



# Minutes of Meeting

<b>Project</b>	<b>RiderTrack</b>	<b>Date</b>	30 <sup>th</sup> Nov2017
		<b>Start-end time</b>	19:00-20:00
<b>Responsible</b>	Alessandro Caprarelli	<b>Location/type</b>	Video Conference Milan – Zagreb

Attended by	Location	Remarks
Alessandro Caprarelli	Milano	Remote
Josip Mališa	Zagreb	Remote
Ivan Kvesic	Zagreb	Remote
Mariano Etchart	Milano	Remote

## 1. Next steps for beta prototype

For the beta prototype we need to handle routes and location of participants.

### CONCLUSION:

Data models to create:

Route	Position
eventId: String [ [lat, long] ]	userId: String eventId: String timestamp: Data trackingSource: String [lat, long]
Index: eventId	Index: (userId, eventId, timestamp)

Routes to create:

GET, POST, PUT, DELETE /api/events/:eventId/route	To change the route. Only the organizer can change the route.
POST /api/events/:eventId/tracking/start	To start the competition and start collecting data. Set the status to ongoing.
POST /api/events/:eventId/tracking/stop	To finish the competition and stop collecting data. Set the status to finished.

GET /api/events/:eventId/participants/positions	Get the most updated position of the participants of that event. It processes the data and returns an ordered list of the participants position. [ position ]
POST /api/events/:eventId/participants/positions	The mobile app will send data to this endpoint. It receives a location object.

Event statuses: 'notStarted', 'onGoing', 'finished'. To add in the event model.

Add to event model the "allowedTrackingSources" field. It is an enum and now accepts only "ridertrack-mobile"

**ACTION:** Have a look at swagger.io for API documentation. Josip will take care of it.

**ACTION:** Ivan will take care of route handling.

**ACTION:** Josip, Mariano will take care of position handling (including ranking).

**ACTION:** Alessandro will take care of the event status and tracking management in general.