T 111.4360 Design of WWW Services

Final Phase

Kuntorastit

May 18, 2014

Group 13

[redacted]

Aalto SCI

Introduction and design

Introduction

Kuntorastit is a service for orienteers. Orienteering - a very popular Finnish sport - is about running in a forest with a map and a compass, finding one's way through predetermined checkpoints marked in the map and finishing as fast as possible. An orienteer (in Finnish: *suunnistaja*) is a person who engages in this sport. Further information about the sport can be found for example from Wikipedia (http://en.wikipedia.org/wiki/Orienteering.)

Local orienteering clubs organize orienteering events (e.g. "Viikkorastit" or "Iltarastit") usually weekly in the summertime. This means that the race responsible person hides the checkpoints in a forest and marks them on a map of that area. Then he prints out a bunch of copies of that map. The location of the gathering point (usually with car parking facilities) is announced in the club's website or in a local newspaper. The local orienteers check the info and come to the event. On place they buy a copy of the map and head to the forest to look for the checkpoints.

The difficult thing is that one club organizes orienteering events in various forests within up to 100 km distance from the club's home town. It is too boring to criss-cross in the same forest each week – the point is that the terrain, the route and checkpoint locations are as unknown as possible for the racer. Therefore the location changes all the time and it's a weekly task for an orienteer to check the location of the next event, for example from different clubs' websites or local newspapers.

We want to solve the problem by providing an online WWW service which shows all the upcoming orienteering events in Finland in one clear, easy-to-use, mobile-friendly, accessible UI.

Currently the Finnish Orienteering Union SSL (Suomen Suunnistusliitto) hosts a calendar (bit.ly/1n5jEsi) of the hobby orienteering events of the country, which includes ca. half of all the events. Nevertheless, it's UI is complex, laborious, ugly and unoptimized for mobile users. It's also difficult to find from their website and even though it's the biggest database of Finnish orienteering events online, the user cannot be sure if they miss an event nearby just because it hasn't been added to the calendar.

Additionally, some local sports clubs have their own orienteering events placed on a map and/or a list on their website. See, for example Espoorastit by the clubs Espoon Akilles and Leppävaaran sisu here: http://www.espoonakilles.fi/er.php. Same problem as in SSL's calendar occurs, but in a smaller scale: users can see only the events organized by those two clubs under the same brand (Espoorastit), even though there are also other clubs' events in the same area. Of course the other clubs usually have their own websites, but the users are forced to switch back and forth different websites with different UIs to get a coherent view of the offerings relevant for them. Also the Espoorastit site is unoptimized for mobile users.

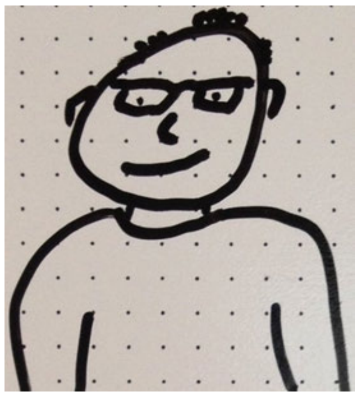
Thirdly, there is Suunnistusnet (http://suunnistus.net/), which is the only notable Finnish online community related to orienteering. They have an event calendar, but only major, usually national orienteering races are listed, not the local weekly events. What Suunnistusnet's calendar lacks is content for average hobbyists – exactly what we want to offer. Positively, Suunnistusnet's 2014 updated website is partially responsive, although one can hardly call its design mobile-friendly, let alone mobile-first.

We want to provide a service that is loved by the end-users. We hope all the orienteering clubs – including also those currently not sending their event data to SSL – start to use Kuntorastit alongside or even over their own various online calendars, or embed our content into their website. In this way we could unify and clarify all the relevant information to get the end users moving with a great outdoors sport. There is also a possibility to build the service into a specific community around the topic of orienteering which would draw advertisers, orienteers, prospective orienteers, sports clubs and the SSL more closely together.

Even though this course does not require that the students get in contact with a possible real customer, we thought that co-operation with Suomen Suunnistusliitto (SSL) would help us in this project. We contacted SSL and told them about this idea. They were eager to help us in any way they could in exchange for the code, service idea and all implementations. After this course we will give this service to them and they can improve it further if they want. We have gotten event data, information about orienteering and test users from SSL.

User group analysis

We have modelled our possible user groups by using personas. We have three personas.



Persona 1: Seppo

- Gender: Male
- Age: 55
- Education/profession: M.Sc (DI), manager in a private company
- Location: Kirkkonummi
- Means of transportation: car
- Tech skills: moderate
- Role: user

Seppo drives to orienteering events weekly after work. He likes orienteering, but he is very busy with work. He uses a desktop PC and an iPad at home and a laptop PC at work, but he finds it most reliable to schedule with a paper calendar. He likes to drive in his car.

Seppo goes to orienteering because it's good exercise, which his wife emphasizes, and he likes to move in the nature. He used to do it more seriously when he was younger. Seppo is not a member of any orienteering club.

Persona 2: Kati

- Gender: Female
- Age: 18
- Education/profession: High school student
- Location: Haaga, Helsinki
- Means of transportation: public transport
- Tech skills: high
- Role: user

Kati lives at home with her parents and a brother. She trains orienteering seriously and aims for the yearly youth world cup. She's a member of Helsingin Suunnistajat. She uses a digital calendar on her smartphone to schedule her schoolwork, cello lessons and orienteering training. She also uses her parents' desktop PC at home to do her schoolwork. She has a heart rate monitor that she uses when orienteering or other exercise and she is interested in how technology can make her other hobbies easier or more interesting. She is used to public transport but her father can also give her a ride when needed.

Persona 3: Niilo

- Gender: Male
- Age: 78
- Education/profession: Retired engineer
- Location: Matinkylä, Espoo
- Means of transportation: car
- Tech skills: low
- Role: payer, user, promoter

Niilo *loves* orienteering. He remembers the day when the sports club Espoon Akilles was founded, the same day that he turned 12. Since then he has been participating actively in the club. He lives with his wife in Matinkylä and they have a desktop PC. They are able to read the news online and use Skype to talk with their relatives, with some help of their son who lives in Porvoo.

Since his retirement Niilo has been the main responsible organizing weekly orienteering events "Espoorastit". He knows everybody around the circles but is concerned that there are too few young people enjoying the hobby. Niilo sees that in order to get more hobbyists to orienteering events, orienteering must be put online. He understands that he himself does not have the skills to do this, but is enthusiastic about listening to younger people about how, where and when they would like to do orienteering. He has heard about events on Facebook and would like to learn to use Facebook or other social medias to promote orienteering events.

Scenarios and use cases

The scenarios and respective use cases, divided by personas, are presented on table 1.

Table 1. Scenarios and Use cases with the original numbering.

#	**Scenario**	**Use case**
1	**Seppo** wants to see where and when the next orienteering event takes place nearby him.	**Seppo** can search upcoming nearby events by his address or looking from the map. Now that he knows the address he can plan a driving route and check the available parking on his navigator or Google maps.
2	**Seppo** is at work and he wants to check the location of the orienteering event that he is going today after work. He can remember the time, but he needs to find the exact place and check how long it takes to drive there from his work. He has orienteering equipment with him, so he doesn't need to visit home before the event.	**Seppo** will check the events today from the service. He will get the exact address from the event's detailed information.
3	**Kati** wants to schedule her summer training sessions and see the whole calendar for all training possibilities in greater Helsinki area.	**Kati** can search for all the events in the summer and mark them into her calendar, which she uses for planning her daily schedule.
5	**Niilo** wants to see more people come enjoy orienteering. He wants to promote the hobby and guide new people when they come to the orienteering events. He has just talked about orienteering to his son's friends at a party they held (at Porvoo at his son's place). Niilo has told the friends that there are a lot of events near Helsinki and that they should come and try it out. He still wants to share some more information about the events the friends.	**Niilo** can inform about the "iltarastit" or other upcoming event via this service. At the party, Niilo mentioned the Kuntorastit-service, but he feels that his friends might not remember to visit the site. He wants to send them a link to the service in general or he could send a link to a specific event that he knows would interest his friends.

Main features

- find nearby events
    - scenario/use case 1 and 3
    - users: all

- check details about a certain event
    - scenario/use case 2
    - users: all
- share a link to an event
    - scenario/use case 5
    - user: Niilo

Structure of the site

We aimed at and almost completely achieved constructing a Single Page App (SPA), where all the necessary functionality is packed to the landing page, and other pages play only supportive role. The originally planned structure of the site is presented in figure 1. Green arrows represent primary user paths (links) between pages. Additionally, black dashed lines divide between the site's 3 functional areas: static information (left), main functionality (center) and additional information (right). The center and left column pages are accessible from every page in the service.

On the landing page (http://final.hype2013.hype.tml.hut.fi/) the user can search for events and sees the results of their search. There is also a small About page (http://final.hype2013.hype.tml.hut.fi/tietoja.php) which contains support material that helps if the user has problems using the service. On about page we also tell briefly about the project and the project group for the sake of transparency.

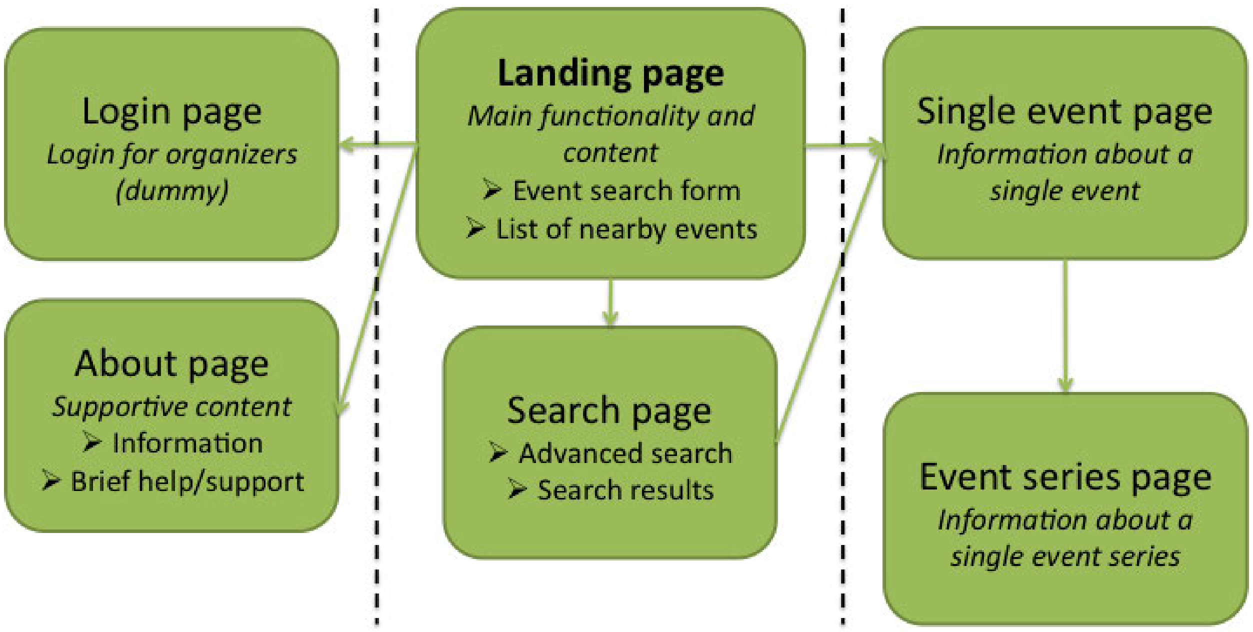


Figure 1: The structure of the site.

We had plans for the service to include a functionality for orienteering club event organizers to log in to the system and manage their event data on a separate page, but eventually had to drop this feature as we did not have enough time for implementing it. We had previously already decided that for users who are not responsible for organizing any events, it is unnecessary to have a login possibility, and therefore it was not implemented either. We decided that normal users don't want to save their favorite searches, so we decided not to provide a login for them.

Reflection

The user groups remained basically the same during the process, thanks to in-depth thinking and consideration in the beginning of the project. We were aiming more at a Twitter-like single page app, but that was out of reach due to our limited technological competence. However, the initial decisions and ideas seem to work to this day, and it seems reasonable to carry out further development.

As can be seen, the number of main features and is lesser than in our design document. We had to leave out some of them due to time issues.

Our initial decisions were good, but we didn't have enough time to implement them all. We would change this by changing this course into an intensive course - we would have more time and energy to focus on this project and do everything related to it.

Content

Content production and update

The main content of the service is located in the landing page, where searching for orienteering events happens. Most of the content on our service is from SSL (Suomen Suunnistusliitto), who gets the data from different orienteering clubs. SSL gathers the data for their Kuntosuunnistuskalenteri, which has the same idea as our service - collect all the orienteering events across Finland.

SSL has provided us with an XML-feed containing the event data, which we have then parsed to form the initial database entries for our server. For us, getting the data from SSL was great - we didn't have to worry about how to get the first data to our site. However, the data is not perfect - some events have crucial information missing (such as precise location coordinates) or some events have unclear misc information (some clubs have provided the price of the event and other details, some haven't). Usually these differences are club-related, meaning that all the club's events have the same things missing or same kind of text. Neither we nor the SSL are responsible for these imperfections - the people in the individual clubs who give the event information to SSL are responsible for typing in the correct information.

To improve the data, the SSL would have to ask the clubs to be more diligent in providing sufficient information about the events.

However, SSL collects this information from the clubs once or twice a year, so if the clubs have some ex-tempore events that were not decided before the data collection, the event won't be visible in SSL's data or Kuntosuunnistuskalenteri. Also, not all orienteering clubs in Finland give event information to SSL, as it is not compulsory, just useful. Our service was supposed to have the feature that event organizers could login and add events to our service - alas, we didn't have the time to implement it.

The event data expires after the event has ended but preserving the data might be useful for statistics and other similar purposes. For example Espoorastit keeps event data available for users for one year. However, as the main point of our service is to enable easy searching of orienteering events in the future, storing past events is not mandatory from that perspective. We said in the design document that we would consider the policy about past events later. We have learned that many users want to see their time results (how fast they collected the required "rastit" or orienteering checkpoints). They are used to the fact that the results are found on the organizer club's web pages, so we feel that there is no need to get the results to our service and show the past events on our service.

Reflection

We see that our service's content does serve all the user groups. Although, as we did not have time to finish the login and adding new events for possible organizers, one of Niilo's needs is not served.

User interface and interaction

User interface

The client's top request regarding the service was *mobile-first approach*. We agree that the latest big trends in web design are exactly mobile first (even mobile-only) and responsiveness. These requirements are fully met in our implementation, thanks to Bootstrap and the design philosophy it involved. Another focus was Flat design, which means that simplicity in graphics is emphasized and different kinds of decorative borders, shadow effects and the like are removed.

Mobile-first paradigm was implemented also by adding an Apple touch icon to the service: if the user wants to save the link for the service so that it can be accessed from the home screen just like any mobile app on his/her iOS device, he/she can click "add to Home" and it's done.

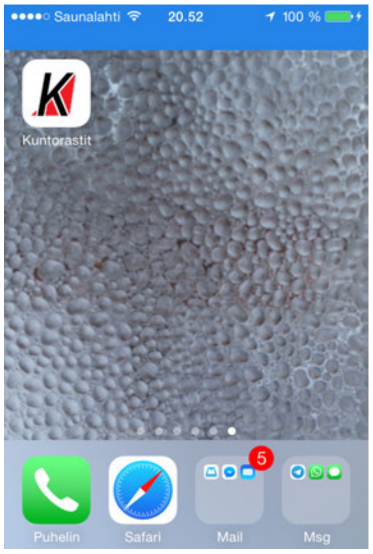
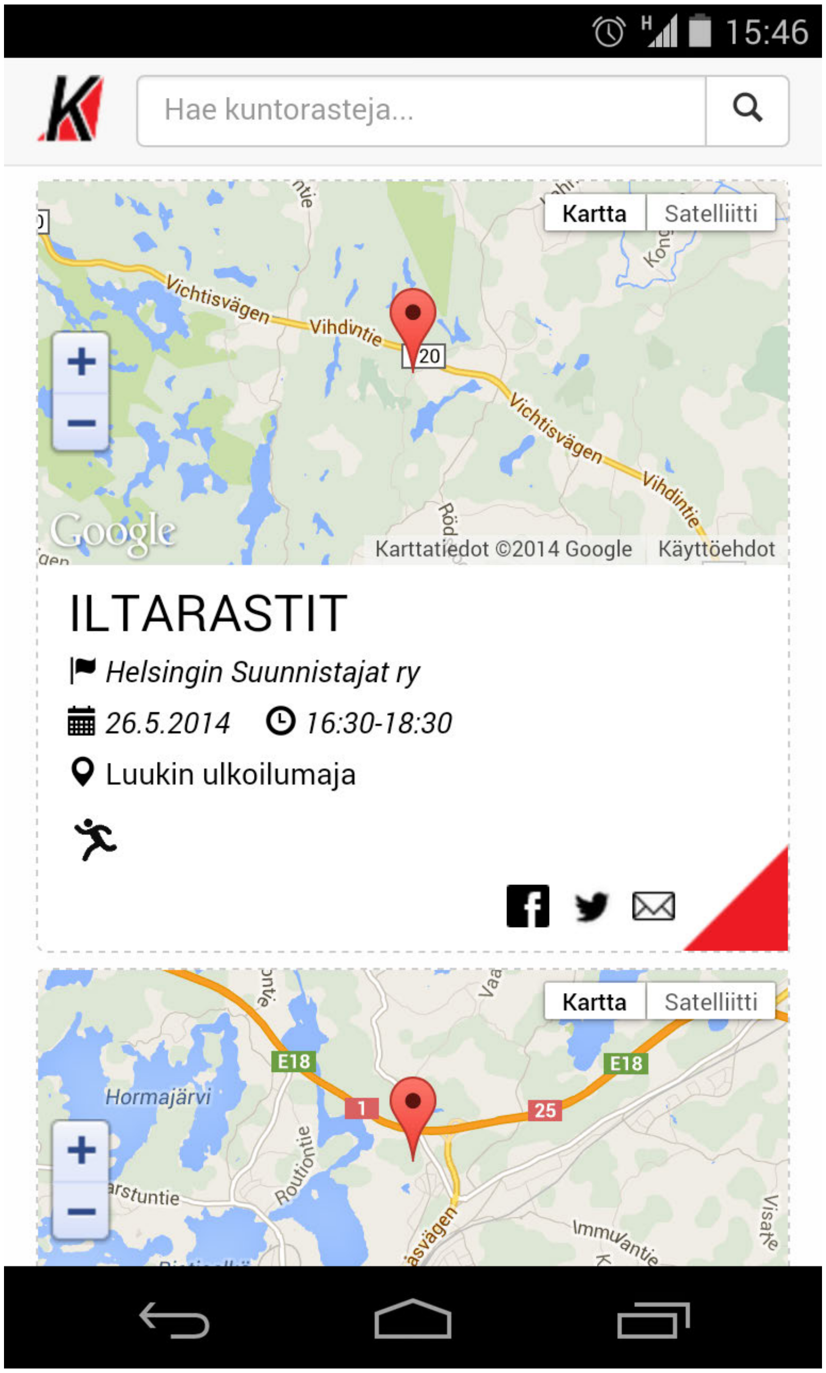


Figure 2: Kuntorastit link as an app-like home screen button on iOS 7 (iPhone 4S).

The following screenshots demonstrate the layout of the most important pages of the service.

Front page

The front page needs to get into action points as soon as possible, and also save space. Therefore the search field and logo is placed in a top bar as a fixed header. On desktop there is more space and the layout is more roomy. Also, the background image is not loaded on smaller screens since there's no use for it there, but it makes the view much more pleasant when seen on a desktop - not to mention that there usually is also more bandwidth to load the image through. This responsive design utilize Bootstrap's screen width breakpoints where the layout adapts to the size of the viewport. For example on desktop the event maps and related information is displayed next to each other but on mobile the map is as wide as the screen and the information lays under it.
When scrolling through the (automatic) search results in the mobile-view the maps are frozen so that the user won't accidentally move the map instead of the whole page, which is likely to happen since the maps are pretty big on the screen. We saw this the best solution because smaller maps would be no use and on the other hand we couldn't leave maps out since the main target group consists of orienteers, who of course are used to process location information through maps.
To provide more intuitive user experience the whole event element serves as a link to single event view, not only the title of it. This is emphasized on desktop view when the dashed border appears around the event element only when hovering the mouse cursor on it. Additionally, the dashed line is a reference to orienteering, where a dashed line means a trail in the forest.



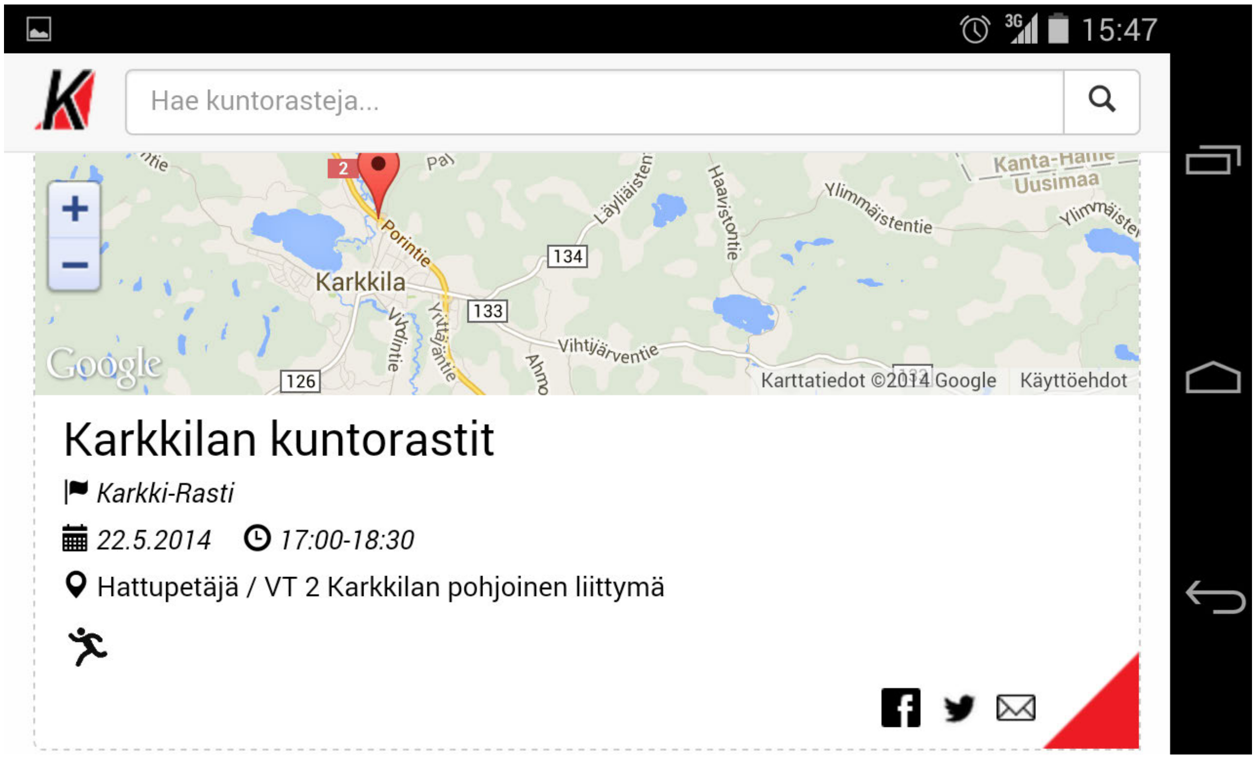
ILTARASTIT

Helsingin Suunnistajat ry

26.5.2014 16:30-18:30

Luukin ulkoilumaja

Figure 3: Front page screenshot; portrait, mobile; browser: Chrome, phone model: LG Nexus 4.


15:47

Hae kuntorasteja...

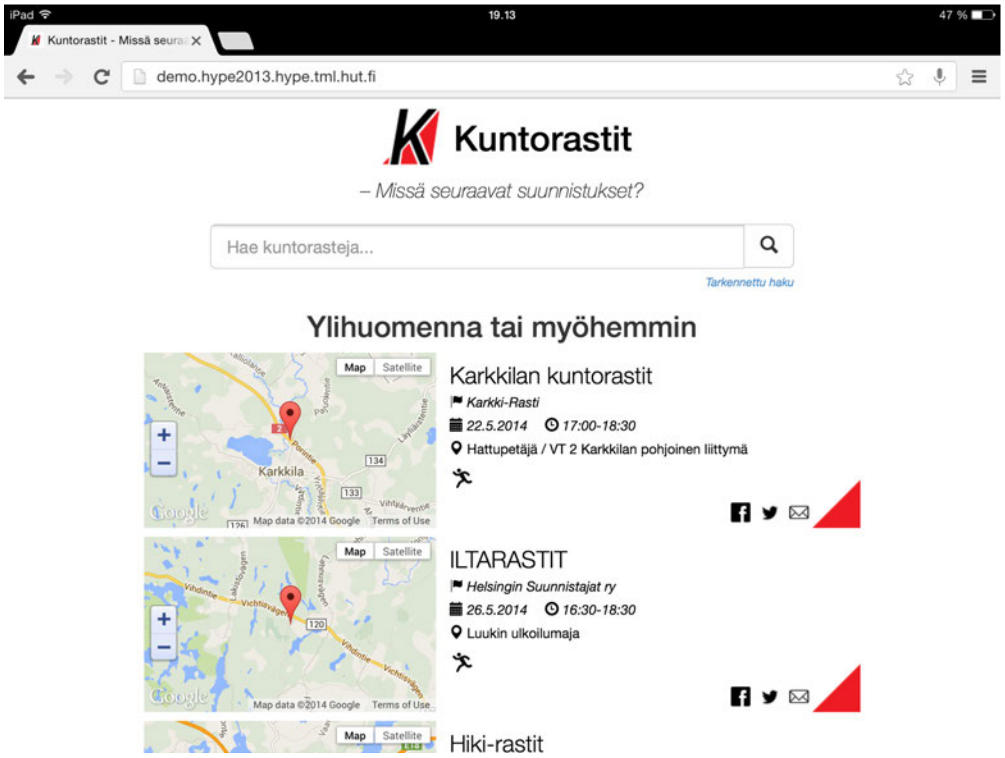
Karkkilan kuntorastit

Karkki-Rasti

22.5.2014 17:00-18:30

Hattupetäjä / VT 2 Karkkilan pohjoinen liittymä

Figure 4: Front page screenshot; landscape, mobile; browser: Chrome, phone model: LG Nexus 4.


Kuntorastit - Missä seura

demo.hype2013.hype.tml.hut.fi

Kuntorastit

– Missä seuraavat suunnistukset?

Hae kuntorasteja...

Tarkennettu haku

Ylihuomenna tai myöhemmin

Karkkilan kuntorastit

Karkki-Rasti

22.5.2014 17:00-18:30

Hattupetäjä / VT 2 Karkkilan pohjoinen liittymä

ILTARASTIT

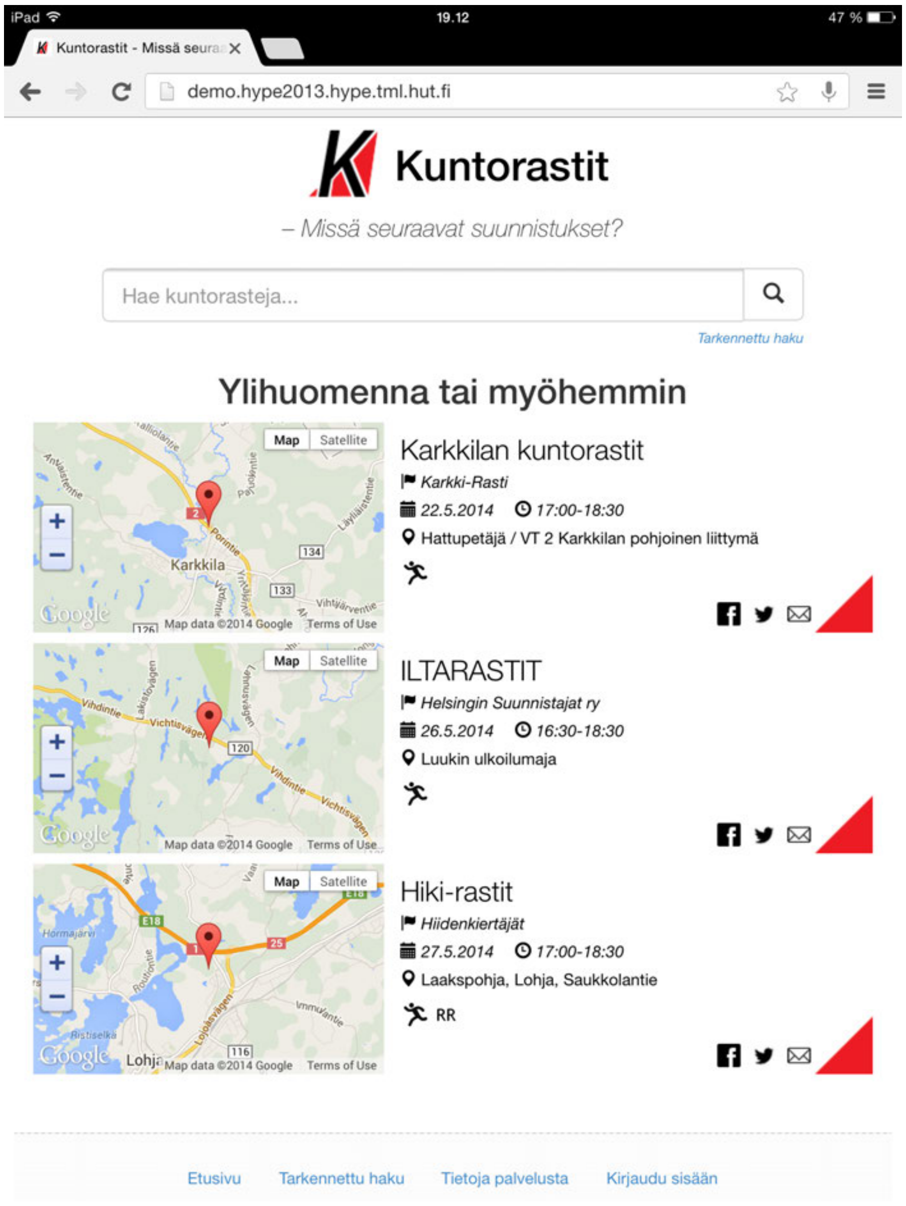
Helsingin Suunnistajat ry

26.5.2014 16:30-18:30

Luukin ulkoilumaja

Hiki-rastit

Figure 5: Front page screenshot; landscape, Apple iPad; browser: Chrome.



iPad 19.12 47 %

Kuntorastit - Missä seura

demo.hype2013.hype.tml.hut.fi

Kuntorastit

– Missä seuraavat suunnistukset?

Hae kuntorasteja...

Tarkennettu haku

Ylihuomenna tai myöhemmin

Karkkilan kuntorastit

Karkki-Rasti

22.5.2014 17:00-18:30

Hattupetäjä / VT 2 Karkkilan pohjoinen liittymä

ILTARASTIT

Helsingin Suunnistajat ry

26.5.2014 16:30-18:30

Luukin ulkoilumaja

Hiki-rastit

Hiidenkiertäjät

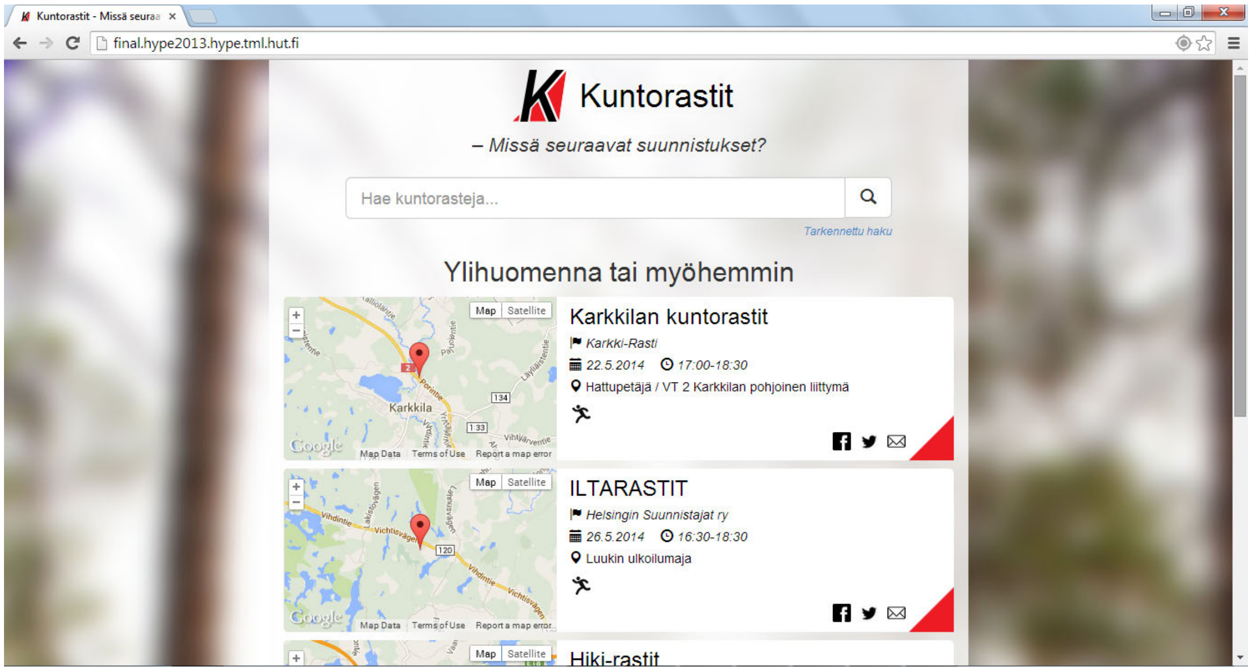
27.5.2014 17:00-18:30

Laakspohja, Lohja, Saukkolantie

RR

Etusivu Tarkennettu haku Tietoja palvelusta Kirjaudu sisään

Figure 6: Front page screenshot; portrait, Apple iPad; browser: Chrome.


Kuntorastit - Missä seuraa

final.hype2013.hype.tml.hut.fi

Kuntorastit

– Missä seuraavat suunnistukset?

Hae kuntorasteja...

Tarkennettu haku

Ylihuomenna tai myöhemmin

Karkkilan kuntorastit

Karkki-Rasti

22.5.2014 17:00-18:30

Hattupetäjä / VT 2 Karkkilan pohjoinen liittymä

ILTARASTIT

Helsingin Suunnistajat ry

26.5.2014 16:30-18:30

Luukin ulkoilumaja

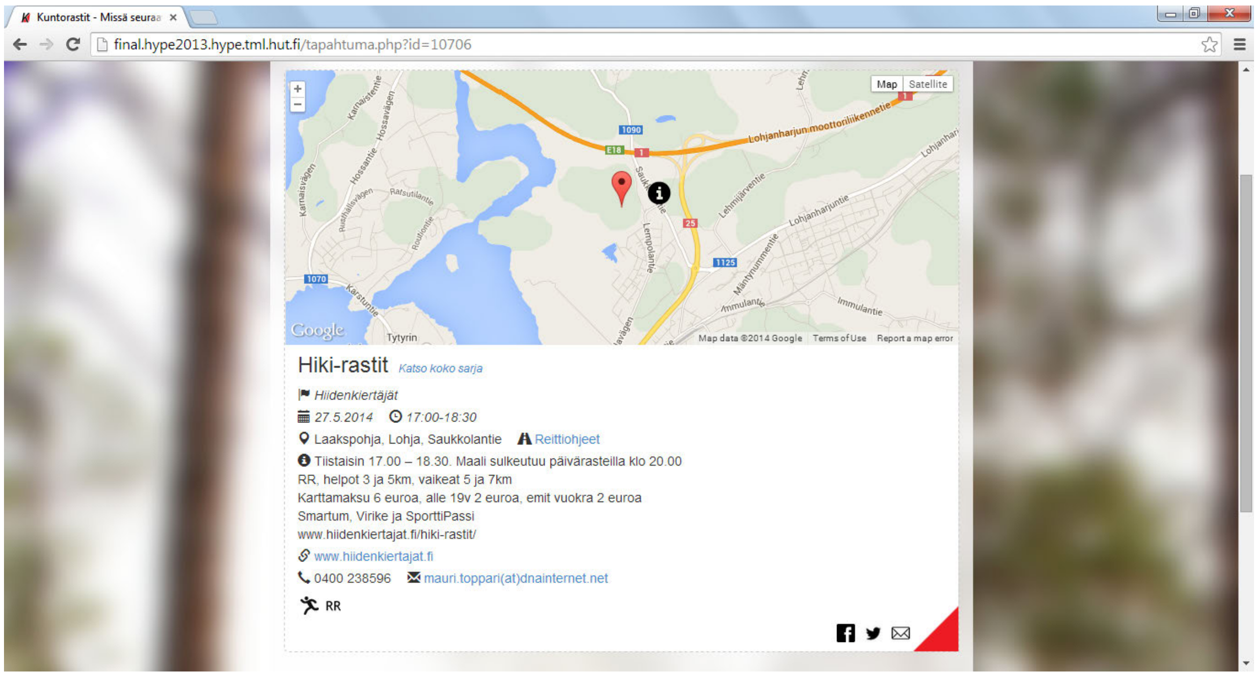
Hiki-rastit

Figure 7: Front page screenshot; landscape, desktop PC; browser: Chrome 34.0.1847.137 m, OS: Windows 7.

Single event page

Single event page gives all the information we have on a single event. The map is big and zoomable, and a link to Google's route directions service is also provided. Tooltip texts are used to explain the information title and event type symbols (mostly glyphicons), although they are pretty self-explanatory on their own. By clicking the link next to the title of the event the user can see a list of all the events that belong to that particular event series on one map and a list. There are also social sharing buttons for each event, with custom messages provided for the user.

Differences between mobile and desktop views are essentially similar as in the front page, as well as the other pages.


Kuntorastit - Missä seura
final.hype2013.hype.tml.hut.fi/tapahtuma.php?id=10706
Map
Satellite
Google
Map data ©2014 Google
Terms of Use
Report a map error

Hiki-rastit *Katso koko sarja*

Hiidenkiertäjät

27.5.2014 *17:00-18:30*

Laakspohja, Lohja, Saukkolantie Reittiohjeet

Tiistaisin 17.00 – 18.30. Maali sulkeutuu päivärasteilla klo 20.00
RR, helpot 3 ja 5km, vaikeat 5 ja 7km
Karttamaksu 6 euroa, alle 19v 2 euroa, emit vuokra 2 euroa
Smartum, Virike ja SporttiPassi
www.hiidenkiertajat.fi/hiki-rastit/

www.hiidenkiertajat.fi

0400 238596 mauri.toppari(at)dnainternet.net

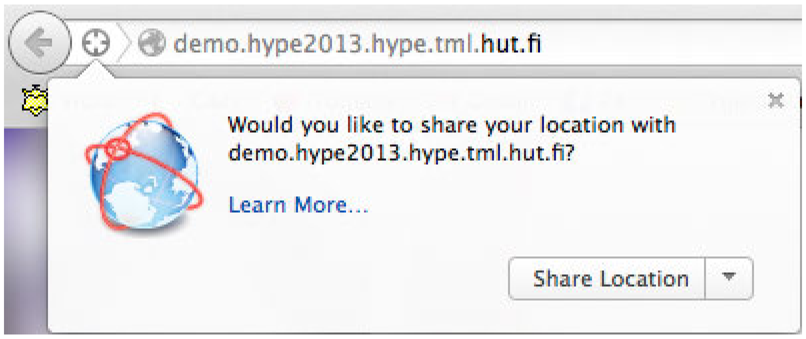
RR

Figure 8: Single event page screenshot; landscape, desktop PC; browser: Chrome 34.0.1847.137 m, OS: Windows 7.

Interaction

The most important persona for our service is Seppo. His use case 1 is presented in the following.

When Seppo first loads the website, the service figures out his computer's location. For that it needs Seppo's permission. It's up to the user's browser how this procedure goes (e.g. Figure 9). On the background of the geolocation procedure the front page already loads. Seppo allows sharing of his location to see only events that are nearby.



Would you like to share your location with demo.hype2013.hype.tml.hut.fi?

Figure 9: Geolocation alert on Firefox v 29.0.1 (Apple OS X 10.9.3).

Seppo notices that there are no events today. He sees that under the heading "Ylihuomenna tai myöhemmin" ("The day after tomorrow, or later") there is an event that looks promising: “Hiki-rastit”. He clicks on that and sees more information on a single event page (Figure 8). He zooms the map in and out and recognizes the place: it’s close to the school where his daughter goes to. Under the map Seppo notices a road symbol with the link “Reittiohjeet”. He clicks on that and is redirected to Google Maps directions tool where he can directly plan his route to the event.

Reflection

The implementation of User Interface had a few different stages, but in the end we came pretty close to what we had in mind already in the design phase. When compared to the design phase mockups there is one big difference: the finished service is much more simple, which can be seen for example in the number of columns that has reduced from two to one on home screen. Second column is thrown directly down to footer. The event element looks notably similar to the design phase mockups.

We had pretty bold visions compared to our technological capabilities, and therefore we tried to narrow down the focus and settle for MVP-like solutions for many features. The professional training, tips and questions we got from Aalto Media Factory’s Web Studio (Oliver Manner & Jon Fabritius) were very valuable. During the spring our competence grew and we were able to implement even more sophisticated solutions. Little by little we came closer to the Single Page App idea we had in mind in the beginning. This development could be carried out even further in the future.

Probably the most difficult thing in Mobile first thinking was that it forces us to concentrate on what is essential, strip and simplify the layout and focus to the core. Therefore the header of the page became very stripped and on bigger screens basically nonexistent. This is something we should’ve done very early but now it was during the last weeks that the massive unfolding header was cleared out.

We weren’t able to provide for 100% functionality on all browsers. Automatic geolocation failed occasionally on our tests on some browsers (Firefox on desktop, Safari on iOS7/iPhone 4S), but fixing those bugs went out of our time scope. The service works best on Chrome.

From our target groups’ perspective our service meets its goals. Because of limited time resources we left some of Niilo’s use cases out. Also Kati’s initial use case was left out and we currently see Kati with similar kind of use cases as Niilo but with a heavy emphasis on mobile devices. In the end we think both Kati’s and Seppo’s needs are acceptably fulfilled.

Architecture and technologies

Architecture

The service architecture is close to SPA (Single-Page App) but with a bit older components. We employed PHP from the start because of the ease of use for beginners. PHP takes care of talking with the database and searching from it. It also writes down data to the database from the XML feed it parses.

User's search queries and location information are passed out to Javascript/jQuery layer which processes the DOM to show only relevant information to the user.

Final and essential touch comes from Bootstrap, which includes CSS and Javascript/jQuery libraries. As written already on design phase, Bootstrap front-end framework [--] provides us with easier implementation of responsive page layouts, which is important since mobile platforms are our main focus. Bootstrap works on the latest releases of all major browsers, but should also degrade smoothly on older browsers such as Internet Explorer 8, which is also a good thing considering some of our users, especially Niilo, may be using a legacy browser.

Database

For our database needs we employed the PostgreSQL 8.4 that was already installed on the server. The database itself consists of two tables: "events" and "parents". The former stores data that is unique for each event, such as coordinates and timestamp, whereas the latter stores general information that applies to most or all of the events in a given series, such as a general description and contact information (thus the name "parents").

All data in the database is parsed with PHP from two separate XML-feeds provided to us directly by the SSL. We also access the database using PHP when performing queries for search results and the like.

Performance

Performance was somewhat of a challenge for us, since we targeted mainly mobile platforms, but decided to display individual Google Maps for each event when listing them on the front- and search-pages. Since loading these maps is tasking, our first implementations were rather slow. To alleviate this load, only a few events are initially fully loaded, with rest of them hidden from view. As the user scrolls down the page, these hidden events are displayed, and their maps loaded one-by-one, so that the load is spread out more evenly.

Security

Security was not discussed very deeply during the project since we left logging in out of the project scope after design phase. The customers' three security requirements remain, but only the third one actually matters. As for other security matters we continue to state that "We are to build the system with respect to each user's right to privacy. However, since we use API's, such as Google's, we cannot promise more than the 3rd party API providers do. This means, that e.g. Google can store user's route description requests for their own purposes. However, that's not what we do ourselves. Instead, we build in a way a blind system, which tailors the content for the user but doesn't store what was tailored to which IP address or so."

One security-related point that we had to consider during the project was the possibility of SQL injections, as the site provides search fields, the input to which is then used to query the database. To prevent this from happening, all queries related to the search field utilize the PHP function pg_query_params, that only allows for one non-empty query command. We also often parse or format the input somehow before performing the query. As it stands, the service should be relatively safe against SQL injections, although we didn't perform any rigorous testing to confirm this.

HTML & CSS

The service is written on HTML5 and CSS. According to W3C validator (http://validator.w3.org/) the code is not valid but the errors and warnings are not very lethal. We used Boostrap's starter template as the basis but built most of the elements - especially the important ones like the event-element - from scratch. We took advantage from for example Bootstrap's forms and inputs, as well as glyphicons and containers. In the end there was about 700 rows of (non-minified, non-Bootstrap) CSS.

Unvalid CSS deals with the use of calc() function which works on newer browsers. If the rows are discarded the results are not catastrophic, and also additional safe but not that good-looking fixed values are given. Another error was about setting a background picture of a div (a map placeholder for events with no precise location) to be justified on bottom, which is not supported by W3C. Third error group was about removing the additional padding from the left side of list elements that caused them not to center horizontally correctly. The solution works on browsers but seems not to be supported by W3C.

HTML errors came from missing alt-texts (footer logos that are mainly there for visual reasons, not necessary), meta tags that came from Bootstrap template, some empty fields that we didn't have time to look at closer, & symbols that hadn't been escaped with ;&amp but we didn’t have time to dig deeper into these bugs since they didn’t distract normal use of the service.

Browser dependencies

The service requires the HTML5 geolocation functionality (navigator.geolocation) to reach its full potential, although the search functionality and event viewing works just as well without it. A JavaScript capable browser is essentially mandatory in order to use the service.

We recommend the Google Chrome browser for viewing the site as, among the major browsers, it best supports the required HTML5 functionality, that we use to locate users. Mozilla Firefox works reasonably well, but fails to send an error if the user declines to location sharing, which breaks the front page JavaScript functionality.

Technologies and installation

The main technologies we utilized to provide the functionality of the service are PostgreSQL, PHP and JavaScript. Our technology stack is quite different to what we had planned during the design phase. This is mainly due to the fact that our previous experience of actually implementing web services was quite limited. PHP was simultaneously both simple and powerful enough for our skill limitations and back-end needs, and it was already installed on the course server, which probably saved us a lot of trouble. We also use PHP to provide quite a lot of our HTML, as majority of it is echoed using PHP scripts.

JavaScript was our front-end language as planned, but in the end we didn’t use the AngularJS framework, as Google Maps API and ordinary JavaScript were quite sufficient for our needs in the beginning. Later we did bring in the jQuery framework to hide and display search results and manipulate the DOM.

Dynamic functionality

The main dynamic functionality of our service is the orienteering event search. This is actually done in two ways. First, we have the scenario of the user entering the front page of our service. Here, a PHP script first queries our PostgreSQL database for all the events due to take place in the next two weeks. These are then echoed into HTML form. Next, JavaScript takes over and sorts the events according to their timestamp and then asks for the user’s permission to locate him/her. After the location request is resolved, the scripts limits the number of events that are initially displayed. If a location was acquired, events that take place over 60 kilometres away (in a simple circular radius) are removed from the results. The radius is based on our meeting with SSL where they stated that casual orienteer hobbyists rarely travel over 50 kilometres for an event. Finally, the script limits the number of events initially displayed down to three. More events are displayed as the user scrolls down the page.

With the actual search page, there are two ways to approach it. The first is the single search field displayed on the front page. The search query entered here is matched against multiple fields in both the event and its parent, and all substrings separated by spaces are possible for matching. This means that results obtained this way might not be optimal. For better matching, we have the improved search, which can be accessed by performing a regular search or selecting “Tarkennettu haku” from below the search field. This search page provides three search fields, the first of which is the same as with the simple search. However, the second search field allows for limitng the results by the city in which they take place (multiple cities can be searched by separating them with spaces) and the third one allows for searching events by the club that organizes them. If multiple fields are filled, the events must match all fields that had search terms entered in them. When the search-button or the enter-key is pressed, the page is reloaded, and the PHP script performs a query using the search terms provided by the user. Then JavaScript again sorts the results by timestamp, and limits the amount of initially displayed events. These search results are not limited by distance from the user. Again, more results are displayed as the user scrolls down the page.

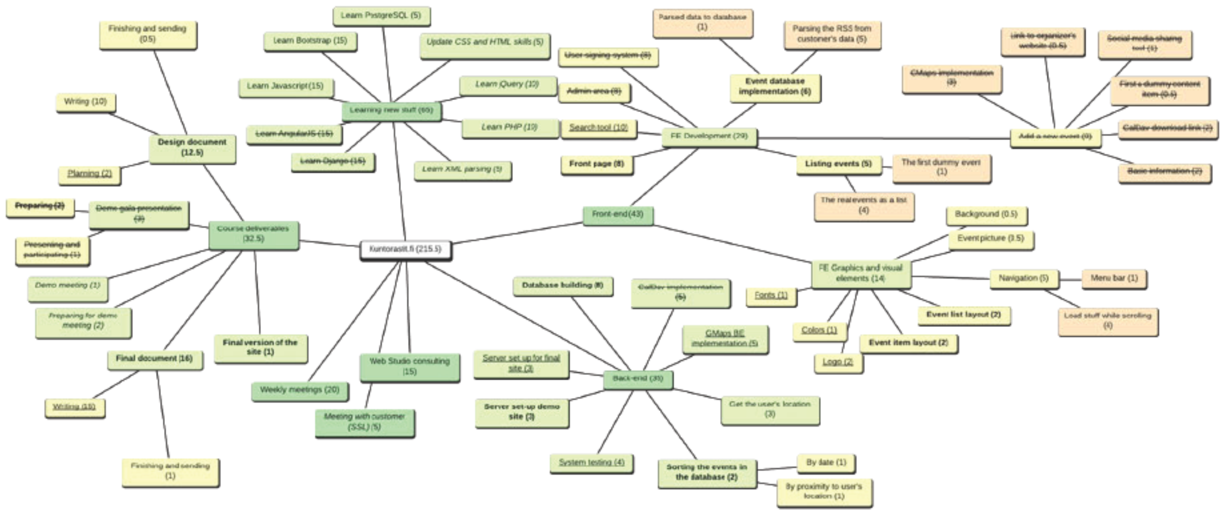
Reflection

In design phase we were struggling with understanding the differences between various Javascript libraries and choosing between different options (AngularJS, node.js etc.). In the early development phase we went down to very simple choices: PHP for back-end and just some plain javascript and jQuery for front-end. It was a good decision, although not the most elegant, modern and quick one for sure. The reason for the shift was simple: there was enough for us to learn the basics on this course. It’s almost impossible to grasp Angular if one has just written their first lines of Javascript!

If we would do this again, we would probably change one thing: switch from PHP to node.js. That would’ve strengthened our knowledge of Javascript, which seems to be pretty dominant language currently when coding web applications. Then again setting up the server with PHP was pretty quick and easy and the strong community helped us out often.

Design of the project

Work breakdown structure



Kuntorasiit.fi (215.5)
Course deliverables (32.5)
Design document (12.5)
Finishing and sending (0.5)
Writing (10)
Planning (2)
Demo gala presentation (3)
Preparing (2)
Presenting and participating (1)
Demo meeting (1)
Preparing for demo meeting (2)
Final document (16)
Writing (15)
Finishing and sending (1)
Final version of the site (1)
Learning new stuff (60)
Learn PostgreSQL (5)
Learn Bootstrap (15)
Update CSS and HTML skills (5)
Learn Javascript (15)
Learn jQuery (10)
Learn AngularJS (15)
Learn PHP (10)
Learn Django (15)
Learn XML parsing (5)
Weekly meetings (20)
Web Studio consulting (15)
Meeting with customer (SSL) (5)
Front-end (43)
FE Development (29)
User signing system (8)
Admin area (8)
Search tool (10)
Front page (8)
Event database implementation (6)
Parsed data to database (1)
Parsing the RSS from customer's data (5)
Listing events (5)
The first dummy event (1)
The real events as a list (4)
Add a new event (8)
GMaps implementation (8)
Link to organizer's website (0.5)
Social media sharing tool (1)
First a dummy content item (0.5)
CalDav download link (2)
Basic information (2)
FE Graphics and visual elements (14)
Background (0.5)
Event picture (0.5)
Navigation (5)
Menu bar (1)
Load stuff while scrolling (4)
Fonts (1)
Colors (1)
Logo (2)
Event item layout (2)
Event list layout (2)
Back-end (35)
Database building (8)
CalDav implementation (5)
GMaps BE implementation (5)
Server set up for final site (3)
Server set-up demo site (3)
Get the user's location (3)
System testing (4)
Sorting the events in the database (2)
By date (1)
By proximity to user's location (1)

Figure 10: Updated work breakdown structure of the project.

- Most important tasks **are bolded**
- Second important tasks <u>are underlined</u>
- Tasks that were not implemented are stroked through
- Tasks that were missing earlier but added to this version *are on italics*

We realised that we treated WBS like a general plan about what to do. It was the first thought-through plan about what to do, and after writing it down we felt that we had a clear picture what to do, so we never actually checked from WBS what to do next. We just discussed together what we should do next, and marked the used time to our time sheet. Because we did not use the WBS as map/plan all the time, it isn't tied to our gant diagram or time schedule.

Use of time

Below is the detailed time table of the hours used in this project. We didn't have time to make a Gant diagram for this final document, so the tasks are not shown in general level or how they distributed to different weeks.

What	**When**	[redacted]	[redacted]	[redacted]
TOTAL		**105.8**	**63.3**	**97.3**

Lectures		14	8	16
First Client meeting	2/5/2014	2	1	2
Design meeting	2/7/2014	0.3	0.3	0.3
1st Lect. assignment		2	2	0
2nd Lect. assignment		2	0	0
Design Gala preparation		4	3	2
Design Gala participation	2/27/2014	2	2	2
Independent research		2	4	5
UI design		10	0	0
Design document		5	6	5
Web studio meeting	3/14/2014	4	2	4
RSS parsing	3/18/2014	2	0	3
Logo design	3/18/2014	1	0	0
Front-end	3/20/2014	2	0	0
Web studio meeting	3/21/2014	1.5	1.5	0
Customer meeting	3/25/2014	1	1	1
UI & Graphics	3/28/2014	2	0	0
About page	3/28/2014	1	3	0
Map & location	3/28/2014	0	0	3
UI design (header)	3/31/2014	2	0	0
Fighting with XML	3/31/2014	0	0	2
Map & location	4/1/2014	0	3	3
UI design (header)	4/1/2014	2	0	0

Project management	4/2/2014	0	1	0
Back-end	4/2/2014	0	0	3
Front-end	4/2/2014	2	0	0
Separate event pages	4/3/2014	0	0	2
PHP scripts and back-end	4/3/2014	0	0	1
Front-end CSS	4/3/2014	2	0	0
XML Parsing	4/7/2014	0	0	3
Database setups	4/7/2014	0	0	1
Front-end	4/7/2014	3	0	0
Demo meeting + prep	4/8/2014	1	0	1
Front-end	4/9/2014	1	0	0
Back-end	4/14/2014	0	0	3
Preliminary search	4/14/2014	0	0	1
Front-end	4/15/2014	1	0	0
Misc	4/24/2014	0	0	1
PM meeting	4/29/2014	0.5	0.5	0.5
Front-end	4/30/2014	1.5	0	0
XML Parsing & Front-end	5/2/2014	0	0	1
Final-Database Setup	5/3/2014	0	0	1
Limited Event Displaying	5/3/2014	0	0	5
Further limiting of events	5/4/2014	0	0	2.5
Single Event Map Display	5/4/2014	0	0	1
Moving Progress to Final	5/4/2014	0	0	0.5
Front-end	5/5/2014	4	0	0
Moving Progress to Final	5/5/2014	0	0	0.5
Planning user testing	5/6/2014	0	2	0
Front-end	5/7/2014	2	0	0

User testing	5/7/2014	0	10	0
Meeting	5/8/2014	1	1	1
Front-end	5/8/2014	0	0	1.5
Front-end	5/9/2014	2	0	0
Front-end & content	5/13/2014	6	0	0
Meeting & PM	5/15/2014	2	2	2
Front-end	5/15/2014	7	0	0
JavaScript	5/15/2014	0	0	4
Front-end	5/16/2014	3	0	0
JavaScript Debug & Misc	5/17/2014	0	0	3.5
Final features	5/18/2014	0	0	2
Moving Progress to Final	5/18/2014	0	0	1
Final document writing	5/18/2014	5	10	6

Reflection

The features that prioritized as most important and important were implemented. The least important features were not implemented. We were somewhat disappointed about not implementing the least important features, but we simply didn't have enough time.

Our estimations about overall time was good, but how distributed between different tasks was not that good. Funny bugs and errors can cause a lot time that is used to fixing details, even though the most important feature is already done, but not working perfectly.

As our WBS doesn't connect to Gant diagram or time schedule, we feel that we have improving to do in our project management skills.

Final reflections

Service evaluation

Does the service work as wanted?

We feel that on the whole, our service works as we wanted it to work. There are some details we would like to hone, but the main idea and features are usable.

Is it useful to users?

We think that yes, our service is usable to users. We also got feedback from actual users in user tests, and they actually said that this is a useful service and they would use it.

Is it useful to our customer?

Yes. Suomen Suunnistusliitto was very happy about this overall idea, and they are happy with the work we have done. They are planning a major re-doing to their web pages and use of internet in marketing and communication, so this project is spot-on in their overall renewing web-strategy.

Project evaluation

Unexpected things

We had absolutely no idea how many choices there were in the technology area - it was like a wild jungle with different technologies, solutions and architecture alternatives, and we had to make the decision without having a go at one. We had help from Web Studio’s counsellors, but afterwards we realized that we should have done some things differently from what they advised (see the section about what we would have done differently).

Most challenging part

Most challenging part was keeping this all together. We had to keep everything - the idea, UI design, technologies, the needs of the users and customer, timetable and course deliverables - together and not get lost in just one part. In all other courses that we have studied, we have focused on only one or two parts, so we are used to that kind of working. On this course we had to focus on everything, and it was hard. However, we see that this is an important skill to learn, because at some point in working life we will be either supervisor or project managers and we’ll have to know how to manage all the area related to a project (although then we don’t have to do so much implementing).

What would you do differently?

We don't feel that we have that many changes to make: one big and technology related issue that we would like to change is the server side technology. We should have used Javascript on the server as well, not PHP. Another big change would be that this course or project work should be done as an intensive course - so that we do this and only this for 4 to 5 weeks. As students we have many other courses (and some have part-time jobs) that we have to do and with this kind of project work you get the best results if you're focusing all your time and resources only on this project.

Did we have project managements issues?

Most of our issues were related to communication channels - we used Scrumy, Facebook, email, SMS, phones and face-to-face meetings as communication channels. Quite often we felt that we had to think twice where was the information we were looking for, and this slowed the work. We should have had just one or two channels or information storages for this project.

What are we proud of?

We are very proud of finishing this course and learning quite hell of a lot about so many things. We are also proud of getting ourselves a *real customer* and a *real project* - it gave us much more motivation to do this service as well as we possibly could.

We are proud of our work - smooth design, readable code and user testing - and it is a pleasure to give it to our customer.

The most important thing we learned

As said earlier, this project should have been an intensive course. We learned that you get better results if you focus on one project at a time.

We feel that we learned the importance of (at least) weekly (or why not daily) face-to-face meetings, in which the team would talk about what to do next and what is working. We did not have that many of these, so the communication and work felt somehow loose and uncontrolled. The meetings would have kept everyone on track about overall schedule and their own personal task list.

We learned that group communication is hard, even with only three people. We can only imagine how it is with up to ten or more.

We feel that we learned a lot about the technological solutions and decision there are in web service projects - as students of information networks, we get to study usability, service design and UI design much more than computer science students, so the technology was the most eye-opening experience.

Last, but not least, we learned that developers/designers don't actually know how the users will feel about and use the service. We got some very good advice from the users in user testing, and we would have never thought about those things ourselves.

Possible extra points

We know that the course staff will give some extra points for parts that have been done with honors. Although we were not asked to give suggestions about these kinds of parts, we felt like highlighting some. They are the following:

Having a real customer

The course did not require that the project should have a real customer, but we had one anyway. We feel that this taught us more, but also took more time and effort than it would have done without a customer.

The relationship to the subject of this project

As only one member of our group [redacted] has done some occasional orienteering, the subject of this project made us non-users. We feel that many other groups chose a subject that was much more close to them personally - and this choice makes them users as well as developers/designers in their project. In other courses (usability and software engineering courses) we have learned that this kind position (being a developer/designer and a user at the same time) makes the team vulnerable and biased in certain thoughts and ideas - the members might feel that they have thought about everything and that everyone else will use the service in a similar way to their way of thinking. Therefore the team might overlook the need for usability or acceptance testing, and in the worst case scenario, they might have produced a service that serves only some (or none) of their intended users.

As we are non-users, we realised that we don't know really anything about how our users think, talk, operate and would use our service. We recognised the need for outside feedback and asked for it.

Having real users and user testing

We asked SSL to get us in contact with some real users (orienteering hobbyists) that could tell us what they think about our service. We got the contact information of 6 users (2 male, 4 female, all of our personas were represented in these users) and did user testing on our service. We got a lot good points about the general idea and context of use of this service - but also opinions about visual or technical details of the service.

We did not have the time to implement all the improvements the users suggested, but we will forward a list of improvements to SSL for further development.