



# RIDERTRACK

*Design Description Document*

# Index

<b>Index</b>	<b>2</b>
<b>Revision History</b>	<b>3</b>
<b>1. Introduction</b>	<b>4</b>
1.1 Purpose of the document	4
1.2 Document Organization	4
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	6
<b>2. Background and objectives</b>	<b>7</b>
2.1 Overview	7
2.2 High level description of functionality	8
<b>3. Architecture and design</b>	<b>8</b>
3.1 High level system architecture	8
3.1.2 Deployment Diagram	8
3.2.2 Component diagram	9
3.2 Cloud environment	10
3.2.1 Containers	10
3.2.2 Continuous integration and delivery	11
3.3 Web server	11
3.3.1 Data serialization	11
3.3.2 High level API schema description	11
3.4 Database	11
3.4.1 High level data models	11
3.5 Data collector	12
3.6 Web application	12
3.7 Mobile app	13
3.8 External services	13
3.9 Mapping	13
<b>4. Graphical user interfaces</b>	<b>14</b>
4.1. Web Application	14
4.1.1. Guests' views	14

4.1.1. Users' Views	17
4.2 Mobile Application for Event Participants	18

## Revision History

Date	Version	Description
27/10/2017	0.1	Initial draft
10/11/2017	1.0	First version

# 1. Introduction

## 1.1 Purpose of the document

This document gives an overview about design, technologies, architecture decisions that have been made and their implementation in the application.

## 1.2 Document Organization

The “Design description” document is essentially structured in six parts:

In this **first section**, it is provide an introduction with purpose and organization of the document.

The **second section** provides an high level overview of the system with its functionalities.

In the **third section**, it is presented an overall description of architectural components and the design choices taken .

Finally in the **fourth section**, it is provided GUI of the system for both the web application and the mobile one.

## 1.3 Intended Audience

The target audience of this document includes:

**Users:** this category holds three different kinds of users, united by the common potential interest in the project to be and distinguished by habits, interests and, especially, by the way “**Rider Track**” would improve their life experience. Below, the distinction:

↳ *Athletes:* runners, bikers, riders, marathon or trail runners, whoever loves and practises competitive sports, which include an itinerary to follow.

2. *Spectators and supporters:* whoever is interested in following a competition.

↳ *Event organizers:* whoever wants to organize and share an event, supplying the most user-friendly and performing experience to their athletes and spectators.

- **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.

## 1.4 Definition and Acronyms

### 1.4.1 Definitions

Definicija	Definicija
<b>REST API</b>	API that adheres to the REST architectural constraints
<b>API endpoint</b>	It is an URL of the API such as: /api/auth/login.
<b>Event Participant</b>	A person who has enrolled and participates in an event
<b>Event Organizer</b>	A person who has created and organizes an event

### 1.4.2 Acronyms and abbreviations

Skraćenica	Definicija
<b>GUI</b>	Graphical user interface
<b>API</b>	Application Programming Interface
<b>GPS</b>	Global Positioning System
<b>JSON</b>	JavaScript Object Notation
<b>FER</b>	Fakultet elektrotehnike i računarstva
<b>POLIMI</b>	Politecnico di Milano
<b>CRUD</b>	Create, read, update, delete
<b>DB</b>	Database
<b>URI</b>	Uniform resource identifier
<b>HTTP</b>	Hypertext transfer protocol
<b>HTML</b>	Hypertext markup language

## 1.5 References

*"Project Vision and Plan"* , Alessandro Caprarelli, Marzia Degiorgi, Mariano Etchart, Giulia Leonardi, Josip Mališa, Ante Brescic, Ivan Kvesić 2017.

*"Requirements Definition"* , Alessandro Caprarelli, Marzia Degiorgi, Mariano Etchart, Giulia Leonardi, Josip Mališa, Ante Brescic, Ivan Kvesić 2017.

SCORE project detail: <http://score-contest.org/2018/projects/ridetrack.php>

## 2. Background and objectives

### 2.1 Overview

The aim of ~~the~~ ~~system~~ ~~is~~ ~~to~~ ~~provide~~ ~~an~~ ~~user-friendly~~ ~~management~~ ~~service~~ ~~for~~ ~~outdoor~~ ~~sport~~ ~~events~~ ~~which~~ ~~will~~ ~~integrate~~ ~~tracking~~ ~~data~~ ~~from~~ ~~multiple~~ ~~sources~~, ~~those~~ ~~existing~~ ~~and~~ ~~future~~ ~~ones~~ ~~that~~ ~~haven't~~ ~~been~~ ~~introduced~~ ~~yet~~.

In particular, as it was explained in the *"Project Plan and Vision"* and *"Requirements Definition"* documents, it will be provided both a responsive web application and a native mobile one. The web application is addressed to all the actors of the system, instead the mobile app is addressed only to participants in order to collect tracking data during an event.

### 2.2 High level description of functionality

According to the deep analysis of requirements made in the *"Requirements Definition"* document, the main functionalities offered by the ~~the~~ ~~system~~ ~~will~~ ~~be~~:

- A user-friendly management service for sport events, that allows event organizers to create and manage the tracking of an event without the obligation to spend an enormous amount of money to provide tracking devices to all participants.

- A flexible service that allows participants to use their own devices, integrating multiple possibilities of input tracking sources.

- An extensible service that provide a dedicated native mobile application for participants, in order to provide further tracking data and the possibilities to receive inciting messages only by authorized spectators.

- A service that allow participants to share private data only with some spectators, such as speed, heartbeat, messages...

- A service that allow spectators to follow participants during an event, and also to access private information of a certain participant if authorized (through an authorization code that the participant shares with some spectators).

- A service that allow Event spectators to track the main data of past events through a dashboard.

- A service that allow participants to manage enrollment to events.



## 3. Architecture and design

### 3.1 High level system architecture

In this section, we focused on the physical components of the independent system, following the overview division. Each physical component and the technologies associated to it, will be described in the following sections.

#### 3.1.2 Deployment Diagram

The system has 5 main components:

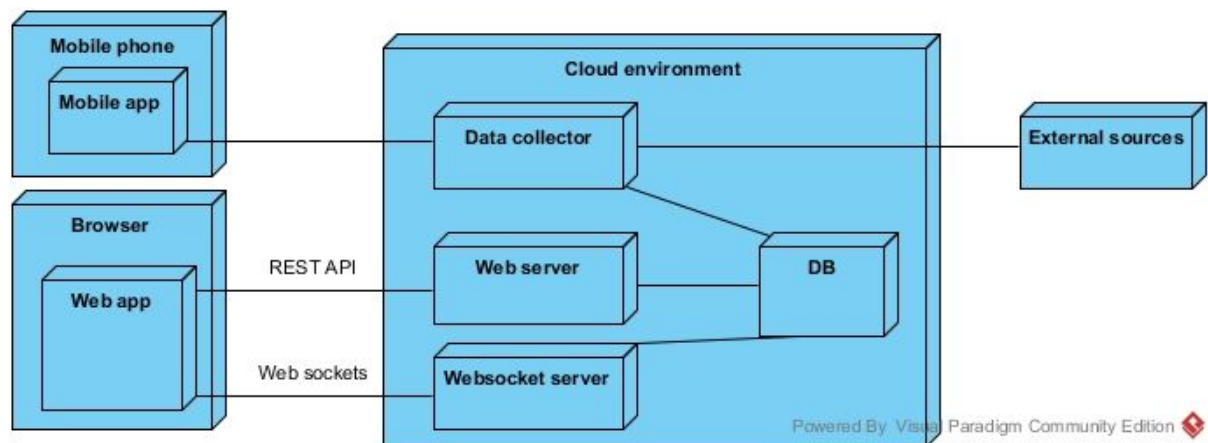
**Web application:** It enables guests, participants and event organizers to interact with the system.

**Mobile application:** It sends real time data to the server and receives messages from guests.

**Web server:** It offers a REST API with support also for websockets.

**Database:** Enables information to be created, retrieved, updated and deleted in documents (MongoDB)

**Data collector unit:** It actively fetches and passively receives from external sources.



<B/ xZ Úz Zujntk ZI { ž BM>M .

#### 3.2.2 Component diagram

The following components diagrams shows some of the most important pieces of the system and their interactions.

Web app components:

Public pages: they are components of the Angular app that are accessible by all the users of the system. They show only general information about the system such as description of the services and contact details.

Registration/login: these components of the Angular app provide forms to register and login into the system. They are visible only to unauthenticated users, called guest. They send request to the REST API and according to the responses they change the views.

Events page: this is one of the most important components of the Angular app. It has to show functionality according the role of the user for each event.

Event page: this component is as important or more as the previous one because it handles all the operations on an event. It shows different functionalities according to the role of the user of such event that is using the app.

Administrator page: this components is shown only to a special user of the application, the system administrator. It gives the possibility to change records of users and events.

Node.js application:

Static file manager: this component of the Node.js app serves the static files such as HTML, js and CSS needed for the Angular app.

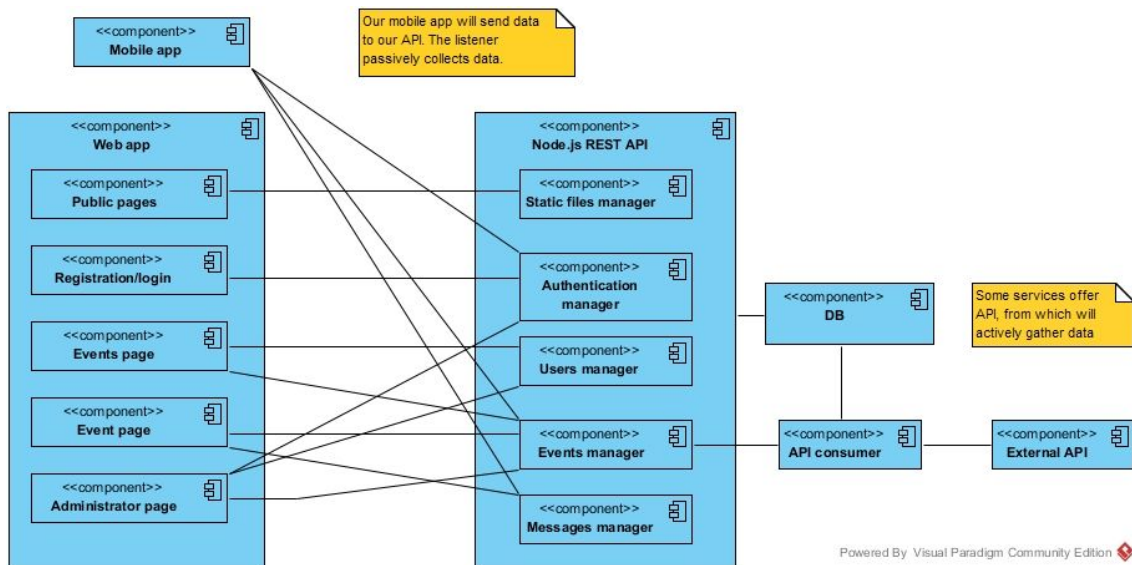
Authentication manager: this component handles the authentication and authorization in the backend. It creates and verifies jwt token and restrict the access to some API endpoints.

Users manager: this component manages the CRUD operations on users in the database.

Events manager: this component manages the CRUD operations on events in the database. It will be largely used by the web application to get data of an event, especially during the event for real time progress. It will also be largely used by the mobile application to send data about the location of the participant.

Messages manager: this component manages message exchanges between spectators and participants.

API consumer: this component will fetch data from external APIs, populating the database with information about location of participants that use supported devices and applications.



*<del>[xZ Ü@nk unl ZI {ž bV]xNR*

## 3.2 Cloud environment

The complete backend of the system will reside on [Microsoft Azure](#) public cloud service.

### 3.2.1 Containers

All the components running in the cloud will be isolated inside containers. [Docker](#) will be used as a containerization technologies.

Each component will have its own Docker file that specifies the build instructions.

All the components will then be orchestrated using a Docker Compose file that specifies the connections among them and runs the complete system.

### 3.2.2 Continuous integration and delivery

Deployments will be managed by [Jenkins](#), an automation tool that offers continuous integration and facilitates technical aspects of continuous delivery.

Changes to the codebase are pushed to Github's repo. Jenkins is attached to service hook that triggers events every time a change is pushed to GitHub.

## 3.3 Web server

The web server is the core component of the system. Most of the users interactions are with the server.

It will be implemented using [Express.js](#) framework on top of a [Node.js](#) web server.

It will also support websockets using the [Socket.io](#) library that integrates very well into an Express.js application.

### 3.3.1 Data serialization

All the communication among the components of the system are done using JSON, a lightweight data-interchange format.

It is easy both for humans and to machines to read, to parse and to generate it.

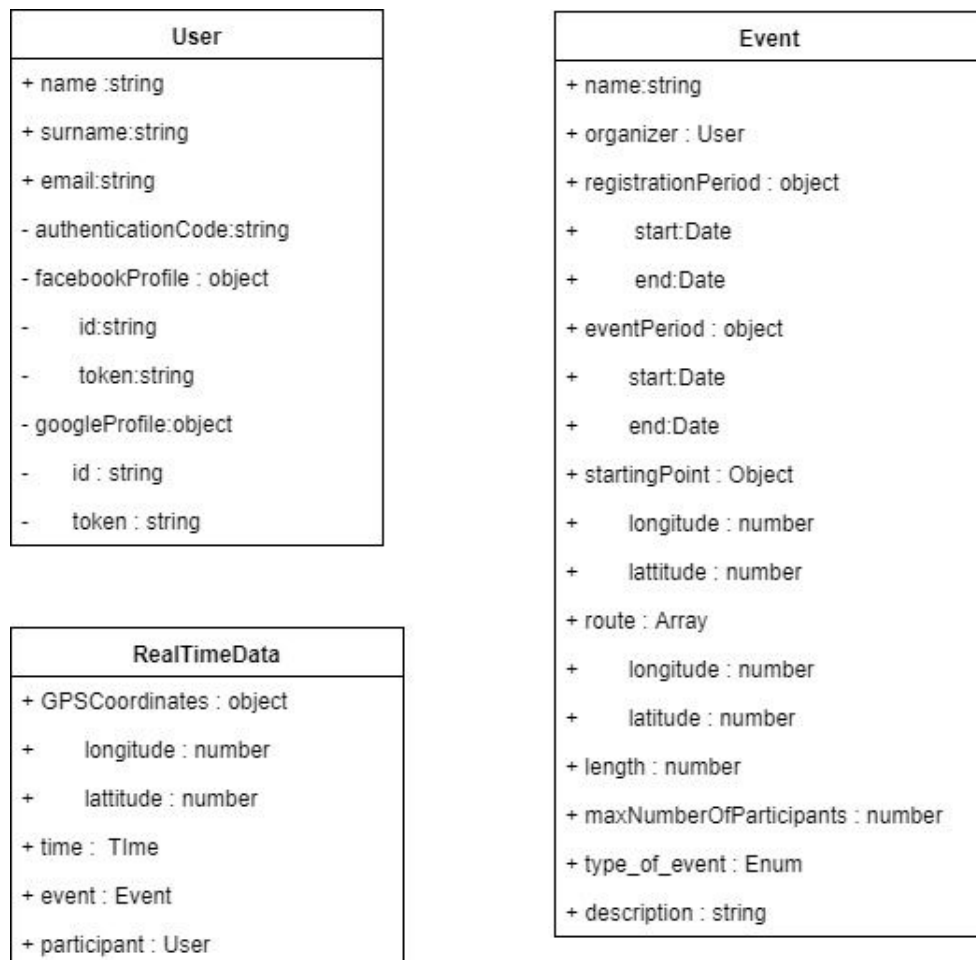
### 3.3.2 High level API schema description

The webserver will offer a REST API that both the web app and the mobile app will use in order to get and modify data.

It will also handle the authentication and the authorization of users.

## 3.4 Database

### 3.4.1 High level data models



<M/ xZ Ūz MINUMZ 2 nXZj ž BMXVK

As shown in the database model diagram the high level data structure of each model item must include the specified fields.

### 3.5 Data collector

The data collector unit will be one of the most important of the complete system. It will have two main functions:

- it will passively listen for data from the mobile application, practically being a REST API. It will be incorporated into the Node.js app.

- it will actively fetch data from external APIs.

### 3.6 Web application

The web application is one of the most important component of the system since it will handle most of the interactions with the user.

It will be done using Angular.js 4, a framework to make reactive web application.

The application will be a SPA and it will adapt views such as events list, event details and others to the role of the user whom is using the application, using component of Angular such as guards and services.

The connection with the backend is done through REST API calls and websockets stable connection.

### 3.7 Mobile app

The aim of the mobile application for participants only is to provide real time data to the server and to offer an unidirectional communication system between spectators and participants. In fact participants can authorize group of spectators to access private data through a code and to send messages to them during the race.

It will be implemented only for Android operating system. Besides it must be optimized and it must not drain the battery of the smartphones.

### 3.8 External services

The system will use **external API** such as fitness apps (runtastic, nike+, google fit...) and tracking services ([mapprogress.com](http://mapprogress.com), [trackleaders.com](http://trackleaders.com), [findmespot.com](http://findmespot.com)) to gather tracking data from multiple sources.

### 3.9 Mapping

The following table is used for the mapping between the user stories and the architectural components. It describes how the functionalities are mapped into the different components.

The first column is a list of user stories from the requirements document and the second one is the component responsible to accomplish them.

User story ID	Component
US9, US10	Mobile app
US1, US2, US3, US4, US5, US6, US7, US8, US9, US10, US11, US12, US13, US14	Web app
US1, US2, US3, US4, US5, US6, US7, US8, US9, US10, US11, US12, US13, US14	Web server
US1, US2, US3, US5, US6, US7, US8, US9, US11, US12, US13, US14	Database

## 4. Graphical user interfaces

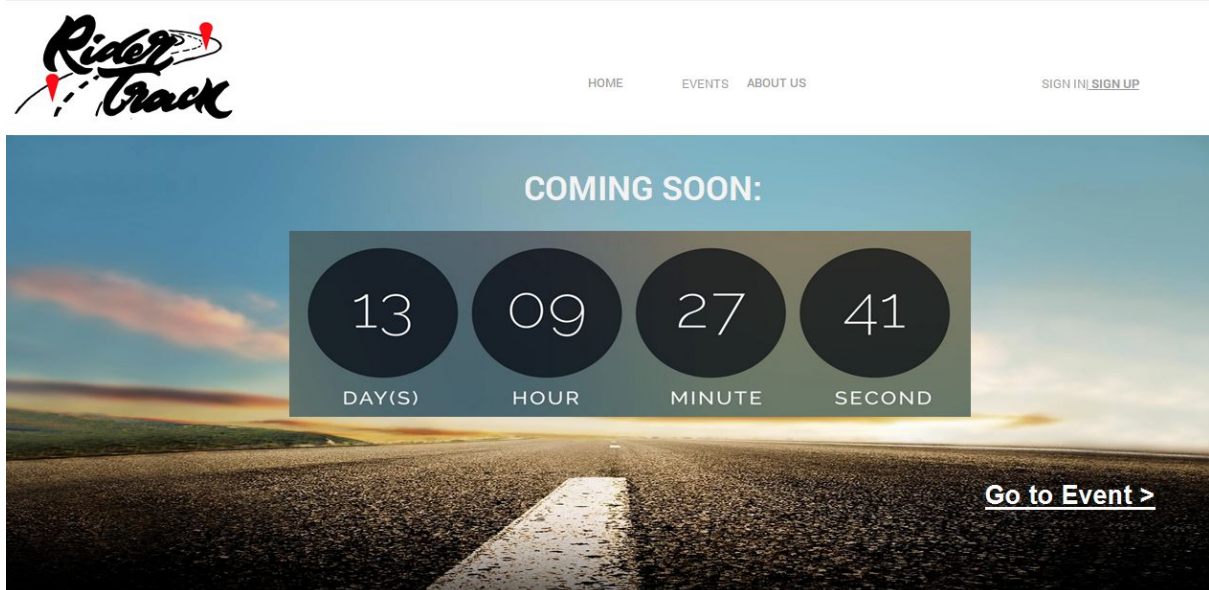
The following mockups show a brief overview of the systems interface. The Web graphical interface is meant for all actors and the Mobile GUI just for participants.

### 4.1. Web Application

#### 4.1.1. Guests' views

This is the public section of the responsive website that allows to see:

- The list of events, and the tracking of a specific event.
- The homepage
- The About Us / Contact Us page
- The sign-in/sign-up forms



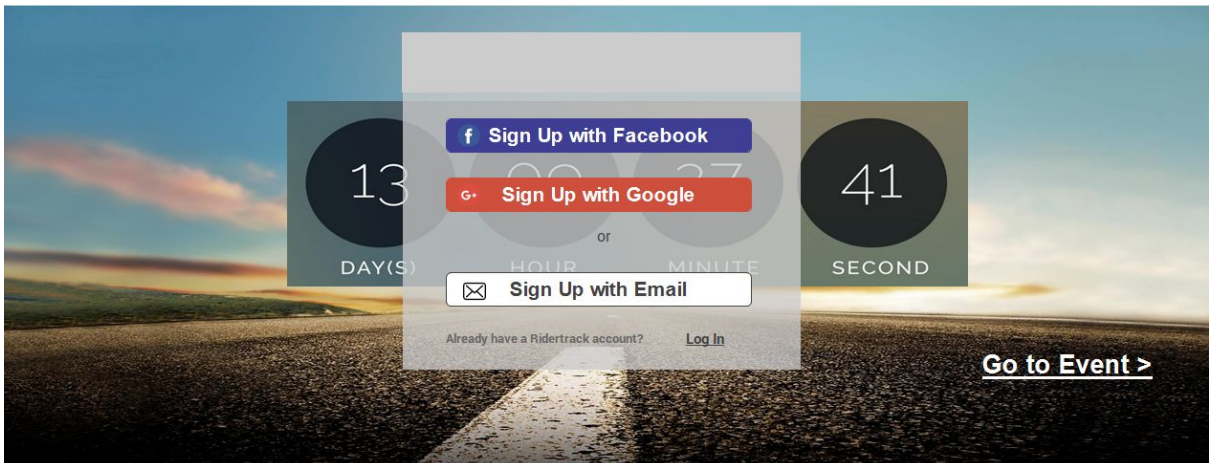
### Why Ridertrack

*Ridertrack is the first platform allowing to track your favorite athlete or the entire event in the most simple way. (...)*

- Easy Mobile app for Participants [Get Now >](#)
- Easy Web app for Event Organizers, Participants and Spectators

2 nM | uU@ | UjW) nk Z' <MZ'

It is possible sign up or sign in choosing the desired methodology: email, Google or Facebook.

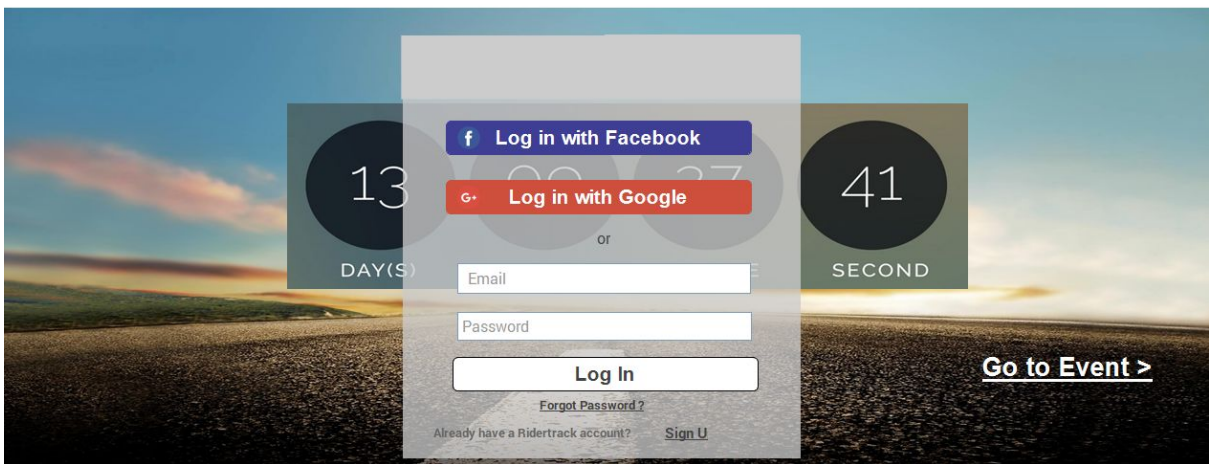


## Why Ridertrack Best

Ridertrack is the first platform allowing to track your favorite athlete or the entire event in the most simple way. (...)

- Easy Mobile app for Participants [Get Now >](#)
- Easy Web app for Event Organizers, Participants and Spectators

2 nM | uÜ©?Z by xMbl " nk ' .



## Why Ridertrack Best

Ridertrack is the first platform allowing to track your favorite athlete or the entire event in the most simple way. (...)

- Easy Mobile app for Participants [Get Now >](#)
- Easy Web app for Event Organizers, Participants and Spectators

2 nM | uÜ©In` b " nk ' .



The public website gives the possibility to follow the tracking of a participant during an event. If the spectator has an authorization code, they may access information about speed and heart rate, additionally can send messages to that participant during the race.

**RiderTrack**

HOME [EVENTS](#) ABOUT US SIGN IN | SIGN UP

← **"Manhattan Marathon 2017"** **Category: Race**  
**Status: On Going**

JOHN SMITH

**Name:** JOHN SMITH  
**Number:** 13  
**Access Private information:**  ▶

**SPEED** **ACTUAL RANKING**

**DISTANCE** **INBOX**

✉ ▶

2 nV / u Y © A W W b ` n M U N K b u M { X | x b ` M " f Z I {

#### 4.1.1. Users' Views

The user can:

- Access the list of events and enroll in an event
- See his/her upcoming events in which he/she has enrolled
- Create an event
- Manage created events
- Manage private information



HOME

EVENTS ABOUT US

[CREATE AN EVENT](#) | [ACTIVE EVENTS](#)

## Create an Event

Create

Name of the Event

Choose Category

Select Registration Period

Add a Description

Date of the Event:

September 2017

MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Select Route/s:



Select Tracking Device

Available:

RFID

GPS Tracker

Devices

Upload Image

2nM / uP@ xZMbl n MI " fZl {

## 4.2 Mobile Application for Event Participants

The event participant can:

Login the application with the same credential of the web app

Listen to messages from authorized spectators during an event

Share data or results on social media, such as Facebook

