

Open Data and Green IT CodeCamp - Spring 2015

Sport Finder

Mansoureh Roustia 0444957

Ibrahim Olanigan 0444821

Veronica Morales 0404557

Manuel Delgado 0385595

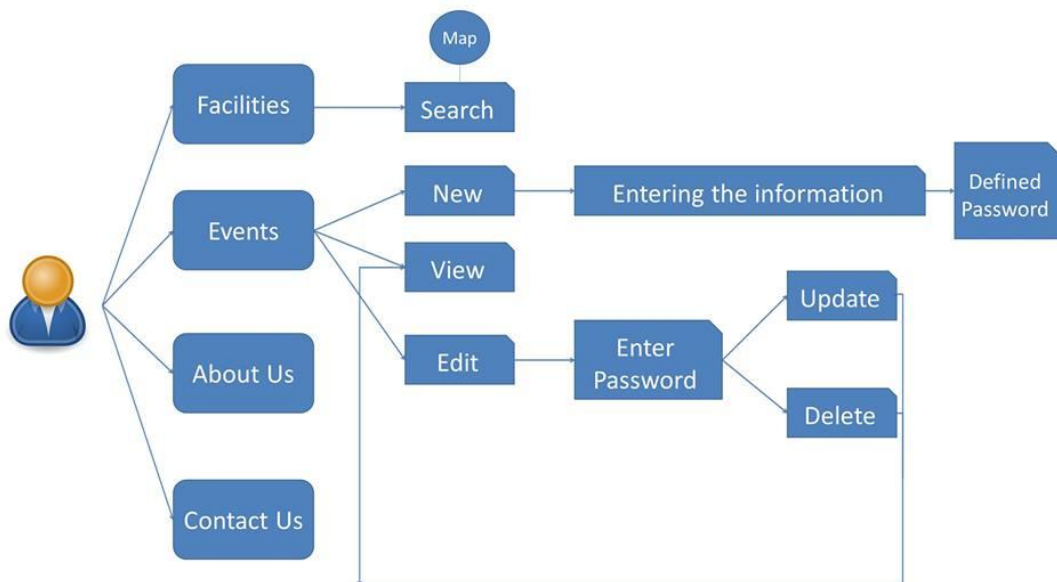


Introduction: What is Sport Finder

Searching for a place where to practice your favorite sport is now easy, fun and possible for everybody. Sport Finder is a web application mobile friendly where users can search for all the sport facilities in the city of Lappeenranta and locate them in a map. Sport Finder uses Open Data provided by the Finnish Government.

With Sport Finder it's possible to find all the sport events that will take place in the area of Lappeenranta. Anyone has the right to create an event that will take place in one of the public places that Lappeenranta offers for the citizens. The main purpose of this application is to make people more interested of sports, inform about all the available possibilities to practice sport in the city and to encourage people to create events and gather with friends and unknown people to practice sport in group.

Once you have created your event, you can access it with your password, edit or delete it. All the possibilities the user can find in Sport Finder are summarized in the following picture.



Picture 1: Utilities of the application



Motivations

Sport Finder is a web application mobile friendly developed during the Open Data and Green IT CodeCamp in Lappeenranta University of Technology.

The main reason why group 1 wanted to develop this project is because we wanted to take Open Data to the next level: to lead the data to the users as information. We are all interested on different sports and we have had difficulties finding a place in the city where to practice them.

The purpose of Sport Finder is to make for the users easy to find all the facilities the city offers to practice sport. Giving the information to the customer in an easy way as an application makes him or her be more interested on exploiting the sport areas and consequently practice more sport. The users can see the facilities listed and located in the map to have more possibilities to end going to practice sport because we eliminate the excuse of not knowing how to arrive to the place.

Another feature Sport Finder offers is the possibility to create events in any of the Sport facilities in the city area. Users can create an event that will be shown in the web as public for letting people join if they want. This feature has been added to the application thinking mainly of those people that would like to practice a team sport and they don't have the enough participants, for people that feel lonely or for people that just want to have a nice time and meet other people. With this functionality we want to solve some social problems and make people interact with each other making some nice activity that they will enjoy.

Goals

The goal of the application consist of encourage people to play sports, improve the health of the citizens, ease the practice of team sports and connect people with similar sport interest.

We set that our main target was to place the sport facilities in the map and the management of events and we reached it during the CodeCamp. In addition, we decided that the website must be responsive for possible ports.

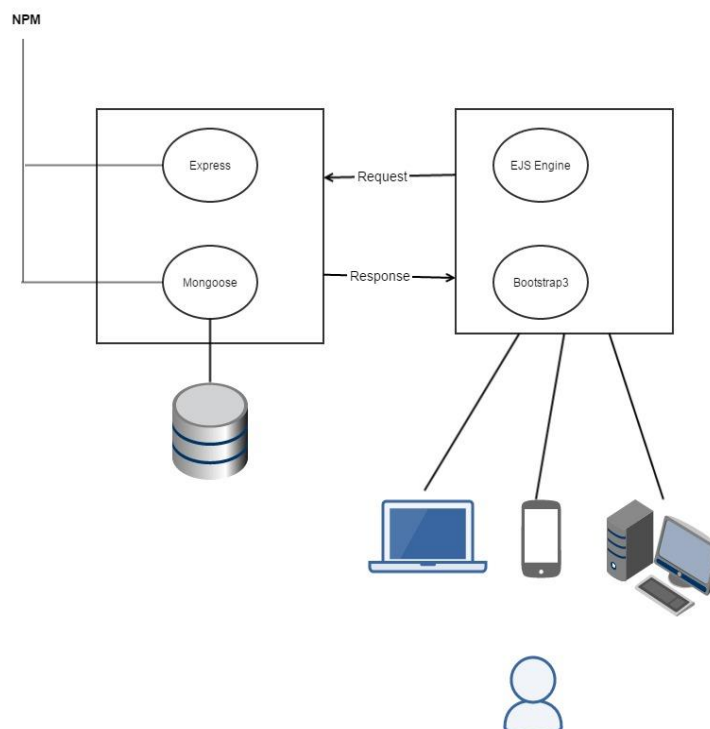
Architecture of Sport Finder

The project has been developed in the coding platform cloud9 for all the members of the team to have access to the code at the same time without problem of conflict with the version. The database we have been working with is MongoDB, concretly with Mongoose since we are working with Node.js and with web framework for Node.js: Express. Inside out DB we have two collections: Events and Facilities.

- The Events collection is where all the events created by users are stored.
- The Facilities collection stores all the sport facilities that the city of Lappeenranta offers. This collection has been imported from the csv file with the facilities that can be found in the web page Lipas (<http://liikuntapaikat.fi/lipas>).

For the client side we have used Embedded JavaScript (EJS) as our view engine. We made use of Bootstrap3 to make our application mobile friendly with an intuitive user interface and nice views for the customer.

The general architecture is defined in the following picture.





Picture 2: General architecture

Back-End Design

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Using Node.js we could create a webserver at the same time we developed our application running on it.

Express.js is a web framework for Node.js with which we acquired simplicity and a cleaner code easier to read. With it we could manage the routes and organize the different views.

Using Mongoose, we were able to define a schema of facilities and another schema of events. This is an example of our use of mongoose:

```
var eventSchema = new Schema({
  'ID':Number,
  'Title':String,
  'Description':String,
  'Date_':Date,
  'Place':String,
  'Address':String,
  'Access':Boolean,
  'MaxGuest':Number,
  'Password':String,
  'Created_at': Date
});
var Event = mongoose.model('Event', eventSchema, 'Events');
```

Front-End Design

The views of the application were rendered using EJS engine and styled using Twitter Bootstrap.

The next features were implemented in this version of the application:

- Find sport facilities by type using REST.
- Locate facilities in the map using the Google API.
- Treatment of events (create, modify and delete).

The simple interface facilitates the usage showing the facilities in the map after selecting one type on the left sidebar menu. Regarding the events, all the events are

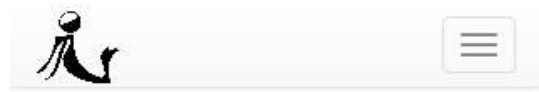
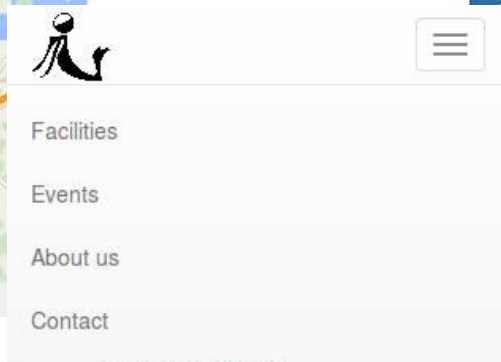
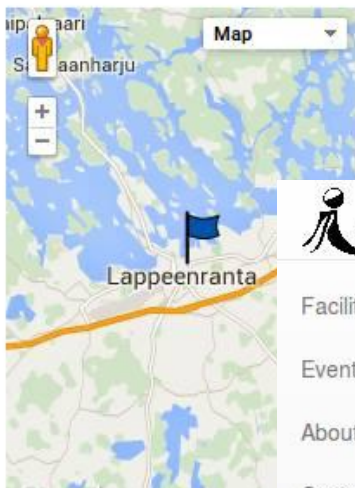


listed in the same page and they include a button for simple editing after providing the password.



Introduce the facility you are looking for:

Bowling alley



Create your event right now!

Golf with friends

We will play golf for 2 hours, bring your own material!

Edit



your own material!

Edit



Results

Finally, we have reached the main goal of the application. This app can help the citizens for finding sport facilities and creating new events for group sports. The issue we confronted in the presentation day took couple of hours of work and the problem leads to unsuccessful presentation, but we completely fixed the problem.



We agree that we have learned new methodologies and programming language participating in this code camp, even the issue before mentioned proved us that we should be more prepared before beginning final presentation.

At last, using new environment (C9.io), frameworks (Express, Bootstrap), non-relational databases (Mongo DB with Mongoose) showed us a different programming approach.

API

The API provided by the application is explained as follows.

/facility/type

GET /facility/type

Facility

Summary

Types of sporting facilities in Lappeenranta

Description

The Facility Type. endpoint returns all the types of facilities available in Lappeenranta.

Responses

Code	Description	Schema
200	An array of facility types in Lappeenranta.	<pre>▼ [▼ Type { type: ▶ string }]</pre>

Try this operation

REQUEST: <http://sports-olanigan.c9.io/facility/type/>

```
Sample response
["Unsupervised beach",
 "Ski jumping hill for training"]
```

/facility/type/facilitytype



GET /facility/type/facilitytype

Facility

Summary

Information about the specific facility type in Lappeenranta.

Description

This endpoint returns all the information about location for the specified sport facility in Lappeenranta. The response includes the name of the facility, the type of the facility and it's XY co-ordinates (based on the World Geodetic System, WGS84)

Responses

Code	Description	Schema
200	An array of the locations of the specified facility type.	<pre>[Facility { name: string type: string xwgs84: string ywgs84: string }]</pre>

Try this operation

SAMPLE REQUEST: <http://sports-olanigan.c9.io/facility/type/Athletics%20field>

```
Response for Athletics field
[ { Name: 'Joutsenon urheilukentta',
  Type: 'Athletics field',
  XWGS84: 28.5057,
  YWGS84: 61.1126 },
  { Name: 'Kimpisen yleisurheilustadion',
  Type: 'Athletics field',
  XWGS84: 28.1985,
  YWGS84: 61.0628 } ]
```


/event/all

GET /event/all

Event

Summary

Sporting events in Lappeenranta

Description

The Events endpoint returns all the sporting events activities in Lappeenranta which have been stored with the application.

Responses

Code	Description	Schema
200	An array of sporting events in Lappeenranta.	<pre>[Event { access: boolean address: string date_: string description: string maxguest: integer title: string }]</pre>

Try this operation

REQUEST: <http://sports-olanigan.c9.io/event/all>

```
[ { "Title": "Golf with friends",
  "Description": "We will play golf for 2 hours, bring your own material!",
  "Date_": "2015-03-21T00:00:00.000Z",
  "Place": "Golf training area",
  "Address": "Golf Field",
  "Access": true } ]
```

Group Experience

We have learnt to work with new technologies like NodeJS and MongoDB, and using peer-programming for collaborative development.

We have learnt how to connect MongoDB with the logic and use the Open Data to capture the data that we need to create an application.



Using the Google API and Bootstrap framework for front-end design, the team has developed a simple application where the users can improve their experience when arranging a sport event.

Future improvements

As future improvements that we did not accomplish during the week:

- Connect our application to Facebook and Twitter in order to ease the process to the users and trying to avoid fakes.
- Send an email to the user with the information of the event and a link to confirm the creation of it.
- Set and control the maximum amount of assistant to each event.
- Place the events in the maps when clicking a facility.
- Port to Android using PhoneGap.

Conclusion

Open Data give users a world of possibilities to develop new applications to make our life easier. Sport Finder will improve your sport experience solving the problems to find a place or activity to practice any exercise you want.

Besides improving people's quality life, while developing this application, the team members improved their coding skills using different technologies like Node.js and MongoDB. As a team we learned to share tasks, giving and getting feedback in order to upgrade our professional skills, helping and motivating each other.