracking	14
3.5.1 Progress visualization on a map	14
3.5.2 Progress visualization on a map (through URL directly)	14
3.6 Event Creation and Management	15
3.6.1 Event Creation	15
3.6.2 Create event without logo	15
3.6.3 Create event without filling all the required fields of the form	16
3.6.4 Edit event	16
3.6.5 Delete event	17
	2

3.6.6 Edit event's route	18
3.6.7 Managing registration of an event	18
3.7 Tracking	19
3.7.1 Start tracking	19
3.7.1 Start tracking when the Registration Period is still open	19
3.7.2 Stop tracking	20
3.8 Event Enrollment	20
3.8.1 Make enrollment	20
3.8.1 Make enrollment before registration period is setted	21
3.8.2 Withdraw enrollment	21
3.9 Archive	22
3.9.1 Details about past enrolled events	22
3.9.2 Details about past organized events	22
3.10 Mobile Application Interaction	23
3.10.1 Share on Facebook real-time progress	23
3.11 Personal Data Management	23
3.11.1 Delete user account	23
3.11.2 Delete user account while still being an event organiser	24
3.12 Support contacting	24
3.12.1 Send email to system admin	24
3.13 Mobile Application Login	25
3.13.1 Successful classic login	25
3.13.2 Unsuccessful classic login	25
3.14 Start sending tracking data from mobile app	26
3.14.1 Start sending tracking data of an event not already begun	26
3.14.1 Start sending tracking data of an event already begun	26
3.15 Change timeout between sending of two consequent position	27
3.16 Logout from the mobile app	27
3.17 System administrator	28
3.17.1 Users overview for system administrator	28
3.17.2 Modification of user	28
3.17.3 Events overview for system administrator	29
3.17.4 Modification of event	29
4. Test coverage	30

Revision History

Date	Version	Description	Responsible
9/12/2017	0.1	Initial draft	Alessandro Caprarelli
22/12/2017	1.0	First versions	Alessandro Caprarelli
3/1/2017	1.1	Fixed layout and minor mistakes	Alessandro Caprarelli
4/1/2017	1.2	Changed the outcome of creation of event according to new choices	Alessandro Caprarelli
5/1/2017	1.3	Added info about script to simulate users and questionnaire to test usability.	Alessandro Caprarelli

1. Introduction

1.1 Purpose of the document

The *Acceptance Test Document* is intended for describing the test procedures for “**Rider Track**” system. The objective of this is to verify that the system matches and satisfies the relative requirements and user stories presented in the *Requirements and Specification Document*. The result of the testing procedure that will be proposed can give stakeholder and clients the confidence on how the final product will perform.

1.2 Document Organization

This document is essentially structured in four parts:

1. In this **first section**, it is presented an overview of the document content and scope.
2. In the **second section**, it is described the procedure used to test the system.
3. In the **third section**, it is provided a series of test cases that can be reproduced following the specified steps.
4. Finally in the **fourth section**, it is shown the coverage of the test cases compared to the requirements of the system.

1.3 Intended Audience

The target audience of this document includes:

- **Project customer:** to check if the work planned by the team is coherent with his requirements and to agree upon the major objectives.
- **Developers:** to allow the developers to understand the project and to work individually with a unique shared idea of the goal to be reached.
- **Stakeholders:** to have information about testing policies and processes.

1.4 Definition and Acronyms

1.4.1 Definitions

<i>Keyword</i>	<i>Definition</i>
Classical registration	Registration process done using email and password.

Classical login	Login process done using email and password.
Social registration	Registration done using social accounts such as Facebook and Google.
Social login	Login done using social accounts such as Facebook and Google.
User	User is intended as a person that uses the Website
Travis-CI	Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub
Mongo DB	Database of “RiderTrack” system

1.4.2 Acronyms and Abbreviations

Abbreviation/Acronym	Definition
FR i.j	Functional Requirement (section “i” and subsection “j”)
US i.j	User Story (section “i” and subsection “j”)
AT i	Acceptance test
Sys admin	System administrator
ORM	Object-Relational Mapping
REST	RESTful Web Services
API	Application Programming Interface

1.5 References

- “Requirements definition” , Alessandro Caprarelli, Marzia Degiorgi, Mariano Etchart, Giulia Leonardi, Josip Mališa, Ante Breščić, Ivan Kvesić, 2017.

- *“Project Vision and Plan”* , Alessandro Caprarelli, Marzia Degiorgi, Mariano Etchart, Giulia Leonardi, Josip Mališa, Ante Breščić, Ivan Kvesić, 2017.
- *“Design Description Document”* , Alessandro Caprarelli, Marzia Degiorgi, Mariano Etchart, Giulia Leonardi, Josip Mališa, Ante Breščić, Ivan Kvesić, 2017.
- SCORE project detail: <http://score-contest.org/2018/projects/ridetrack.php>

2. Testing processes

During the development of Ridertrack both automated and manual testing have been done by team's members to guarantee the maximum affidability of the system.

Furthermore, a questionnaire with questions about usability has been given to potential real users in order to test user-friendliness and usability in different browsers, platform and devices. The sample of users was random. The results will be released in the acceptance test report document.

2.1 Automated tests

These tipology of tests are used for the system backend, that is based on a node.js application. Here a set of unit and integration tests have been defined, that can be run using Mocha.js.

The unit tests check that the Mongoose, an ORM for MongoDB, models respect the rules of the data models defined in the design document.

The integration tests check that the REST endpoints offered by the node.js app behave as expected and written in the API documentation.

All the tests, around 50 at the time of writing, are run at every push by Travis-CI granting anytime a complete coverage of new updates added in the system. In fact if and only if all the tests pass, it deploys the application to Heroku.

2.2 Manual tests

The team has checked all the implemented features using manual testing. Each feature has been tested firstly by team's members who has implemented it and then by the other team members to ensure quality of manual testing.

Moreover, a script that simulates real users has been done in order to test functionalities that involve real time data, such as collection of data and real time progress page of event. The script, once started, enrolls and starts sending positions about random users in order to simulate a real race. Fake data has been generated using

<http://www.mockaroo.com/>. Positions are generated along the route in order to make them as real as possible.

3. Test cases

3.1 Registration

3.1.1 Classic registration

Test ID	AT1
Test description	Registration using classical registration
User Story ID	US 1
Requirement ID	FR 1.1, FR 1.2
Precondition	The email has never been registered and it's not related to any accounts registered using social authentications.
Test procedure	<ol style="list-style-type: none">1. User opens website2. User clicks on the "Register" button in the navbar3. User correctly (valid email address, matching passwords) fills all the fields of the form4. User click "Register" button at the bottom of the form
Expected result	User is successfully registered. "My events" page is shown and it does not contains any events.

3.1.2 Classic registration with already used data

Test ID	AT2
Test description	Registration using classical registration with an already registered email
User Story ID	US 1
Requirement ID	FR1.1, FR1.2
Precondition	The email has been already registered either with classical registration or it is associated with a social registration
Test procedure	<ol style="list-style-type: none">1. User opens website2. User clicks on the "Register" button in the navbar

	<ol style="list-style-type: none"> 3. User correctly (valid email address, matching passwords) fills all the fields of the form 4. User click "Register" button at the bottom of the form
Expected result	User registration is rejected. An error in the registration form is shown.

3.1.3 Social registration

Test ID	AT3
Test description	Registration using social registration
User Story ID	US 1
Requirement ID	FR1.3,FR1.4
Precondition	The email associated to the social account has not been already registered either with classical registration or with another social account
Test procedure	<ol style="list-style-type: none"> 1. User opens website 2. User clicks on the "Register" button in the navbar 3. User clicks on either "Facebook" or "Google" button at the bottom of the registration form 4. User gives permission to the app in the Facebook dialog
Expected result	User is successfully registered. "My events" page is shown and it does not contains any events.

3.1. Social registration without permission

Test ID	AT4
Test description	Permissions not given during social registration
User Story ID	US 1
Requirement ID	FR 1.3,FR 1.4
Precondition	-

Test procedure	<ol style="list-style-type: none"> 1. User opens website 2. User clicks on the “Register” button in the navbar 3. User clicks on either “Facebook” or “Google” button at the bottom of the registration form 4. In the auth page of the social service, the user does not give the requested permissions.
Expected result	<p>User registration is rejected.</p> <p>An error in the registration form is shown.</p>

3.2 Login

3.2.1 Classic login

Test ID	AT5
Test description	Login using classical login
User Story ID	US 5, US 12
Requirement ID	FR 6.1, FR 6.4, FR 14.1
Precondition	The email has been already registered with classical registration and user is not currently logged in.
Test procedure	<ol style="list-style-type: none"> 1. User opens website 2. User clicks on the “Login” button in the navbar 3. User correctly (valid email address, correct password) fills all the fields of the form 4. User click “Login” button at the bottom of the form
Expected result	<p>User is successfully logged in.</p> <p>“My events” page is shown.</p>

3.2.2 Login with wrong credentials

Test ID	AT6
Test description	Login with wrong credentials (email or password)
User Story ID	US 5, US12
Requirement ID	FR 6.4, FR 14.1

Precondition The user credentials do not exist in database or the wrong password is entered.

Test procedure

1. User opens website
2. User clicks on the “Login” button in the navbar
3. User incorrectly (wrong email or password) fills one of the fields of the form
4. User click “Login” button at the bottom of the form

Expected result User is not able to log in and error message is shown.
User is prompt to re-enter his credentials.

3.2.2 Social login

Test ID AT7

Test description Login using social login

User Story ID US5

Requirement ID FR 6.2, FR 6.3

Precondition The email associated to the social account has already been registered with social registration.

Test procedure

1. User opens website
2. User clicks on the “Login” button in the navbar
3. User clicks on either “Facebook” or “Google” button at the bottom of the login form

Expected result User is successfully logged in.
“My events” page is shown.

3.3 Logout

Test ID AT8

Test description Logout

User Story ID US5, US12

Requirement ID FR 6.5, FR 14.2

Precondition The user is already logged in the website with one of the methodology allowed.

Test procedure	<ol style="list-style-type: none"> 1. User clicks on the user icon in the navbar 2. User selects the logout option
Expected result	User is successfully logged out. The public home page is shown.

3.4 Event View

3.4.1 Event list and Event Details

Test ID	AT9
Test description	Visualize the event list and the details of an event
User Story ID	US 2, US 6
Requirement ID	FR 2.1, FR 2.2, FR 7.1, FR 7.2
Precondition	At least one event exists.
Test procedure	<ol style="list-style-type: none"> 1. User opens the website. 2. User clicks on “All events” in the navbar of the website. 3. User filter results adding the category “RUNNING”. 4. User selects an event.
Expected result	The event detail page is shown with all the details provided by the organizer at that time.

3.4.2 Organized/Enrolled events view

Test ID	AT10
Test description	Visualize the list of my enrolled events and my organised events
User Story ID	US 8.4, US 8.6
Requirement ID	FR 10.7, FR 10.9, FR 10.11
Precondition	User is logged in.
Test procedure	<ol style="list-style-type: none"> 1. User opens the website. 2. User clicks on “My events” in the navbar of the website.

Expected result User sees a tab with his/her enrolled events and organized events and can choose between them.

3.5 Progress Tracking

3.5.1 Progress visualization on a map

Test ID	AT11
Test description	Visualize the progress of an event that is “Ongoing”
User Story ID	US 3, US 9.3
Requirement ID	FR 4.1, FR 4.2, FR 11.1, FR 11.2, FR 11.3
Precondition	The event should be started by the organizer
Test procedure	<ol style="list-style-type: none"> 1. User opens the website. 2. User selects the event that want to follow in “All events” section or in the “Home page”. 3. User click on the “map” button.
Expected result	The “progress” page is shown with the route of the selected event, the participants progress are shown within the route. Also click on a participant further information are displayed.

3.5.2 Progress visualization on a map (through URL directly)

Test ID	AT12
Test description	Visualize the progress of an event which has no route set through an URL
User Story ID	US 3, US 9.3
Requirement ID	FR 4.1, FR 4.2, FR 11.1, FR 11.2, FR 11.3
Precondition	The event exist and has no route set. User knows the event Id of the events he want to see.
Test procedure	<ol style="list-style-type: none"> 1. User opens the website through URL with manually writing the event ID/progress/map. 4. User selects the “map” tab.
Expected result	The “progress” page is shown without the route of the selected event, the participants progress are shown within

the route. User can see a message “The route is not available yet” and has an option to go back to the event details page.

3.6 Event Creation and Management

3.6.1 Event Creation

Test ID	AT13
Test description	Create an event
User Story ID	US 7.1
Requirement ID	FR 9.1, FR 9.2
Precondition	User is already logged in.
Test procedure	<ol style="list-style-type: none">1. User goes to “My events” page2. User clicks on the “Create event”3. User correctly fills all the fields (name, type, date, county, city, logo) of the form4. User clicks on “Create” button at the end of the form
Expected result	New event is successfully created. Events/event:id/manage page is shown.

3.6.2 Create event without logo

Test ID	AT14
Test description	Create event without logo
User Story ID	US 7.2, US 8.4
Requirement ID	FR 9.3, FR 9.5
Precondition	User is already logged in and wants to edit an event he/she already created.
Test procedure	<ol style="list-style-type: none">1. User goes to “My events” page2. User clicks on the “Create event”3. User correctly fills all the fields (name, type, date, county, city) of the form except logo4. User clicks on “Create” button at the end of the form

Expected result The event is created with a default logo (Ridertrack logo).

3.6.3 Create event without filling all the required fields of the form

Test ID	AT15
Test description	Create event without filling all the required form fields
User Story ID	US 7.2, US 8.4
Requirement ID	FR 9.3, FR 9.5
Precondition	User is already logged in and wants to edit an event he/she already created.
Test procedure	<ol style="list-style-type: none">1. User goes to “My events” page2. User clicks on the “Create event”3. User doesn’t fill all the required fields of the form (name, type, date, county, city)4. User clicks on “Create” button at the end of the form
Expected result	“Create” button is unavailable. An event can’t be created.

3.6.4 Edit event

Test ID	AT16
Test description	Edit an existing event
User Story ID	US 7.2, US 8.4
Requirement ID	FR 9.3, FR 9.5
Precondition	User is already logged in and wants to edit an event he/she already created.
Test procedure	<ol style="list-style-type: none">1. User clicks on “My events” tab .2. User clicks on the “Organized” tab in my-events page3. User clicks on the “Edit event” button of the event he/she wants to edit4. User correctly fills the fields of the form which he wants to edit

	5. User clicks on the “Update” button in events/event:id/manage page
Expected result	An event is successfully edited. Events/event:id page is shown.

3.6.5 Delete event

Test ID	AT17
Test description	Delete an existing event
User Story ID	US 7.3, US 8.4
Requirement ID	FR 9.4, FR 9.5
Precondition	User is already logged in and have already created an event.
Test procedure	<ol style="list-style-type: none"> 1. User goes on “My events” . 2. User clicks on the “Organized” tab in my-events page. 3. User opens the event that wants to delete. 4. User selects the “Delete” button in event detail page. 5. User confirms.
Expected result	The event is deleted and not shown anymore in the event list.

3.6.6 Edit event’s route

Test ID	AT18
Test description	Edit a route of an event
User Story ID	US 7.4, US 8.4
Requirement ID	FR 9.2
Precondition	User is already logged in and wants to edit a route of an event he/she already created.
Test procedure	<ol style="list-style-type: none"> 1. User opens website 2. User clicks on the “Organized” tab in my-events page 3. User clicks on the “Edit event” button of the event he/she wants to edit 4. User clicks on the “Edit route” button on the event manage page

	<ol style="list-style-type: none"> 5. User adds the new route 6. User clicks on the “Save” button in events/event:id/manage/route page 7. User clicks on the “OK” button of the confirmation alert pop out
Expected result	An event route is successfully edited. Events/event:id/manage page is shown.

3.6.7 Managing registration of an event

Test ID	AT19
Test description	Open/Close the registration to an event
User Story ID	US 7.5, US 7.6, US 8.4
Requirement ID	FR 9.6, FR 9.5
Precondition	User is already logged in and he/she has already created an event without adding the registration period.
Test procedure	<ol style="list-style-type: none"> 1. User goes to “My events” page 2. User selects the organized event option 3. User click on the “Edit” button of the event that wants to modify 4. User fills up the enrollment dates 5. User updates the event clicking on the “Update” button
Expected result	The event is correctly updated and the registration period is set to manage enrollment automatic controls. Events/event:id page is shown.

3.7 Tracking

3.7.1 Start tracking

Test ID	AT20
Test description	Start the tracking of an event
User Story ID	US 7.7, US 8.4
Requirement ID	FR 9.7, FR 9.5

Precondition User is already logged in and he/she has already created an event. The registration period is closed and the event is not yet started.

Test procedure

1. User goes in “My events” page
2. User selects organized event tab
3. User clicks on “Start event” on the event that wants to start

Expected result The event changes the state to “Ongoing” and the user is redirected to the “Progress” page.

3.7.1 Start tracking when the Registration Period is still open

Test ID AT21

Test description Start the tracking of an event when the registration period is still open

User Story ID US 7.7, US 8.4

Requirement ID FR 9.7, FR 9.5

Precondition User is already logged in and he/she has already created an event. The registration period is open.

Test procedure

1. User clicks “My events” in the navbar of the website
2. User clicks “Start Tracking” button on an event organized

Expected result A window message is shown with an alert. The event status remain unchanged.

3.7.2 Stop tracking

Test ID AT22

Test description Stop the tracking of an event

User Story ID US 7.8, US 8.4

Requirement ID FR 9.7, FR 9.5

Precondition User is already logged in and he/she has already created an event. The registration period is closed and the event is already started.

Test procedure	<ol style="list-style-type: none"> 1. User goes on “My events” page 2. User selects organized events tab 3. User clicks on “Stop event” button on the event that wants to stop
Expected result	The event changes the state to “Passed”. From the “Progress” page the tracking is not accessible anymore. While the user stays in the same page.

3.8 Event Enrollment

3.8.1 Make enrollment

Test ID	AT23
Test description	User enrolls in an event
User Story ID	US 8.1, US 8.6, US 8.7
Requirement ID	FR 10.1, FR 10.10
Precondition	User is already logged in and the registration period of the event selected is specified and open.
Test procedure	<ol style="list-style-type: none"> 1. User goes on “All events” page choosing it in the navbar of the website 2. User chooses the event in which he/she wants to enroll and selects it 3. User clicks on enroll button
Expected result	The User receives a confirmation message and can see the events in which he/she has enrolled in “My events” page.

3.8.1 Make enrollment before registration period is setted

Test ID	AT24
Test description	User enrolls in an event
User Story ID	US 8.1, US 8.6, US 8.7
Requirement ID	FR 10.1, FR 10.10
Precondition	User is already logged in and the registration period is not opened or not setted.

Test procedure	<ol style="list-style-type: none"> 1. User goes on “All events” page choosing it in the navbar of the website 2. User chooses the event in which he/she wants to enroll and selects it 3. User clicks on enroll button
Expected result	A window with an alert message is opened. The user cannot enroll in the event yet.

3.8.2 Withdraw enrollment

Test ID	AT25
Test description	Withdraw enrollment in an event
User Story ID	US 8.3, US 8.6, US 8.7
Requirement ID	FR 10.3, FR 10.10
Precondition	User is already logged in and he/she has already enrolled in the event.
Test procedure	<ol style="list-style-type: none"> 1. User goes in “My events” page 2. User selects “Enrolled” tab 3. User chooses the event in which he/she wants to withdraw enrollment 4. User clicks on “withdraw enrollment” button and confirm
Expected result	The User is removed from the event as a participant and he/she cannot see the event anymore in “My events” page

3.9 Archive

3.9.1 Details about past enrolled events

Test ID	AT26
Test description	See details about past events in which the user was enrolled
User Story ID	US 8.6, US 8.7, US 8.10
Requirement ID	FR 10.8, FR 10.9

Precondition	User is already logged in and he/she has previously participate in an event.
Test procedure	<ol style="list-style-type: none"> 1. User goes in “My events” page 2. User selects Enrolled tab 3. User selects the event that is passed
Expected result	The User is redirect to the event detail page of past events and can see all the information about it.

3.9.2 Details about past organized events

Test ID	AT27
Test description	Display list of event created
User Story ID	US 8.4, US 8.5, US 8.10
Requirement ID	FR 10.11, FR 10.12
Precondition	User is already logged in and he/she has previously created events that can be passed or planned or ongoing.
Test procedure	<ol style="list-style-type: none"> 1. User goes in “My events” page 2. User selects organized events
Expected result	The User can see future, ongoing and past events ordered by date.

3.10 Mobile Application Interaction

3.10.1 Share on Facebook real-time progress

Test ID	AT28
Test description	Share on facebook real-time progress
User Story ID	US 10.3
Requirement ID	FR 12.3
Precondition	User is already logged in, the event status is ongoing and he/she is sending tracking data.
Test procedure	<ol style="list-style-type: none"> 1. User clicks button “Share status”

Expected result Details about the user's progresses are instantly shared on his/her Facebook profile.

3.11 Personal Data Management

3.11.1 Delete user account

Test ID	AT29
Test description	Delete user account
User Story ID	US 11.2
Requirement ID	FR 13.3
Precondition	User is already logged in.
Test procedure	<ol style="list-style-type: none">1. User opens website2. User clicks on the "My Profile" button in the navbar3. User clicks on the "delete Account" button
Expected result	User is prompted to confirm the deletion of his account. With the click confirm user account is successfully deleted in the database. After delete guest is redirected to public home page.

3.11.2 Delete user account while still being an event organiser

Test ID	AT30
Test description	Delete user account while still being the event organiser
User Story ID	US 11.2
Requirement ID	FR 13.3
Precondition	User is already logged in.
Test procedure	<ol style="list-style-type: none">1. User opens website2. User clicks on the "My Profile" button in the navbar3. User clicks on the "delete Account" button
Expected result	User is prompted to confirm the deletion of his account. With the click confirm user account is unsuccessfully deleted and is shown the following message "You cannot delete your account because you have created events".

3.12 Support contacting

3.12.1 Send email to system admin

Test ID	AT31
Test description	Contact the system support
User Story ID	US4.2
Requirement ID	FR5.2
Precondition	-
Test procedure	<ol style="list-style-type: none"> 1. User opens website 2. User clicks on the “Contacts” button in the navbar 3. User fills the form with the info and a message he/she wants to send to the system support 4. Users clicks on the “Send message” button.
Expected result	<p>The message is sent to the system support.</p> <p>The user get notified (“Your message has been successfully sent!”).</p>

3.13 Mobile Application Login

3.13.1 Successful classic login

Test ID	AT32
Test description	Successful log into mobile app
User Story ID	US 5.1
Requirement ID	FR 6.1, FR 6.4
Precondition	User is already registered and has already installed Ridertrack mobile app in his/her smartphone.
Test procedure	<ol style="list-style-type: none"> 1. User opens Ridertrack app in his/her smartphone . 2. User clicks on “Login” button in the screen appeared. 3. User correctly (inserting valid email and correct password) fills the login form .

	4. User clicks “Sign in” button.
Expected result	After a short loading, “Your next events” screen appears, showing to the user the list of all the upcoming events in which he/she is enrolled.

3.13.2 Unsuccessful classic login

Test ID	AT33
Test description	Log into the mobile app with wrong credentials (email or password)
User Story ID	US 5.1
Requirement ID	FR 6.1, FR 6.4
Precondition	User is already registered and has already installed Ridertrack mobile app in his/her smartphone.
Test procedure	<ol style="list-style-type: none"> 1. User opens Ridertrack app in his/her smartphone . 2. User clicks on “Login” button in the screen appeared. 3. User incorrectly (inserting wrong email and password) fills the login form . 4. User clicks “Sign in” button.
Expected result	User is not able to log in and error message is shown. User is prompt to try again the login procedure.

3.14 Start sending tracking data from mobile app

3.14.1 Start sending tracking data of an event not already begun

Test ID	AT34
Test description	Launch the tracking on the mobile app
User Story ID	US 9.1
Requirement ID	FR 11
Precondition	User is already logged in the mobile app (the current screen is “Your next events”) and enrolled in the event yet to start.
Test procedure	<ol style="list-style-type: none"> 1. User clicks on the upcoming event, in which he/she wants to be tracked.

Expected result A screen with chronometer, speed in km/h and distance from the finish line appears. All the counters are set to zero. Only when the organizer decrees the actual starting of the race, the counters show the real parameters.

3.14.1 Start sending tracking data of an event already begun

Test ID	AT35
Test description	Launch the tracking on the mobile app
User Story ID Requirement ID	US 9.1 FR 11
Precondition	User is already logged in the mobile app (the current screen is “Your next events”) and enrolled in the event. The event has already been started by the organizer.
Test procedure	<ol style="list-style-type: none"> 1. User clicks on the event, in which he/she wants to be tracked.
Expected result	A screen with chronometer, speed in km/h and distance from the finish line appears. The counters show the parameters real time. The chronometer marks time passed from the launch of the tracking.

3.15 Change timeout between sending of two consequent position

Test ID	AT36
Test description	Change timeout between sending of two consequent positions from mobile app.
User Story ID Requirement ID	US 9.6 FR 11.7
Precondition	User is already logged in the mobile app (current screen is “Your next events”).
Test procedure	<ol style="list-style-type: none"> 1. User clicks on the menu icon on the top right of the screen. 2. User clicks on the option “Data Sender” in the scroll down menu appeared.

	<ol style="list-style-type: none"> 3. User clicks on the button corresponding to the chosen timeout. 4. User clicks on “Set timeout” button.
Expected result	User is redirected to “Your next events” screen and the application will send tracking data during all the upcoming events according to the new timeout.

3.16 Logout from the mobile app

Test ID	AT37
Test description	Logout from Ridertrack mobile application.
User Story ID	US 5.2
Requirement ID	FR 6.5
Precondition	User is already logged in the mobile app (current screen is “Your next events”).
Test procedure	<ol style="list-style-type: none"> 1. User clicks on the menu icon on the top right of the screen. 2. User clicks on the option “Logout” in the scroll down menu appeared.
Expected result	User is successfully logged out. The login page is shown.

3.17 System administrator

3.17.1 Users overview for system administrator

Test ID	AT38
Test description	Users list in system administrator panel.
User Story ID	US 14.1
Requirement ID	FR 16.1
Precondition	User, with role ‘admin’, is already logged in the web site.
Test procedure	<ol style="list-style-type: none"> 1. User clicks on the “Manage users” link in the navbar.
Expected result	The users table is shown.

3.17.2 Modification of user

Test ID	AT39
Test description	Modification of user detail by sys admin.
User Story ID	US 14.2
Requirement ID	FR 16.2
Precondition	User, with role 'admin', is already logged in the web site.
Test procedure	<ol style="list-style-type: none">1. User clicks on the "Manage users" link in the navbar.2. User clicks on the row of the user to modify. A popup is shown with a form with data already filled in.3. User changes the data.4. User clicks on "Save" button.
Expected result	The popup closes with no error messages.

3.17.3 Events overview for system administrator

Test ID	AT40
Test description	Events list in system administrator panel.
User Story ID	US 13.1
Requirement ID	FR 15.1
Precondition	User, with role 'admin', is already logged in the web site.
Test procedure	<ol style="list-style-type: none">1. User clicks on the "Manage events" link in the navbar.
Expected result	The events table is shown.

3.17.4 Modification of event

Test ID	AT41
Test description	Modification of user detail by sys admin.
User Story ID	US 13.2
Requirement ID	FR 15.2
Precondition	User, with role 'admin', is already logged in the web site.

Test procedure	<ol style="list-style-type: none"> 1. User clicks on the “Manage events” link in the navbar. 2. User clicks on the row of the event to modify. A popup is shown with a form with data already filled in. 3. User changes the data. 4. User clicks on “Save” button.
Expected result	The popup closes with no error messages.

4. Test coverage

All the functional requirements are covered by at least one test case.

The following table maps each functional requirements with the test cases.

<i>Functional requirement*</i>		<i>Test case</i>
FR1	Registration	AT1, AT2, AT3, AT4
FR2	Spectators events view	AT9
FR4	Spectators progress tracking	AT11, AT12
FR5*	Contact the system Support	AT31
FR6	Login	AT5, AT6, AT7, AT8, AT32, AT33, AT34, AT37
FR7	Users events view	AT9
FR9	Event management	AT13, AT14, AT15, AT16, AT17, AT18, AT19, AT20, AT21, AT22
FR10	Event enrollment	AT10, AT23, AT24, AT25, AT26, AT27
FR11	Users progress tracking	AT11, AT12, AT34, AT35, AT36
FR12	Interaction	AT28
FR13	Personal data management	AT29, AT30
FR14	Sysadmin login	AT5, AT6, AT8

FR15	<i>Sysadmin events management</i>	AT40, AT41
FR16	<i>Sysadmin users management</i>	AT38, AT39

***Note:** the missing functional requirements (FR3, FR5, FR8) were dropped and not implemented at all. Refer to “Requirements definition” and “Project vision and plan” for more details.