



# RIDERTRACK

*Acceptance test plan Document*

# Index

<b>Revision History</b>	<b>4</b>
<b>1. Introduction</b>	<b>5</b>
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
<b>2. Testing processes</b>	<b>8</b>
2.1 Automated tests	8
2.2 Manual tests	8
<b>3. Test cases</b>	<b>9</b>
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
<b>4. Test coverage</b>	<b>30</b>

# 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

# 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>
<b>Classical registration</b>	Registration process done using email and password.

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

#### 1.4.2 Acronyms and Abbreviations

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

### 2.1 Automated tests

These topology 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.



## 3. Test cases

### 3.1 Registration

#### 3.1.1 Classic registration

<b>Test ID</b>	AT1
<b>Test description</b>	Registration using classical registration
<b>User Story ID</b>	US 1
<b>Requirement ID</b>	FR 1.1, FR 1.2
<b>Precondition</b>	The email has never been registered and it's not related to any accounts registered using social authentications.
<b>Test procedure</b>	<ol style="list-style-type: none"><li>1. User opens website</li><li>2. User clicks on the "Register" button in the navbar</li><li>3. User correctly (valid email address, matching passwords) fills all the fields of the form</li><li>4. User click "Register" button at the bottom of the form</li></ol>
<b>Expected result</b>	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

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

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

### 3.1.3 Social registration

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

### 3.1. Social registration without permission

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

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

## 3.2 Login

### 3.2.1 Classic login

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

### 3.2.2 Login with wrong credentials

<b>Test ID</b>	AT6
<b>Test description</b>	Login with wrong credentials (email or password)
<b>User Story ID</b>	US 5, US12
<b>Requirement ID</b>	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.

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

### 3.4 Event View

#### 3.4.1 Event list and Event Details

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

#### 3.4.2 Organized/Enrolled events view

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

**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

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

<b>Test ID</b>	AT12
<b>Test description</b>	Visualize the progress of an event which has no route set through an URL
<b>User Story ID</b>	US 3, US 9.3
<b>Requirement ID</b>	FR 4.1, FR 4.2, FR 11.1, FR 11.2, FR 11.3
<b>Precondition</b>	The event exist and has no route set. User knows the event Id of the events he want to see.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User opens the website through URL with manually writing the event ID/progress/map.</li> <li>4. User selects the “map” tab.</li> </ol>
<b>Expected result</b>	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

<b>Test ID</b>	AT13
<b>Test description</b>	Create an event
<b>User Story ID</b>	US 7.1
<b>Requirement ID</b>	FR 9.1, FR 9.2
<b>Precondition</b>	User is already logged in.
<b>Test procedure</b>	<ol style="list-style-type: none"><li>1. User goes to “My events” page</li><li>2. User clicks on the “Create event”</li><li>3. User correctly fills all the fields (name, type, date, county, city, logo) of the form</li><li>4. User clicks on “Create” button at the end of the form</li></ol>
<b>Expected result</b>	New event is successfully created. Events/event:id/manage page is shown.

### 3.6.2 Create event without logo

<b>Test ID</b>	AT14
<b>Test description</b>	Create event without logo
<b>User Story ID</b>	US 7.2, US 8.4
<b>Requirement ID</b>	FR 9.3, FR 9.5
<b>Precondition</b>	User is already logged in and wants to edit an event he/she already created.
<b>Test procedure</b>	<ol style="list-style-type: none"><li>1. User goes to “My events” page</li><li>2. User clicks on the “Create event”</li><li>3. User correctly fills all the fields (name, type, date, county, city) of the form except logo</li><li>4. User clicks on “Create” button at the end of the form</li></ol>

**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

<b>Test ID</b>	AT15
<b>Test description</b>	Create event without filling all the required form fields
<b>User Story ID</b>	US 7.2, US 8.4
<b>Requirement ID</b>	FR 9.3, FR 9.5
<b>Precondition</b>	User is already logged in and wants to edit an event he/she already created.
<b>Test procedure</b>	<ol style="list-style-type: none"><li>1. User goes to “My events” page</li><li>2. User clicks on the “Create event”</li><li>3. User doesn’t fill all the required fields of the form (name, type, date, county, city)</li><li>4. User clicks on “Create” button at the end of the form</li></ol>
<b>Expected result</b>	“Create” button is unavailable. An event can’t be created.

### 3.6.4 Edit event

<b>Test ID</b>	AT16
<b>Test description</b>	Edit an existing event
<b>User Story ID</b>	US 7.2, US 8.4
<b>Requirement ID</b>	FR 9.3, FR 9.5
<b>Precondition</b>	User is already logged in and wants to edit an event he/she already created.
<b>Test procedure</b>	<ol style="list-style-type: none"><li>1. User clicks on “My events” tab .</li><li>2. User clicks on the “Organized” tab in my-events page</li><li>3. User clicks on the “Edit event” button of the event he/she wants to edit</li><li>4. User correctly fills the fields of the form which he wants to edit</li></ol>



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

### 3.6.5 Delete event

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

### 3.6.6 Edit event’s route

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

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

### 3.6.7 Managing registration of an event

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

<b>Test ID</b>	AT20
<b>Test description</b>	Start the tracking of an event
<b>User Story ID</b>	US 7.7, US 8.4
<b>Requirement ID</b>	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.

<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User goes on “My events” page</li> <li>2. User selects organized events tab</li> <li>3. User clicks on “Stop event” button on the event that wants to stop</li> </ol>
<b>Expected result</b>	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

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

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

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

### 3.8.2 Withdraw enrollment

<b>Test ID</b>	AT25
<b>Test description</b>	Withdraw enrollment in an event
<b>User Story ID</b>	US 8.3, US 8.6, US 8.7
<b>Requirement ID</b>	FR 10.3, FR 10.10
<b>Precondition</b>	User is already logged in and he/she has already enrolled in the event.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User goes in “My events” page</li> <li>2. User selects “Enrolled” tab</li> <li>3. User chooses the event in which he/she wants to withdraw enrollment</li> <li>4. User clicks on “withdraw enrollment” button and confirm</li> </ol>
<b>Expected result</b>	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

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

<b>Precondition</b>	User is already logged in and he/she has previously participate in an event.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User goes in “My events” page</li> <li>2. User selects Enrolled tab</li> <li>3. User selects the event that is passed</li> </ol>
<b>Expected result</b>	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

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

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

**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

<b>Test ID</b>	AT29
<b>Test description</b>	Delete user account
<b>User Story ID</b>	US 11.2
<b>Requirement ID</b>	FR 13.3
<b>Precondition</b>	User is already logged in.
<b>Test procedure</b>	<ol style="list-style-type: none"><li>1. User opens website</li><li>2. User clicks on the “My Profile” button in the navbar</li><li>3. User clicks on the “delete Account” button</li></ol>
<b>Expected result</b>	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

<b>Test ID</b>	AT30
<b>Test description</b>	Delete user account while still being the event organiser
<b>User Story ID</b>	US 11.2
<b>Requirement ID</b>	FR 13.3
<b>Precondition</b>	User is already logged in.
<b>Test procedure</b>	<ol style="list-style-type: none"><li>1. User opens website</li><li>2. User clicks on the “My Profile” button in the navbar</li><li>3. User clicks on the “delete Account” button</li></ol>
<b>Expected result</b>	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

<b>Test ID</b>	AT31
<b>Test description</b>	Contact the system support
<b>User Story ID</b>	US4.2
<b>Requirement ID</b>	FR5.2
<b>Precondition</b>	-
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User opens website</li> <li>2. User clicks on the “Contacts” button in the navbar</li> <li>3. User fills the form with the info and a message he/she wants to send to the system support</li> <li>4. Users clicks on the “Send message” button.</li> </ol>
<b>Expected result</b>	<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

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



	4. User clicks “Sign in” button.
<b>Expected result</b>	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

<b>Test ID</b>	AT33
<b>Test description</b>	Log into the mobile app with wrong credentials (email or password)
<b>User Story ID</b>	US 5.1
<b>Requirement ID</b>	FR 6.1, FR 6.4
<b>Precondition</b>	User is already registered and has already installed Ridertrack mobile app in his/her smartphone.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User opens Ridertrack app in his/her smartphone .</li> <li>2. User clicks on “Login” button in the screen appeared.</li> <li>3. User incorrectly (inserting wrong email and password) fills the login form .</li> <li>4. User clicks “Sign in” button.</li> </ol>
<b>Expected result</b>	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

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

**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

<b>Test ID</b>	AT35
<b>Test description</b>	Launch the tracking on the mobile app
<b>User Story ID</b> <b>Requirement ID</b>	US 9.1 FR 11
<b>Precondition</b>	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.
<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User clicks on the event, in which he/she wants to be tracked.</li> </ol>
<b>Expected result</b>	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

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

	<ol style="list-style-type: none"> <li>3. User clicks on the button corresponding to the chosen timeout.</li> <li>4. User clicks on “Set timeout” button.</li> </ol>
<b>Expected result</b>	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

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

### 3.17 System administrator

#### 3.17.1 Users overview for system administrator

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

### 3.17.2 Modification of user

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

### 3.17.3 Events overview for system administrator

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

### 3.17.4 Modification of event

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

<b>Test procedure</b>	<ol style="list-style-type: none"> <li>1. User clicks on the “Manage events” link in the navbar.</li> <li>2. User clicks on the row of the event to modify. A popup is shown with a form with data already filled in.</li> <li>3. User changes the data.</li> <li>4. User clicks on “Save” button.</li> </ol>
<b>Expected result</b>	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>
<b>FR1</b>	<b>Registration</b>	AT1, AT2, AT3, AT4
<b>FR2</b>	<b>Spectators events view</b>	AT9
<b>FR4</b>	<b>Spectators progress tracking</b>	AT11, AT12
<b>FR5*</b>	<b>Contact the system Support</b>	AT31
<b>FR6</b>	<b>Login</b>	AT5, AT6, AT7, AT8, AT32, AT33, AT34, AT37
<b>FR7</b>	<b>Users events view</b>	AT9
<b>FR9</b>	<b>Event management</b>	AT13, AT14, AT15, AT16, AT17, AT18, AT19, AT20, AT21, AT22
<b>FR10</b>	<b>Event enrollment</b>	AT10, AT23, AT24, AT25, AT26, AT27
<b>FR11</b>	<b>Users progress tracking</b>	AT11, AT12, AT34, AT35, AT36
<b>FR12</b>	<b>Interaction</b>	AT28
<b>FR13</b>	<b>Personal data management</b>	AT29, AT30
<b>FR14</b>	<b>Sysadmin login</b>	AT5, AT6, AT8

<b>FR15</b>	<b><i>Sysadmin events management</i></b>	AT40, AT41
<b>FR16</b>	<b><i>Sysadmin users management</i></b>	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.