



MOBILE FIRST AS A BEST PRACTICE IN WEB DESIGN

Otto Varrelä

Bachelor's thesis
May 2015
Degree Programme in Media

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media

VARRELA, OTTO:
Mobile First as a Best Practice in Web Design

Bachelor's thesis 41 pages
May 2015

The purpose of this thesis is to explore and explain the advantages of using mobile first design principles as a best practice in modern web design and development. This thesis examines the benefits of using mobile first in the creation of websites from the viewpoints of a designer, developer and user.

The theoretical section provides a brief examination of the development of the Internet towards the mobile revolution. The effects of the revolution are examined both now and in the near future, in order to create a better understanding of the backgrounds. The fundamentals of modern web design are presented and explained in relation to mobile first design, as well as the planning and development of content and its requirements for mobile devices. The practical part includes the application of mobile first design and principles in the form of a website project.

The research results show an exponential growth of mobile device usage and predict that the trend will continue in the future. The results of the practical application illustrate that the development process of a website becomes more efficient and straightforward by applying mobile first design practices. Using mobile devices as the starting point results in refined websites, both content and performance-wise.

The growing use of mobile devices requires the adaptation of mobile first design as the guiding principle in web design and development. Making web sites mobile-friendly and usable on all devices is imperative for user experience, but also because failure to do so will have negative penalty on the site's visibility in search engines. Embracing mobile first, as the focus on the content and usability is augmented, will ultimately result in better websites.

Key words: mobile first, mobile devices, web design, web development.

CONTENTS

1	INTRODUCTION	6
2	THEORY OF MODERN WEB DESIGN	9
2.1	Mobile first web design	11
2.1.1	Factors leading to the development of mobile-first web design.....	12
2.1.2	The design principles	13
2.1.3	Responsive layout & fluid grid	13
2.1.4	Media queries	14
2.2	Differences between conventional responsive and mobile first web design	14
2.3	Separate mobile sites vs. responsive mobile first design.....	16
2.4	Content is king	17
2.4.1	Situational content and content optimization.....	19
2.4.2	Mobile content optimization	19
2.5	Usability is everything	20
3	MOBILE FIRST DESIGN APPLIED IN PRACTICE	22
3.1	Starting point for the project	22
3.2	Concept for the site	23
3.3	The design.....	24
3.3.1	Design structure	25
3.3.2	The tools for designing.....	26
3.4	Development process	27
3.4.1	Implementing the map.....	28
3.4.2	Coding the rest of the website.....	29
3.5	Content management interface	29
3.6	Testing.....	31
3.7	Publishing the site	32
4	PERFORMANCE RESULTS	34
5	CONCLUSION	36
	REFERENCES.....	38

ABBREVIATIONS AND TERMS

HTML – Hypertext Markup Language

It is a standardized markup language for describing web pages. HTML documents consist of specific tags that are interpreted by the web browser and displayed accordingly. (W3Schools: HTML Introduction 2015.)

HTML5

Newest version of HTML, it adds many improvements to the original markup language. It was standardized in October 2014 and is intended to subsume all the earlier versions still in use. (W3Schools: HTML5 Introduction 2015.)

CSS – Cascading Style Sheets

It is a language used for describing the look and formatting of markup language (W3Schools: CSS Introduction 2015).

CSS3

Like HTML5, CSS3 is a more advanced version of the earlier CSS language. It is divided into several separate documents called modules, which are developed separately. Each module adds new capabilities and/or features, still preserving backward compatibility. (W3Schools: CSS3 Introduction 2015.)

SASS - Syntactically Awesome Style Sheets

A coding language that is intended to clarify and to speed up the writing of CSS (A list apart 2011).

JavaScript

It is the most popular programming language in the world, used alongside HTML and CSS in the creation of websites (W3Schools: Java Script Introduction 2015).

jQuery

It is a fast, small, and feature-rich JavaScript library. It simplifies JavaScript greatly and due to that fact is the most popular JavaScript library (jQuery 2015).

Foundation

It is the most advanced, responsive front-end framework in the world. The framework is mobile friendly and ready for you to customize it any way you want to use it. (Foundation 2015.)

Google Maps API

It allows the embedding and customization of Google Maps on your website (Google Developers 2015).

Wordpress

A very a popular Open Source content management system (CMS), originally designed as a blogging tool but has since evolved to suit the needs of various website types (Wordpress 2015).

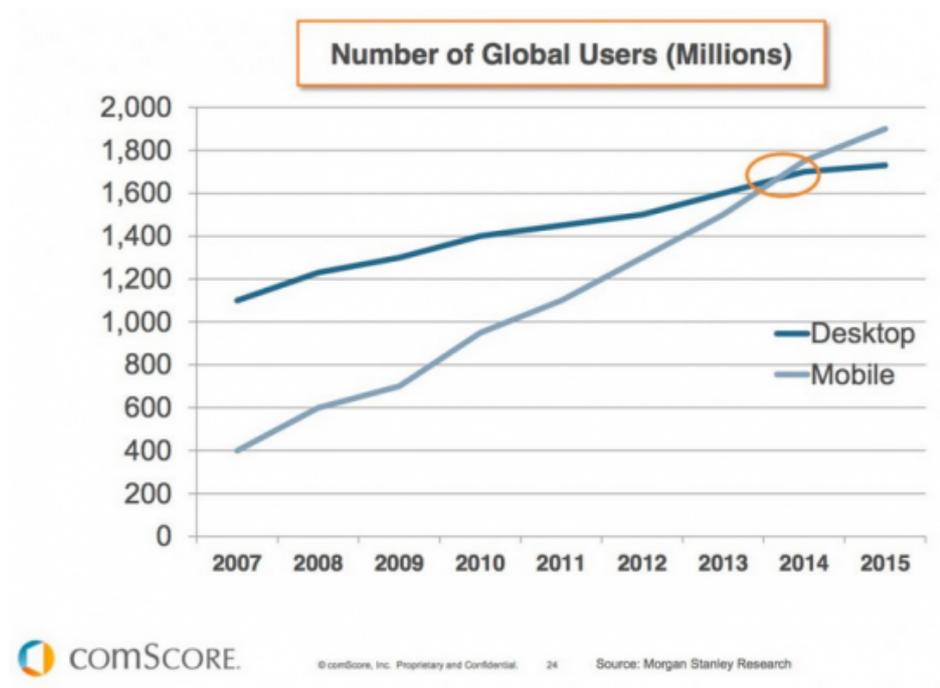
Viewport

The display area of a web browser, measured in width * height. In mobile devices this usually equals the size of the device screen. (Quirkmode: A tale of two viewports 2015.)

1 INTRODUCTION

The concept of Internet has been in existence since the 1960's, yet the term itself was only defined as late as 1995 (Internet Society). Internet is an ever-evolving network and during the recent years the usage of Internet has changed drastically and irreversibly.

Since the foundation of the World Wide Web, Internet has been mainly used on desktop computers with standardized horizontal screen sizes. All of that has changed greatly within the last 5-7 years. Now, for the first time in the history of the Internet, the global usage has shifted. Mobile device usage now exceeds that of the desktop usage; it is to say mobile app and mobile browser usage combined. (TechCrunch 2014.)



PICTURE 1. User numbers in mobile and desktop (Mobile Marketing Statistics 2015).

This shift in the usage has driven web design to evolve and to adapt; this has created new kinds of development models in order to answer to the needs of the consumers. The amount of smart-mobile devices has grown exponentially since the release of the Apple iPhone in 2007. This can be described as a turning point in the history of smart-phones and mobile networks. (Kelly 2012.) The growth is estimated to be ever increasing as the less-developed countries enter the market en-masse (GSMA 2014).

The ever-growing number of mobile devices has sparked a different approach on designing and coding websites. Instead of designing websites in the conventional way, primarily for desktop devices, the process should start from the mobile standpoint from which it is further expanded to take full usage of the increased screen size.

The design approach in question is called mobile first. Mobile first approaches website design from the standpoint that content and functionality must come first. The main argument being that starting from small and expanding to larger sizes, progressively enhancing the result, is much easier and effective. Resulting in a much faster performance over slower connections, such as mobile networks.

The aim of this thesis is to research this change in relation to web design and in particular; how it should be approached in practice. The need for more and more streamlined website design has sparked a design approach that answer's the new challenges of Internet usage.

One of the most important aspects of mobile Internet usage is speed, meaning how fast the site loads. It is the single most important aspect in a mobile website. Studies show that the time users are willing to wait for the site to load is 2-3 seconds, after which the amount of visitors that actually reach the site diminishes on average by 40%. (Patel 2014.) As it stands, it is crucial that the website has been optimized and designed so that the mobile load time is as short as possible.

A lot of different aspects affect the load time of a site and the developer can't control all of them, but the most essential is good design (Patel 2014). Redundant code and downloading content not meant for mobile devices is one of the biggest bandwidth problems. Bad code slows the site down, as the browser has to parse through the useless lines of code in order to achieve the intended result.

This thesis' argument is towards the use of mobile first web design as a best practice when designing modern websites. It is a practice that especially web designers and developers should take under careful consideration.

The thesis is based on research, experience from work and application of the principles to Savon Arkkitehtuuriopas website, arksavo.fi, developed on Wordpress. The theory

part of the thesis covers the advantages that mobile first web design offers and how it results in a better web experience. The design and development principles are covered in theory section and their usage explained, while the practical application demonstrates their benefits in practice.

2 THEORY OF MODERN WEB DESIGN

The theory of modern web design is in constant motion and new techniques are introduced constantly. It is impossible to ignore mobile devices and the effect they have in the shaping of web design. The technology is advancing so rapidly that developers are rather required to educate themselves constantly if they are to stay up-to-date.

Since the introduction of smart mobile devices, Internet usage has changed rapidly. Within the last years, Internet usage has shifted from desktop to mobile. Like other technologies, mobile device development has seen great advances, which has increased the popularity of smart mobile devices to unforeseen levels. Mobile phones and tablets have surpassed the conventional computers in popularity, as shown by picture 1, which means that the medium on which Internet is being used has changed greatly. (Murtagh 2014.)

The biggest difference has come in the form of the diminished screen size but also in the volatile nature of mobile networks and the restrictions they place on data usage. In the current market there are as many screen sizes, as there are devices. This has led to the necessity of responsive website code, which adapts to all viewport sizes in order to provide a viable user experience based on the device used.

Responsive web design was originally defined by Ethan Marcotte in 2010, it means that a website does not have a fixed layout that restrict its uses to only certain devices. In short a responsive website uses fluid grids, flexible images and media queries to scale seamlessly according to the viewport size. (Marcotte 2010.)

Although responsive principles make the site more usable in mobile landscape, it in itself by no means is the perfect answer due to the fact that if the site was originally designed for desktop, its structure isn't suited for the vertical screen. Much can be done with CSS but in the end it comes down to the core structure of the site – how the HTML tags are arranged.

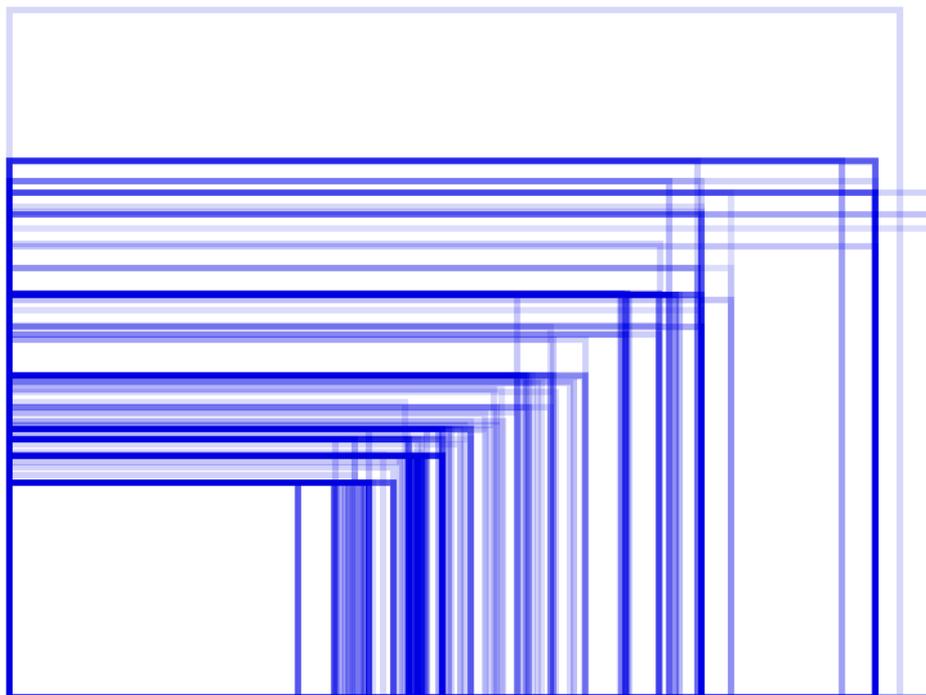
The fact that there are no standardized screen sizes and a vast amount of different operating systems in use means that these days site design requires a lot of testing with a

number of devices. (Chelariu 2013.) Even mobile devices that use Android as their platform vary from producer to producer, a functionality that works in most devices can fail to work in another version of the same operating system. In those cases a fallback is required in order to compensate for the missing functionality. This same rule also applies to older mobile devices. (Hoffman 2012.)

As can be seen from the following visualization images the amount of different size screens is immense, and underneath we still only have examples from iOS & OS X and Android devices. Nokia / Microsoft, Xiaomi etc. devices add to that number and they use completely different operating systems that behave and process data differently.

Sizes	iPhone 5	iPhone 6	iPhone 6+	iPad	iPad Retina
Devices	iPhone 5, 5C, 5S iPod Touch 5g	iPhone 6	iPhone 6+	iPad iPad 2 iPad Mini	iPad Air iPad Mini Retina
Resolution	1136 x 640	1334 x 750	2208 x 1242 (1080 x 1920) ⁴	1024 x 768	2048 x 1536
PPI	326 ppi	326 ppi	401 ppi	132 ppi	264 ppi (mini retina 326 ppi)
Logical resolution (points)	320 x 568	375 x 667	414 x 736	1024x768	1024x768

PICTURE 2. iOS screen fragmentation (Lew 2014).



PICTURE 3. Android device screen fragmentation (OpenSignal 2014).

These depictions show how varied the field is within the mobile device market, there are many producers and excruciating amount of different screen sizes. The matter comes even more complicated when you have to factor in the screen rendering on high quality displays (such as retina), for the physical width and height do not correspond with the CSS width and height. Instead they are dependent on how the points are rendered into pixels on the device and of the pixels-per-inch (PPI) value of the screen. The PPI value tells a user how many pixels fit into an inch and how large the pixels appear in real world. (The Ultimate Guide To iPhone Resolutions 2014.)

Thus, for example, the dimensions given above in the “Apple devices are resolutions” are correct for the mobile devices and yet they are incorrect from CSS standpoint. Instead in CSS the iPhone 5’s screen size is translated into 320x568 pixels, instead of the physical value of 640×1136. This due to the fact that every pixel is rendered as double the size in order to display them as designed. (Larson 2012.)

2.1 Mobile first web design

Mobile first web design was created to answer the ever-growing need of well-designed websites for mobile. The main principle being that instead of starting with the desktop layout, you start with the mobile layout from which you gradually expand the design to meet the needs of a desktop sized screen. This eliminates the problem that arises from trying to cram content that has been designed for a large horizontal screen into a small vertical screen. (Wroblewski 2011, 18.)

The fundamental logic is that if it functions on a small mobile screen, it will work on any device - no-matter how large the screen size becomes. Albeit the mobile styled site might not be that visually pleasing, on a large screen, but it will work perfectly fine. No horizontal scrollbars due to content pushing out of the viewport. (ZURB: Mobile first 2012.)

The advantages that this approach provides are even more apparent when actually coding the site. Small exception may apply due to the fact that mobile layout is simplified and trimmed down, because of the restricted display area. In practice that means that

your code is more streamlined and compact as there are fewer rules you have to overwrite - the fewer lines of code, the smaller the file size. (ZURB: Mobile first 2012.)

2.1.1 Factors leading to the development of mobile-first web design

After the launch of the iPhone in 2007 mobile network usage had exploded, in USA AT&T (sole carrier of iPhone in USA during that time) had seen a 4 932% increase in mobile data traffic by the end of 2009. Luke Wroblewski was the first person to talk about the idea of “mobile first”—web design in 2009. His research showed that mobile usage was about to increase exponentially while websites themselves were still designed mainly for desktop. The arguments for adopting the principles of mobile first approach were based on the facts:

- Mobile usage was exploding, as indicated by his research.
- Mobile design forces the designer to focus and prioritize the content, due to the small screen size – there is no room for unnecessary elements.
- The devices themselves offer new possibilities due to new capabilities, such as GPS tracking, multi-touch input etc.

(Wroblewski 2009.)

Wroblewski also released a guidebook on the subject in 2011, which has since been considered as the corner stone of the whole design approach. The book is also used as a source material in this thesis.

Brad Frost further supported the idea in a chat with Google titled: Responsive vs. Separate Sites in 2012. Frost summarized his argument as “End-users don’t care about your responsive web or your separate sites, they just want to be able to get stuff done”. That is exactly what mobile first allows; the site will usable regardless of the device. (Google Developers 2012.)

2.1.2 The design principles

Starting from small and building your way up is the main principle in mobile first web design. Concentrating on the content and how to display it in the most rational way is starting point. Ideally all the visual eye-candy is only added at the very end of the design process, keeping in mind that mobile and desktop might have different requirements in that area. As an example, touch devices don't have hover function – a function commonly used in web design. Hence for mobile the solution has to be different.

Progressively enhancing the design, while going up to larger layouts, offers the designer also the freedom to add and/or display content in a different manner compared to the mobile view. And due to the simpler nature of the mobile view, it is easy to expand the appearance into a more comprehensive layout – with heavier images (better quality) and functionalities. (Metamonks 2013.)

2.1.3 Responsive layout & fluid grid

A responsive layout is designed so that it provides an optimal viewing experience regardless of the screen size. Unlike before websites should not be designed as pixel perfect anymore into a certain fixed width, as time when screen sizes were at least a certain width is gone. Instead developers should design the 'feel' of the website, in the sense that regardless of the end user's device, the site would function and feel the same across all platforms. (Nimesh 2012.)

A fluid grid is a naturally flowing grid framework that grows or decreases in size within the dimensions of its parent element, in a HTML document the outlining parent element is `<html>`-tag. The layout is given a maximum width that defines it at its widest. It is further divided into columns, which defined with percentages. Grids are most often divided into 12 or 16 columns – thus a single column, in a grid composed of 12 columns, is 1/12 of the parent's width. As the screen size changes, the elements will be proportionally scaled to match that as well. (Nimesh 2012.)

For the grid to also function on smaller devices, the columns have to be adjusted with media queries to adequately situate the content it is holding. For example an adjustment

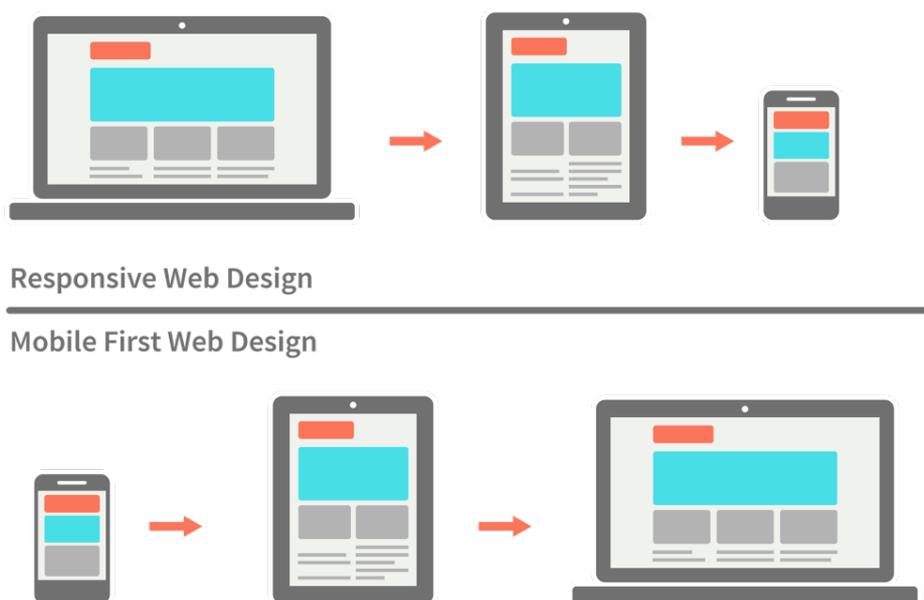
could be such that a single column on a small device will be 1/3 of the screen instead of the original 1/12. This is done in order to keep the site usable. As an example the original dimension of 1/12 out of 320px (iPhone screen width) is a really narrow column. (Nimesh 2012.)

2.1.4 Media queries

Media queries are targeted rules for different media types or devices in CSS. The query is given rules and within those rules it will target the defined media type / device and define the styles to be used for that state. As an example: “media all and (min-width: 768px)” targets all media devices with a viewport size above the designated width. Thus elements can be given rules that apply only to the designated states. (W3C 2012.)

The logic behind them is that they are always to be considered as “IF”-sentences – if the set conditions are true, then and only then will the CSS within the query be used. If the conditions don’t apply, nothing happens and the default CSS styles will be used. (Coyier 2013.)

2.2 Differences between conventional responsive and mobile first web design



PICTURE 4. Difference between the two approaches (Metamonks 2013).

All web design that adapts to the screen size is responsive web design, but there is significant difference between conventional responsive web design and mobile first web design.

Conventional responsive web design, or graceful degradation, means that a website that has been principally designed for a larger screens but is capable of adjusting the design to smaller screens. With graceful degradation the designer creates the design primarily for a desktop device. The ample screen space allows the designer a lot of freedom when it comes to the content in horizontal and vertical direction. This can lead to superfluous content, as the need to prioritize content does not play such a vital role. It means that the main message, the most crucial information, isn't displayed as effectively because the user's screen is flooded with secondary content. Another aspect of graceful degradation is that stripping down a ready product or a design can often lead to watered-down solution that does not measure up to the original. (Johnson 2013.)

Mobile first, or progressive enhancement, approaches the process from the opposite end – the sites are designed to work first on small screens. Due to the restricted space designers are required to determine what piece of information is most important and how to display it effectively. This approach often leads to a well-structured site, where the most important information is easily accessible. Focus on the content and functionality, are the most important aspects of a mobile site – the rest like visuals, although important, come to play after the first two points. (Hazzard 2014.)

These two approaches might seem quite similar, yet the differences are rather significant. As in mobile first the website you create is a product in itself – a product that functions well and is trimmed down to the most vital elements. Now expanding that product to suit the needs of larger mediums, allows you the opportunity to offer an even more robust end-result by progressively enhancing it. (Johnson 2013.)

In the case of graceful degradation content for the mobile user is being served according to the largest platform it was designed to. The problem is that the device will load all the content, whether the platform needs it or not. From there, the mobile version of the site has to ignore or hide content according to the rules set to it. This translates directly into increased loading times which, as covered earlier, is a key element when dealing with mobile sites. (Johnson 2013.)

The terms relating to responsive design are used in various ways around the Internet and new terms are being made up, to better describe certain approaches. As a standing principle the concept of responsive design is easier to grasp if it is simply defined as either graceful degradation or progressive enhancement / mobile first.

2.3 Separate mobile sites vs. responsive mobile first design

The arguments for and against separate mobile sites are varied and there really is no definitive answer. From user perspective a separate site becomes obsolete if the mobile experience provided with mobile first design is good enough. This does not mean that all sites can achieve that, at least not without completely overhauling their design.

For both cases the starting point is the same, both are created to serve a mobile device user – a mobile first design is then further expanded for desktop use. The popularity of separate mobile sites has diminished but they still exist and in some cases can offer an advantage over the regular site.

As a default a well designed mobile first website is more agile and easier to regularly update than two separate sites. One of the main arguments against a separate mobile website is the fact that updating and maintaining two sites is not cost effective and can increase workload, as the same content has to be updated for both sites and differences in the styling can cause complications. (Thomas 2013.)

Implementing new features also requires double the work, as the feature has to be coded and thought out for both versions of the site, and the site's SEO (search engine optimization) requires some extra work due to the duplicate content. (Robinson 2015.)

One of the beneficial factors of a separate site can be that it allows for a completely different design and interface – cases where the desired functionality for mobile users differs greatly from that on desktop. Also performance-wise it is often easier to serve the user mobile-optimized content separately (images, scripts and CSS) although with a sophisticated CMS it possible to dynamically create and serve the optimized content.

The separate site is commonly developed as a web application and serves a particular function or service. In many cases companies that have a complex desktop version of their site have opted for a separate mobile website. A well functioning mobile version of the site might require re-designing the site completely from scratch – which can be an extremely long and expensive project, depending of the complexity of the site. (Wroblewski 2011.)

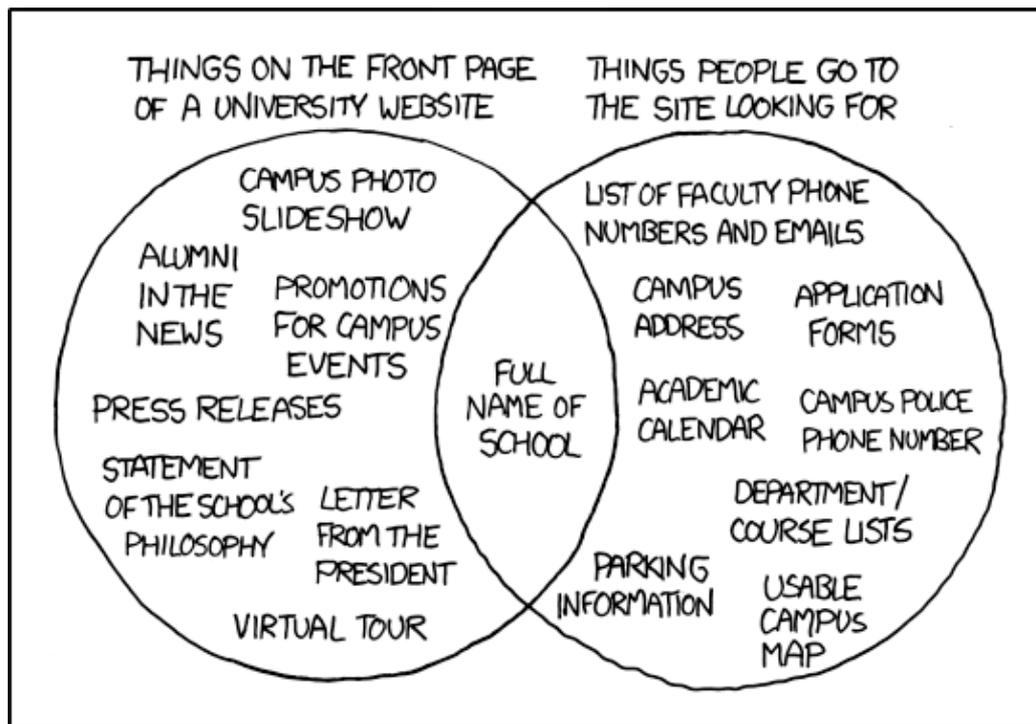
Good examples of the use of separate mobile site can be seen especially among sports betting and e-commerce companies. Also large Internet services such as Facebook, YouTube and Flickr use a separate mobile website with the content being served pre-optimized by the same servers for both mobile and desktop version. In these cases the desktop site is so complex and full of information that turning it around in responsive format for mobile would be nearly impossible while retaining good user experience. (Mozilla Developer Network 2013.)

Ideally in the future there will be no need for separate sites for different devices but till the technology is at the level it needs to be, separate mobile sites remain a viable option in certain cases. It should be emphasized that the decision to create a separate site should not be taken lightly, because of the additional workload it creates.

2.4 Content is king

Content is king, a phrase made popular by Bill Gates in 1996. It was the title of his article about the opportunities and possibilities that the revolution of Internet provides for content publishers – which in itself was prophesied by Gates to face an overhaul, as Internet allows anyone and everyone to publish their content. The key point of the article was that the new medium, Internet, will eventually take over as the main distributor of information and will engage people in a completely new way as long as the content offers them personal involvement and interest. Gates foresaw in 1996 that the easy accessibility and close to zero cost upkeep for the publisher, will create so much content that eventually everyone's interests will be served but that the differentiating aspect will be in the content itself; how it is offered and how engaging it is. Those who succeed in providing better content than others shall rule in terms of traffic. (Gates 1996.)

Much has changed since Bill Gates wrote his article but content is still the ruling and deciding element in the Internet. Google and other search engines analyze website contents and rank the pages accordingly; the better the content, the higher the site is in search results. The search algorithms are extensively complex and as an example Google uses 200 different factors to rank a site but the content is certainly one of the most important ones. (Grabowski 2014.)



PICTURE 5. A humorous depiction of extremely bad content design – which unfortunately is not that uncommon (XKCD 2010).

Google has further increased the importance of content as the search algorithm is being updated to significantly favor mobile optimized websites. It means that sites that are not mobile optimized will be in a disadvantage and will see their search position hampered on mobile devices. The updated will not affect desktop searches but as search queries done with mobile devices are estimated to surpass desktop searches in the near future, it would not make sense to ignore the direction the Internet is moving towards to. (Google 2015.)

2.4.1 Situational content and content optimization

Different platforms are used in different manners and in different situations which can lead to the necessity of prioritizing the on the site content differently on different devices. The mobile site might need a different content prioritizing than the desktop version if it is analyzed that mobile users more often than not are searching for information not accessible for them straight away. (Mads Buch Stage 2014.)

This content prioritization can be called situational design. It means that the user is served with a tailored version of the content – a version that best utilizes the available screen size and the user-behavior data on each device. New technologies offer many tools for even further customization. As an example location data can be used to determine if the client is on the move or relatively close to a store and accordingly provide the user with useful information, such as opening hours. Even a little customization can make a huge difference in user experience. (Mads Buch Stage 2014.)

Although situational design can offer a great user experience, it can also backfire and thus it is crucial to really analyze user behavior before doing drastic changes to the served content.

Situational content design should be done by prioritizing content, not by removing it because the users are as likely to need the same information on mobile and desktop – a company can never assume to know better what content their customers want. If content, accessible on other platforms, is removed, the visitor is being heavily restricted and it translates directly to the user experience of the site. Re-organizing content to subpages and hiding it behind additional taps is acceptable on mobile devices, as due to the screen size you are bound to do either that or scrolling. (Krug 2014, 148.)

2.4.2 Mobile content optimization

Regardless of device a websites load time should be short. Obviously on mobile devices the biggest factor is the mobile network in use, but under normal conditions it isn't the biggest impediment. The biggest bottleneck on mobile device comes in the form of the

device's performance – the more files and the bigger files it has to process, the slower the performance. (Johansson 2013.)

Images are often the heaviest elements on websites and account for the majority of data downloaded from a website to a browser. Rendering and scaling images to fit the screen size eat CPU (central processing unit) resources of mobile devices and thus increase the load time. It is important to serve mobile devices different images to those on desktop, primarily because the desktop images are usually immense in size, especially if compared to the needs of smartphones. Another reason is that any unnecessary pixels in images result in performance drops, albeit small separately but combined can amount to extra seconds in load time. (Grigorik 2014.)

Optimizing all the other resources (for example HTML, CSS and JavaScript files) used by the website is as important as optimization of images. Eliminating unnecessary resources and minifying (the process of removing redundant code and then compressing it) the remaining resources results in a better performing website, as the browser has less data to process and render. (Grigorik 2014.)

2.5 Usability is everything

Usability can be defined as the ability of an average user to understand and to operate the object to accomplish something efficiently (without it requiring a lot of thought). (Krug 2014, 9.)

Usability can also be considered as user-centered design – the products are built to serve the user with the best possible way. The purpose and audience of a site determine many aspects of the design and layout, thus usability is one of the first elements considered in the layout – how to effectively serve the target group. (Why Usability Is The Most Important Aspect Of Designing A Site 2013.)

As a basic concept, usability is one of the most important aspects of product design. On websites its importance is especially augmented, for Internet users aren't patient nor do they want to spend time guessing where they might find the information they are searching for. Users do not read websites; they merely scan through the sites looking for

those keywords that interest them at the time. If the site is hard to use (the navigation isn't easy to understand) or the primary content (the main message – what the site all-about) isn't self-explanatory and instantly recognizable, it will lead to users abandoning the site and searching for other, better, options. (Krug 2014, 11-23.)

On mobile devices usability has even greater significance as the users move faster and read less than on larger screens (Krug 2014, 144). Managing the content and prioritizing it means that you are bound to force the user to interact, either scroll or tap, more with the site. The user will be fine with this as long as they are certain of going to the desired direction, but due to the screen constraints the visual clues have to be sufficient enough. Otherwise the amount of mistakes the user makes while browsing increase and this in turn leads to frustration and the loss of trust towards the site. If the site is well designed it is easy for the user to keep track of his/her bearings, and to reach the desired content efficiently. (Krug 2014, 149-153.)

3 MOBILE FIRST DESIGN APPLIED IN PRACTICE

The website project for this thesis was done for the Savon Arkkitehdit SAFA (SaSAFA, the local chapter of Finnish Association of Architects). The purpose of the project was to replace the old paper format architectural guide of Kuopio with a more current and agile approach. It meant transforming and extending the paper format into a website that covers the whole Savo province (instead of just Kuopio), is easy to update and to maintain and has a map that shows the locations. The key criteria for the site were usability, both in mobile and desktop, and clarity of design.

3.1 Starting point for the project

SaSAFA had had the paper guide of architectural sites of Kuopio in existence for several years already. The first version of it was published in 2003 and a subsequent updated version in 2010. The guide contains over 180 buildings or locales (locales include parks and other such locations) with a small picture of each one and just the most basic information; information such as when the building had been built and who designed it.

As a website is not as constrained as a paper format guide is, a lot more detailed information can be added per item. Thus each building can have it's own page which allows for additional content, such as multiple images and a short history of the building. As the earlier guide was lacking additional information, it merely served those with prior information on the listed buildings. Such approach limits the user group of the guide greatly, as it offers nothing to the common visitor.

In the first meeting it was agreed that I would act as the main designer and developer for the site. The wireframe and the project plan were approved at a later date in a board meeting of SaSAFA.

3.2 Concept for the site

First it was important to determine what the site should accomplish and what the main motivations behind it were. Also determining the audience of the site was another part factor of the initial process.

The first task was to go through the paper guide and contemplate on how to best represent it in a website format. Part of that process is the search of viable benchmarks, sites that have similar functions. Benchmarking is meant as a tool for finding out how similar sites have been executed and evaluating where those sites succeed and/or fail in their design, content or usability.

An integral part of the guide is the map attached to it and it was one of the key factors when evaluating the crucial elements of the website. Adding a map to a webpage is rather simple but in this case the requirements were that it should contain all the locations of the buildings and locales, which meant hundreds of markers on the map. Also just a map with location markers is hardly satisfactory and offers the user very limited information, so ideally the markers should offer additional information on each specific item when clicking them. Beyond those requirements, the information had to be updateable.

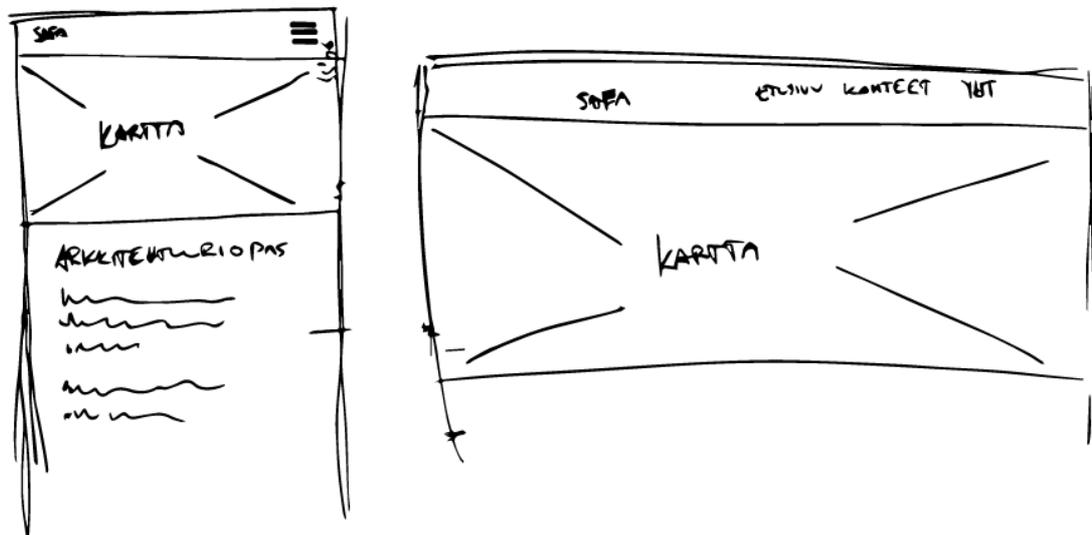
Researching the most viable method or service for achieving a specific goal can be challenging, for there are many options that are quite similar to each other but offer slightly different approaches. In my case I had a couple of options to start with and eventually decided in the favor of Wordpress.

Wordrepps allows for the implementation and management interface of a map according to the needs of the project. As the content was to be fed and updated by the members of SaSAFA, the interface had to be simple enough for a regular user. Wordpress is also easily customizable and due to its popularity it is extremely well documented. The last, a less important, factor was that it is completely free to use.

3.3 The design

As an informational website the main elements for the design of the site were the map and the listing of the buildings and locales. An architectural guide is heavily location based; so I decided that on the homepage the main element would be a map with all the locations shown with markers – the markers opening a small dialog box containing the address and link to the item page when clicked, and an introductory text for the site. This way each building or locale page would be accessible directly from the markers on the map or from the listing situated on a separate subpage.

I started the actual designing of the site by doing a few quick sketches of the wireframe on scrap paper during the initial project meeting. The purpose of these sketches was to simply record my first ideas for the site during the conversation. On the sketches I outlined the mobile view a bit and the expansions of the map and main navigation for the desktop view.



PICTURE 6. Scan of the original sketches, done during the initial project meeting.

Quick sketches are often useful, because they record the initial thoughts on the subject and can then be used as a starting point for the actual design and structure of the site.

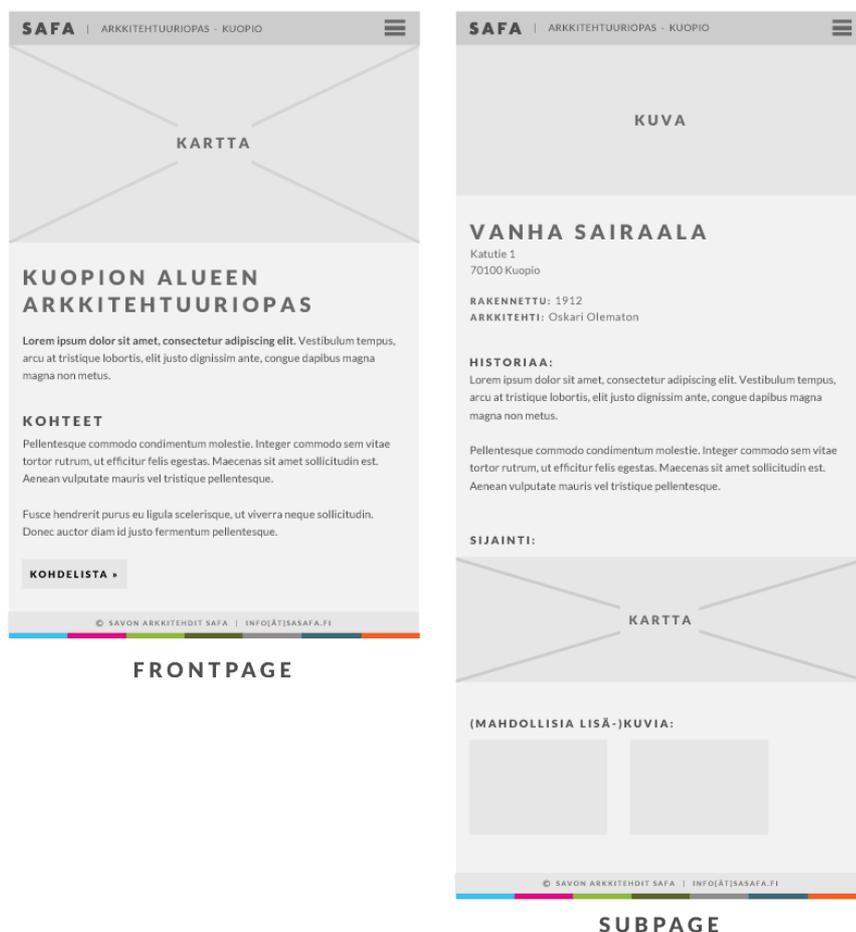
Creating the wireframe required analyzing what elements were needed on the website, and how they should be arranged for the best content flow. Content planning is important, as it has a strong bearing on the site's structure.

The goal was to keep the design light and simple, as the map in itself is a heavy element processing and loading wise. I opted for a very flat design, as it offers simplicity and requires the least amount of code hence producing less data to be downloaded.

The main structure is built into a grid that has a maximum width of 1000 pixels; it means that the elements inside the grid will never exceed 1000 pixels in width. If the viewport width is less than 1000 pixels, the elements will have a width of 100%.

Since the site was for SaSAFA, it was requested that it would contain parts of the visual identity of SAFA (. The visual elements of SAFA used on the website are the SAFA-logo (with an additional text specifying that it is the architectural guide of SaSAFA) and SAFA's brand element, a bar with six different colors.

3.3.1 Design structure



PICTURE 7. The finished mobile wireframes of the arksavo.fi website.

The front page wireframe is really close to the original sketch and the only major difference is the footer, which was not drawn to the original sketch. Navigation in the mobile design is concealed and can be toggled from the menu button. Concealing the navigation is most often the best practice due to the space requirements.

The building subpages have full width image of the building/locale on the top of the page, the point is to give a general idea of the object and catch the focus of the visitor. The image is followed by a block containing detailed information about the building (name, address, year it was built and the name of the architect), which is by formatting, parts of the text, given more hierarchical weight.

The main text content part is designed to act as the main text content (adding the background or historical description of the object). This element was missing from the original guide and without it the untutored visitors lack understandable and relatable content.

The desktop version of the design includes the same elements as the mobile version, but in a grander scale and with the navigation links visible. The adaptation of the elements to the desktop is thus painless and requires little extra work.

Work experience has enforced my view that the system of creating a complete visual layout isn't as efficient as a more wireframe-centered workflow. In that model the designer creates a wireframe and a visual composition of the website – the main elements and styling of the front- and subpage. It is then the job of the front-end web developer to create the site based on those.

Creating a working wireframe with correct typography allows for the prototyping of the site without the concentration shifting to visual elements (images etc.), which can be easily added by replacing their placeholders as I did on this site as well.

3.3.2 The tools for designing

One of the most important tools of designing anything is the ability to analyze and benchmark your design decision against other similar websites. Analyzing how other

designers and developers have achieved the same solution will often lead to a better end product and in providing a better user experience.

Deciding between what software to use for the creation of a wireframe or website layout is totally up to the designer – I find Adobe Illustrator as the most suitable tool for designing the wireframes for websites and applying needed visuals to those. One of the benefits is that all the graphics are in vector format; so using them in a responsive design is simple.

3.4 Development process

Once the suitable CMS for the project had been chosen the coding part could be started. With a multiple page website it is a good practice to use a CSS framework, which allows for a more efficient workflow. CSS frameworks offer pre-written standardized rules that can be used throughout the site and make the process faster.

Foundation 5 acts as the CSS framework of this website. It offers a good basis for website's development. The framework offers a great grid and a set of very useful styles built-in, building a custom grid is not complicated but it consumes a lot of time.

Coding according to the principles of mobile first means that the code has to be unobtrusive, whether it is HTML, CSS or JavaScript code. It is an important part of creating a well performing website on any device, for the code is more streamlined.

Part of writing unobtrusive code is the use of HTML5 semantics properly. This means that full advantage is taken of the new HTML tags introduced by HTML5. A good practice is to divide major areas in the structure with the <section> tag and nested content should then be wrapped in accordance to their nature, as an example <article> tag can be used to wrap certain kind of content. Having a clear structure also helps to keep the updating process a tad bit simpler.

CSS3 offers a lot of new possibilities for a developer. There are so many different declarations introduced via CSS3 that going into detail about them is not practical in this context. A major introduction especially useful in this project is the ability to set back-

ground images to fill the whole container – no matter what size the image or the container holding it.

In order to write cleaner and more manageable CSS, it is recommended to use a CSS preprocessor. The preprocessors in essence allow the developer to write a more sophisticated CSS code. The advantages the preprocessors offer over regular CSS are great; as an example it allows the use of variables (useful for colors, sizes etc.), reusability of rule blocks and the extension other elements styles. (1stwebdesigner 2012.)

An integral factor in the creation of lightweight website is the minification of the final code files. It means that HTML, CSS and JavaScript files are stripped of any unnecessary data – the code is compressed, so that the file size is as small as possible. (Google Developers 2015.)

3.4.1 Implementing the map

Getting the map interface working was the key factor; it was also the starting point. The idea was to have a separate post (or article) for each location so that they could be pulled as a listing on a subpage but also as individual markers on the map. The best way of achieving this result is to create a category and assign it for the posts that you would like to display on the map, also this way the visibility of posts can be easily limited.

I implemented Google Maps to the site using their JavaScript API (application program interface). Embedding the Maps interface is free and it can be customized to a great extent with the condition that Google logo and copyright are kept visible. As the map can be served marker information with JavaScript, each post is fed with the location of the building or locale in latitude and longitude coordinates along side all the information.

With PHP the coordinates, the address and the name of the item are printed into readable format inside HTML elements that are simultaneously created to the website but hidden until Maps has fully loaded. While loading Google Maps process the data from the created HTML elements and outputs it as markers on the map. To make the process

a little bit subtler I have added a loading spinner that fades away after the map is fully loaded.

3.4.2 Coding the rest of the website

Once the map was functioning, work could begin on the development of the actual structure of the site. The visual elements can be added later, after the structure is in place.

For the building/locale listing subpage a similar logic is used as on the front page. In a corresponding manner, the data from the posts is printed with PHP into HTML elements that are created simultaneously. Each HTML element gets outputted with the name of the building/locale and the main image in thumbnail format. To cut down the load time, a lazy loader script (a JavaScript method where content is loaded only when visible in the viewport) is used, it is commonly used to reduce the amount of data downloaded – Google Images is probably the most known example of its use.

The site has basically only two distinct states that differ from each other – the desktop and mobile versions. The different styling for these states is given with media queries. Otherwise the site adapts to the screen sizes perfectly.

3.5 Content management interface

The back-end of Wordpress is very user-friendly and allows the developer customization, so that those responsible for updating information to the website have an easier task.

I have customized the back-end so that when a post is given the custom category of building, the post page is updated and several additional input fields are loaded (the normal text/code editor of Wordpress obviously always present). The additional input fields are:

- Main image

The data from those fields has a distinct function in the layout of the building pages and it is outputted or formatted in a specific way. This way the website updater doesn't have to worry about giving each field a distinct id or class.

The main content for the posts is added in the text editor of Wordpress – the design includes a separate image gallery for each building/locale, which is added in the text editor.

A great part of Wordpress is that due to its vast user base there are thousands of plugins, which make it easier for a websites manager to update and control the data on the site. SEO, for example, is one of the factors that can be easily enhanced with a plugin. The plugin helps the manager to achieve the best results in search engines by simplifying and explaining the necessary information for SEO.

3.6 Testing

Before any website is launched it should be thoroughly tested by someone other than the designer or web developer, as they might be blind to their own illogical mistakes and in order to make sure that the site accessible for the audience of the website. It is of course impossible to test the website on all devices, and it is most probable that a modern website will not work on a decade old Internet Explorer 7, so applying the rule of majority is the best approach. (Maslen 2014.)

Due to the lack of the actual mobile devices, the arksavo.fi website was primarily tested for mobile devices by emulating them in Chrome developer mode, which provides a fairly accurate result.

On desktop the task is easier as the browsers themselves are the only variables. Testing the 4 biggest browsers (IE, Chrome, Firefox and Safari) in desktop mode suffice easily for the general purposes.

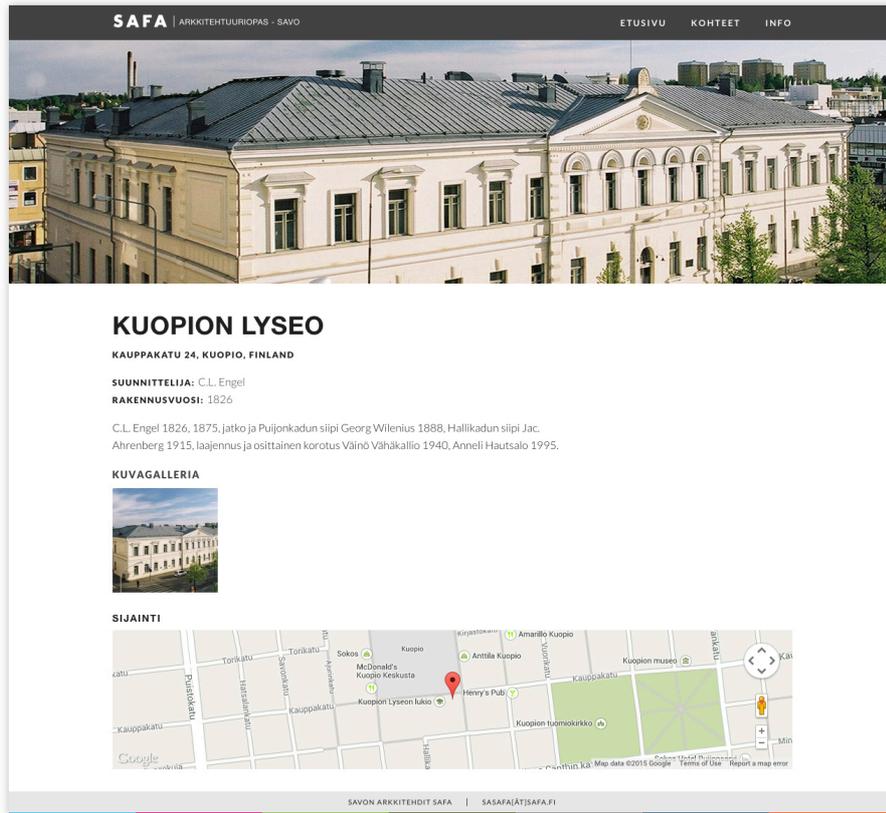
3.7 Publishing the site

After the site's technical completion the site was reviewed by SaSAFA members, who fed the original guide's content and the additional new information to the site in January and February of 2015. The site was launched in the end of February 2015.

The website's address is <http://arksavo.fi/>, it can also be reached via the subdomain savo.safa.fi. Underneath is the preview of the site in mobile and desktop views:



PICTURE 9. The finalized arksavo.fi website in mobile view.



PICTURE 10. Desktop view of the finalized arksavo.fi building page.

After the content had been fed to the site it still needed work on the SEO as it contained a lot of new content. A portion of the SEO is already accomplished by creating code that is well structured and makes good use of the semantic elements.

Currently the site is still managed by myself and there are certain additional features that might be added to the site later on. In the future the site will be managed by the members of SaSAFA.

4 PERFORMANCE RESULTS

As the objective was to build a site that is as lightweight as possible, it is interesting to test and analyze it afterwards. The tests are quite accurate but as the test servers are outside of Finland factors such as network connection, DNS lookups and server response times can weaken the results by some degree.

For analyzing the site I used Pingdom's, GTmetrix's and Google's website performance tools. The first two offer a lot of details, while Google's tool offers a more general overview, that help developers to enhance the performance of their site.

The performance ratings are a combination of tens of different variables (such as compression of code, image optimization, external resource requests etc.) that are measured while the site loads. Underneath is the breakdown of the different pages.

Front page performance report:

- The page is 870 kilobytes in size (the weight of Google Maps is around 65% of the total).
- Load time from a server in Stockholm is between 500 and 900ms. Google Maps is the heaviest element and requires the most of the load time.
- On a 3G connection the complete load time is around 3.9 seconds, which it is due to Google Maps – as the visitor sees a spinner until the map has loaded, the load time seems shorter.
- Pingdom gives a performance rating of 80/100 and GTmetrix 89/100. Both deduct points from the Google Maps load time and external requests.

Performance report for the item-listing subpage:

- Total size of the page is 4.2 megabytes, which is due to the amount of the thumbnail-images (the average thumbnail-image size is around 26 kilobytes).
- Complete site load time is around 11.3 seconds, but with the lazy load method the actual load for a user is nearly instant.
- On mobile there is no significant difference in load time.

- Since there is no Google Maps present the performance ratings are higher. Pingdom gives 96/100 and GTmetrix 99/100.

Performance report of a single building subpage:

- The values may differ between pages but on average the size is between 800 and 900 kilobytes per page.
- Page load time is between 1 and 2 seconds.
- The full-width image on top of the page is the heaviest element and takes the longest to load.
- Google Maps loads a great deal faster as it has to only process a single marker.
- The performance ratings are 86/100 from Pingdom and 91/100 from GTmetrix

From the results given by the performance tools, we can summarize that the goals of getting a well performing site were achieved almost flawlessly – as a side-note it is nearly impossible to get a perfect 100 from those tests, because perfect doesn't exist.

Arksavo.fi also compares exceptionally well to the average statistics of websites in 2014. The average size of websites in 2014 was 1,935 kilobytes, while pulling on average 95 HTTP requests (browser sending a message to the server requesting files and content) per page. (Buckler 2014.) The arksavo.fi front page is 870 kilobytes in size and there are 73 HTTP requests being made. The CSS files of the project website are only 13 kilobytes in total size, while the average in 2014 was 57 kilobytes.

One of the most important factors in the development process was the usability on mobile devices, which is close to perfection according to Google. In mobile usability Google PageSpeed Insights gives the website 98/100 for the front page and 100/100 from the subpages. On the front page 2 points were deducted due to the Google Maps zoom controls being too close together according to Google PageInsights.

5 CONCLUSION

Considering the future and how Internet usage is changing, it is imperative to understand that mobile devices will become the most important medium in the coming years. Mobile device traffic to websites can't be ignored if a website is to have relevance in the developing Internet. It can also be argued that mobile optimization should become the top priority for the majority of website owners if they aim to serve their whole audience.

For web developers and designers it means that the website's mobile version requires more focus and comprehensive planning, because its importance will keep on increasing and eventually is going to surpass the desktop version in importance – for some websites and services, mobile is already the most important channel. Making mobile the top priority does not mean undercutting the desktop version, rather it means transforming the mindset and approach when designing and developing websites.

This in mind, the arguments for the use of mobile first design in the creation of websites are compelling. Especially considering that the most important factor of a website is to be usable for the visitors – everything else in the end, is secondary. By designing according to the principles of mobile first, the website is designed from the start to work on all devices and the workflow is more natural – the design is expanded from small to large and from simple to complicated. In short, when designing with mobile first the development could be stopped at the initial level and the website would still be as usable on all devices – not as pleasing visually, but functioning perfectly.

Experience from the project and from work, support the notion that mobile first also creates slimmer websites code-wise. Once the mobile version is styled, the desktop version requires fewer changes to be overwritten than creating the site the other way around. Conventional responsive design requires overriding a big percentage of the desktop styling in order to accommodate mobile devices. This is shown in an exemplary manner in the practical project, as the sites CSS is over 75% smaller in size than the average.

Changing a mindset is challenging, but in the field of web design and development a point has been reached where it is necessary to start re-evaluating the work models in

order to keep up with the evolving landscape, because the need for mobile optimized content is growing fast. Designing with the principles of mobile first leads to better websites for both mobile and desktop in almost all aspects, since content planning and usability become top priorities. Better websites lead to a better Internet as a whole, from which all benefit.

REFERENCES

- 1st webdesigner. 2012. Understand the Power of Sass and Why You should use CSS Preprocessors. Read 18.4.2015. <http://www.1stwebdesigner.com/power-sass-why-use-css-preprocessors/>
- Bosomworth D. 15.1.2015. Mobile Marketing Statistics 2015. Read 27.1.2015. <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- Buckler C. 29.12.2014. Average Page Weight Increases 15% in 2014. Read 4.5.2015 <http://www.sitepoint.com/average-page-weight-increases-15-2014/>
- Chelariu R. 21.3.2013. Responsive Web Design With Physical Units. Read 4.2.2015. <http://www.smashingmagazine.com/2013/03/21/responsive-web-design-with-physical-units/>
- Coyier C. 22.5.2013. Logic in Media Queries. Read 12.2.2015. <https://css-tricks.com/logic-in-media-queries/>
- Demaree, D. 29.11.2011. Getting Started with Sass. Read 18.1.2015. <http://alistapart.com/article/getting-started-with-sass>
- Design Your Way. 2013. Why Usability Is The Most Important Aspect Of Designing A Site. Read 17.4.2015. <http://www.designyourway.net/dr/why-usability-is-the-most-important-aspect-of-designing-a-site/>
- Foundation. 2015. Read 21.1.2015 <http://foundation.zurb.com/>
- Gates B. 1.3.1996. Content Is King. Read 11.4.2015. <http://www.craigbailey.net/content-is-king-by-bill-gates/>
- Google. 26.2.2015. Finding more mobile-friendly search results. Read 12.4.2015. <http://googlewebmastercentral.blogspot.in/2015/02/finding-more-mobile-friendly-search.html>
- Google Developers. 14.11.2012. Chrome Mobile Monthly: Responsive vs Separate Sites. Read 10.2.2015. <https://youtu.be/SVPJcw5-WNc>
- Google Developers. 2015. Minify Resources (HTML, CSS, and JavaScript). Read 27.4.2015. <https://developers.google.com/speed/docs/insights/MinifyResources>
- Google Developers. 2015. What are the Google Maps APIs? Read 21.1.2015. <https://developers.google.com/maps/faq#whatis>
- GSMA. 6.11.2014. Half of the world's population connected to the mobile internet by 2020, according to new gsma figures. Read 27.1.2015. <http://www.gsma.com/newsroom/press-release/half-worlds-population-connected-mobile-internet-2020-according-gsma/>

Grabowski P. 14.7.2015. 30 Most Important Google Ranking Factors A Beginner Should Know. Read 20.4.2015. <http://positiononly.com/blog/seo/google-ranking-factors>

Grigorik I. 1.4.2014. Optimizing Content Efficiency. Read 29.4.2015. <https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/>

Grigorik I. 7.5.2014. Image optimization. Read 29.4.2015. <https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/image-optimization>

Hazzard F. 13.7.2014. The differences between responsive design, mobile-first design and native apps in plain language. Read 4.3.2015. <http://www.buzzquake.com/differences-responsive-design-mobile-first-design-and-native-apps-plain-language/>

Hoffman C. 20.11.2012. Why Your Android Phone Isn't Getting Operating System Updates and What You Can Do About It. Read 6.2.2015. <http://www.howtogeek.com/129273/why-your-android-phone-isnt-getting-operating-system-updates-and-what-you-can-do-about-it/>

Johansson J. 3.4.2013. How To Make Your Websites Faster On Mobile Devices. Read 28.4.2015. <http://www.smashingmagazine.com/2013/04/03/build-fast-loading-mobile-website/>

Johnson J. 5.4.2013. Mobile First Design: Why It's Great and Why It Sucks. Read 27.2.2015 <http://designshack.net/articles/css/mobilefirst/>

jQuery. 2015. What is jQuery? Read 18.1.2015 <http://jquery.com/>

Kelly H. 30.6.2015. 5 ways the iPhone changed our lives. Read 26.1.2015. <http://edition.cnn.com/2012/06/28/tech/mobile/iphone-5-years-anniversary/>

Krug S. 2014. Don't Make Me Think Revisited. A Common Sense Approach to Web and Mobile Usability. 3rd edition. San Francisco: New Riders.

Larson K. 13.9.2012. iPhone 5 Display Size and Web Design Tips. Read 9.2.2015. <http://www.kylejlaron.com/blog/2012/iphone-5-web-design/>

Leiner B., Cerf G., Clark D., Kahn R., Kleinrock R., Lynch D., Postel J., Roberts L., Wolff S. 2012. Brief History of the Internet. Read 25.1.2015. <http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet>

Lew B. 2014. iOS Resolution Quick Reference. Read 6.3.2015. <http://www.iosres.com/>

Marcotte E. 25.5.2010. Responsive Web Design. Read 21.2.2015. <http://alistapart.com/article/responsive-web-design/>

Maslen T. 14.7.2014. Prioritizing Devices: Testing And Responsive Web Design. Read 3.5.2015. <http://www.smashingmagazine.com/2014/07/14/testing-and-responsive-web-design/>

Metamonks. 2013. Why 'mobile first' is the new 'responsive'. Read 8.2.2015. <http://metamonks.com/mobile-first-vs-responsive/>

Mozilla Developer Network. 2013. Separate sites for mobile and desktop. Read 27.4.2015. https://developer.mozilla.org/en-US/docs/Web_Development/Mobile/Separate_sites

Murtagh R. 8.7.2014. Mobile Now Exceeds PC: The Biggest Shift Since the Internet Began. Read 29.1.2015. <http://searchenginewatch.com/sew/opinion/2353616/mobile-now-exceeds-pc-the-biggest-shift-since-the-internet-began>

Nimesh R. 2012. How Fluid Grids Work in Responsive Web Design. Read 11.2.2015 <http://www.1stwebdesigner.com/fluid-grids-in-responsive-design/>

OpenSignal. 22.8.2014. Android Fragmentation Visualized. Read 6.3.2015. <http://opensignal.com/reports/2014/android-fragmentation/>

Patel N. 5.6.2014. Great Expectations: 47% of Consumers Want a Web Page to Load in Two Seconds or Less. Read 29.1.2015. <http://insights.wired.com/profiles/blogs/47-of-consumers-expect-a-web-page-to-load-in-2-seconds-or-less>

PixelCut. 2014. The Ultimate Guide To iPhone Resolutions 2015. Read 6.2.2015. <http://www.paintcodeapp.com/news/ultimate-guide-to-iphone-resolutions>

Quirksmode. 2015. A tale of two viewports. Read 21.1.2015. <http://www.quirksmode.org/mobile/viewports.html>

Robinson J. 6.4.2015. Mobile SEO: Responsive Design vs. Separate Mobile Site vs. Dynamic Serving. Read 26.4.2015. <http://www.clickseed.com/seo/responsive-design-vs-separate-mobile-site-vs-dynamic-serving>

TechCrunch. 21.8.2014. Majority Of Digital Media Consumption Now Takes Place In Mobile Apps. Read 24.1.2015. <http://techcrunch.com/2014/08/21/majority-of-digital-media-consumption-now-takes-place-in-mobile-apps/>

Thomas D. 14.2.2013. Why Responsive Web Design Has To Win Out. Read 28.3.2015. <http://www.smashingmagazine.com/2013/02/14/responsive-web-design-planning-future/>

Stage M. 27.4.2014. I hate Responsive Design — and you should too! Read 2.4.2015. <https://medium.com/@madsbuchstage/i-hate-responsive-design-and-you-should-too-8cd8d18c8068>

W3C. 19.6.2012. Media Queries - W3C Recommendation. Read 13.2.2015. <http://www.w3.org/TR/css3-mediaqueries/>

W3Schools. 2015. CSS Introduction. Read 16.1.2015. http://www.w3schools.com/css/css_intro.asp

W3Schools. 2015. CSS3 Introduction. Read 16.1.2015. http://www.w3schools.com/css/css3_intro.asp

W3Schools. 2015. HTML Introduction. Read 16.1.2015.
http://www.w3schools.com/Html/html_intro.asp

W3Schools. 2015. HTML5 Introduction. Read 16.1.2015.
http://www.w3schools.com/html/html5_intro.asp

W3Schools. 2015. JavaScript Introduction. Read 17.1.2015.
http://www.w3schools.com/js/js_intro.asp

Wroblewski L. 3.11.2009. Mobile first. Read 18.2.2015
<http://www.lukew.com/ff/entry.asp?933>

Wroblewski L. 1.9.2011. Why Separate Mobile & Desktop Web Pages?. Read
26.4.2015 <http://www.lukew.com/ff/entry.asp?1390>

Wroblewski L. 2011. Mobile first. 1st edition. New York: A Book Apart.

XKCD. 30.7.2010. University Website. Read 2.4.2015. <https://xkcd.com/773/>

ZURB. 2012. Mobile first. Read 28.1.2015. <http://zurb.com/word/mobile-first>