Preferable Techniques for Showing Large Tables on Small Screens

A Qualitative Usability Study of Design Solutions

Linnea Sandelin
Abstract

Preferable Techniques for Showing Large Tables on Small Screens

Linnea Sandelin

Each day the web is becoming more and more mobile friendly, and users use smartphones and tablets to do errands on the web now more than ever. In the race towards becoming more mobile friendly, some elements are lagging behind and need more attention in order to catch up the rest of the web. One of these elements is a large table. When a table with large amounts of data is to be shown on a small screen today, many different techniques exist. Showing the entire table or removing information from the table are just two examples. In scenarios where the table content should be readable without removing any information, very few solutions exist that have a high usability. In this project, a prototype has been developed with the hope of offering a way to show large tables on small screens with a usability that is higher than what has been achieved on the web so far.
Acknowledgments

I would like to express my gratitude towards a number of people who have made this thesis project possible. To start with, a big thank you to Sammi Haj Hassine at Valtech for mentoring me throughout the project, always answering my questions, teaching me numerous new things and for always going the extra mile to help me do a better job. A big thank you to Anders Jansson at Uppsala University for giving me valuable feedback and for always helping me during times of need. Thank you Ludvig Norinder for brainstorming solutions with me, for answering many of my technical questions and for general support throughout the project. Thanks to Caroline Otterud and Teresia Schullström at Valtech for taking time out of your schedules to help me evaluate existing solutions from a usability perspective and for helping set up the user tests. Finally, a big thank you to all of the people who willingly volunteered as test users in exchange for cookies and snacks.
List of Figures

1. A large table shown in its desktop manner on a smartphone screen.  
2. A large non-responsive table showing only parts of the table content at a time.  
3. A responsive table with a fixed leftmost column allowing a user to maneuver through the remaining parts of the table.  
4. A responsive table where the user is allowed to choose which columns to display, if the user does not choose any specific columns three default columns are displayed.  
5. A responsive table showing how a user can choose which columns to display.  
6. A mode for normal usage of tables suggested by Tajima and Ohnisi with possibilities to collapse or open specific columns or rows. 
7. A mode for recording table information suggested by Tajima and Ohnisi with possibilities to copy or manipulate specific columns or rows.  
8. A mode for cell focus in tables suggested by Tajima and Ohnisi with possibilities to view specific cell information together with column or row headers that are not fixed when scrolling through a table.  
9. A browsing mode, to the left, and manipulation mode, to the right, for large tables on small screens suggested by Xu and Shi.  
10. Original table shown in the customized user interface developed by Xu et al.  
11. Table with column Singer collapsed by a user in the customized user interface developed by Xu et al.  
12. The existing prototype with a fixed leftmost column used for usability tests, the table content used is tablets.  
13. Existing prototype where the user gets to choose which columns to see, menu hidden.  
14. Existing prototype where the user gets to choose which columns to see, menu shown.  
15. The new prototype developed with computer information used for user tests.  
16. The new prototype developed with computer information used for user tests, rows removed.  
17. An example of how a table structure can be altered to show information in chunks.
List of Tables

1  Summarized results from the twelve user tests. ............. 45
Glossary

**Desktop first** is historically the most common way to develop a website. The version displayed on a desktop screen is developed first and a possible mobile version is developed after this.

**Mobile first** is a method sometimes used when developing websites. Instead of developing the desktop version of the site first, which is the traditional approach, the mobile version is first developed.

**Qualitative usability testing** refers to usability test methods where the purpose is to receive answers to questions like *why* something has a high or low usability and *how to fix* usability issues that exists with a product or system. To achieve the best results, these tests are often done while a tester and test person are both present.

**Quantitative usability testing** refers to usability test methods where the purpose is to collect opinions from many people simultaneously. Answers are often collected via forms or surveys and the hope is to receive answers to usability questions related to *how many* or *how much*.

**Responsive design** is a way to design web pages that dynamically can change and rearrange the content and appearances on the site depending on the size of the screen it is displayed on.

**Usability** ISO 9241 definition: *The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.*

**Usability testing** is a test method that can be used to decide the usability of a product or system. Different test methods exist but the overall purpose is to reveal usability problems and to find out to what level potential users are satisfied with the product or system. Usability tests are often performed during development and can lead to a redesign of products or systems, with the hope of a higher usability in the end product.
1 Introduction

In today’s society the number of people who surf the web on mobile devices such as mobile phones and tablets is increasing rapidly (Bosworth, 2015; StatCounter, 2014). This relatively new kind of usage has introduced a paradigm shift for how websites are designed and developed. From a historical viewpoint, websites have mainly been developed for users working at desktop computers with large screens, but the increased number of users surfing on small screens has called for a more mobile friendly web.

In order to meet this new requirement, different methods and techniques for how web pages are to be developed have been brought forward along with recommendations on how to adapt websites to fit on smaller screens. New methods and techniques are continuously developed and the web constantly evolves. Two different techniques have been brought forward to help with making websites more mobile friendly. Developing a special mobile version of a web page is one way to satisfy mobile users, another is to make a web page responsive so that it appears differently depending on the screen size, read more about these techniques in section 2.1. Other approaches that have been introduced lately are new methods for how to begin when developing a website. The method desktop first, where a desktop version of a web page is first develop and then this page is adapted to fit on smaller screens, is the method that historically has been used. Today, this method has been accompanied by the the new approach mobile first, where the mobile version of a website is develop first and this version is then adapted to fit on larger screens.

With the introduction of these new approaches and techniques problems naturally follow. One of the problems is how to display data and content on small screens without losing valuable information or letting the usability of the website or application suffer. One specific issue that occurs when a website is scaled down to fit on a smaller screen is how to display tables with large amounts of data. Fern et al. summarize this problem well by saying “Making tables, charts and graphs mobile friendly is like squeezing 10 pounds of sugar into a 5 pound bag” (Fern et al., 2014). Displaying information present in a large table while maintaining a high usability and readability is a problem that still needs solving for small screens.

1.1 Problem Description

To succeed with making a website mobile friendly, the presentation of content and elements on the site often differs depending on whether it is displayed on a desktop or a small screen. When a website is displayed on a small screen it is often required to rearrange content and elements on the page to maintain a high usability, in some cases removal of certain parts of a page is also needed to make the remaining parts of the page readable. Many mobile adaptions of content and elements are already present, and standards for how to design certain elements and content exist. Elements that still lack perfect mobile versions are large tables.

When a table with large amounts of data is made to fit on a small screen today it is common to remove certain columns, rows or cells or to redesign the table completely to make it usable on the new screen size, this can for example mean to convert the table into a pie chart or similar. In cases where a table
can not be stripped of information one solution is to display the table in its original desktop manner on smaller screens, another is to let the user see only small parts of the table at a time.

In cases where tables include a lot of data and cells the first scenario will result in a user being presented with a very large table shrunken down to fit on a small screen. An example of a table presented in its desktop manner is provided by the Filament group and can be seen in figure 1(Filamentgroup). The data in the table will become very small and hard to see which in turn will force the user to zoom around in the table to read its content, removing the overview of the table information. The second scenario will provide the user with a higher chance of being able to read the information present in the table, but the ability to get an overview of the information is lost from start. An example of a table presented in this matter from the Filament group can be seen in figure 2(Filamentgroup).

Figure 1: A large table shown in its desktop manner on a smartphone screen.
Other solutions include removing data or transforming the table by redesigning it. Presenting the table with some columns, rows or cells removed or as a pie chart are two common methods used to make large tables more mobile friendly, see more solutions like these in section 3.2. These methods might increase the usability for the fields and cells that remain in the table, but problems that exist with these solutions are that data is removed and the user is thereby deprived of potentially valuable information present in the original table.

With the methods that are used for showing large tables on small screens today there is no way to show all content in a table while maintaining a high usability. Whether to remove information in order to show the content that is left with a high usability or to show all table content with a low usability is a choice that website owners and creators often have to make. The question that remains is: what happens when information cannot be removed but a high
usability is still sought for?

1.2 Thesis Goal

The aim with this project is to produce one or a few prototypes that will help overcome the problems that exist regarding showing tables with large amounts of data on small screens while maintaining a high usability. The hope and intention with the prototype is that it will help with tackling some of the problems described in section 1.1. A solution that can provide a user with a good overview of information in a large table without removing data and while having a high usability can solve many of the usability problems that exist with these kinds of tables today. In order to measure if the usability of the developed prototype is higher and to see if more information is conveyed to the user than in existing solutions, usability tests will be performed on the developed solution and already existing solutions in parallel. These usability tests will help with showing whether the developed solution has a higher usability than existing solutions do, as well as showing if the developed solution provides a better overview of information present in the tables shown.

1.3 Limitations

The term mobile device often refers to computers of some kind that are small enough to be hand held. Smartphones and tablets are two technical devices that fall under this definition. In this thesis project mobile devices are stated to be the reason for a shift in internet usage and a factor that is causing the changing requirements on websites. Even if mobile devices in general are behind this shift, the focus will not be to look at large tables and their usability on mobile devices, the focus will be exclusively on smartphones. The reason for this is that tablets in general have a screen size that is much closer to the size of a desktop computer screen, due to this many of the problems discussed in this report are not as noticeable on a tablet as they are on a smartphone.

The project will not focus on investigating how many columns are needed to make a table difficult to see on a smartphone, but will focus exclusively on very large tables where usability problems exist and where table content is hard to grasp.

Accessibility is a factor that is important to take into consideration when websites are developed. To make it possible for screen readers to convey a website’s content, for example, is an important thing to think about to not exclude users. Due to the limited amount of time available for this project, accessibility will not be a factor that is taken into consideration in the developed prototype. The focus will be to try and develop a prototype that has a high usability regarding seeing and interacting with it on a touchscreen. Exploring more accessibility aspects like, for example, making it easy for screen readers to read content or to make it possible to tab through the developed prototypes in a correct order are possible extensions of this work that would be interesting and meaningful to work further with.
1.3.1 Table Content

The content in tables that are to be viewed on small screens will have great impact on how the data is to be represented. Different types of data will require different display options in order to be viewed most efficiently. For example, in cases where different columns or rows are to be compared these fields must be possible to view next to each other for an efficient comparison to be possible. On the other hand, in cases where data is to be viewed over time, much space must be available to see a time span that is as long as possible simultaneously.

To achieve the set goals for this project, it was important to find table content that in desktop view took up a lot of space on a screen, this translated to many columns and rows being required in the tables used. To make it necessary for several columns or rows to be present on the screen simultaneously it was considered appropriate to use tables where a need to compare columns or rows existed.

When these requirements were taken into consideration the table content considered most appropriate for this project was data that is to be compared internally. This means tables that include much data where different cells are to be compared column or row wise. The belief is that data with these characteristics can benefit from being displayed with the developed solution.

For the user tests and during development product comparisons have been used. In a scenario like this, each product is represented by a column or a row and a need to see several products next to each other simultaneously exists. Comparisons between different columns or rows is the main task users will perform, requiring several columns or rows to be visible simultaneously. This data was also chosen for its ease to use during user tests. Many users have used sites like these to compare different products, for example a computer, before purchasing one. Due to this, test scenarios that users can relate to can be created.

The focus of the project has not been to investigate which data types the developed solutions are most appropriate for, data types have been chosen to create scenarios with a connection to real usage among a large crowd of people.
2 Background

Different techniques and components can be used to develop a web page. But on a modern website three major parts are often present, these are HTML, CSS and JavaScript (Lane et al., 2012). These three players have different roles and purposes on a web page, and by interacting with each other they can help with making the most of a visitor’s experience on a website.

2.1 Responsive Design

From a historical perspective websites have mainly been developed for desktop computers, often implying a screen with a minimum resolution of about 640 by 480 pixels (Lane et al., 2012). With the introduction of mobile devices, such as tablets and smartphones, this limit has decreased and made it necessary to design websites that are adapted for smaller screens. Two different approaches have been brought forward to solve this problem, one solution is to design parallel websites for the different mediums and screens the website is to be displayed on. This solution often results in side effects, these include a lot of duplicated code and extra effort. The other solution that helps with overcoming some of these side effects and that is used more and more is making a website responsive (Lane et al., 2012).

A website that is responsively designed can dynamically change by rearranging content on the page depending on the size of the screen it is displayed on. Instead of having to duplicate code and make changes depending on a device’s screen size, breakpoints can be set to instruct the website about when to change its appearance. The HTML-content for a responsive website can always be the same and styling written with CSS can instead alter how the content is presented to the user and when the presentation should change, read more about this in section 2.3. (Schade, 2014; Lane et al., 2012).

2.2 HTML

HTML, *Hypertext Markup Language*, is a language used to specify a website’s content and structure. When a user visits a website through a browser the browser downloads a file, an HTML document, interprets the file and presents the information to the user as a website (Lane et al., 2012).

Markup is used to make it possible for browsers to interpret the information present on the page correctly. The browser knows how to display elements to the user by interpreting what kind of information an element is tagged as. A browser can for example interpret if an element on a page is a heading, a paragraph, an image or similar (Lane et al., 2012). Within tags it is also possible to add extra information, metadata, to the content of an element. This information can be specified using attributes and values that are added to an element in order to describe its content (Lane et al., 2012). An attribute can for example be setting a specific id for an element that later can be used to access the element via CSS or JavaScript.

Several versions of HTML have been used throughout the years, at this time HTML5 is the version most used on the web (Lane et al., 2012).
2.3 CSS

CSS, Cascading Style Sheets, is a language used to add styling to a website. It is often used to style HTML-content, but can be used together with other XML-based markup languages as well. CSS can for example be used to set fonts that are to be used, to add colors to different elements and to place or arrange objects on a website. It can also be used to apply styling to a website that differs depending on the medium or screen it is displayed on, for this purpose media queries can be used. (W3C, 2015). CSS can be written directly in HTML-code but can preferably be placed in separate files to increase the maintainability of a website by separating content and styling.

The latest version of CSS that is used today is CSS3, this version is still being developed and is not yet supported by all browsers(Lane et al., 2012).

2.3.1 Media Queries

Media queries is a concept introduced in CSS which allows developers to apply style changes to a website depending on a few different factors, for example the size of the medium displaying the website(Schade, 2014; W3C, 2012; Lane et al., 2012). An example of a style change of this kind is that content on a web page can be rearranged depending on the screen size the website is displayed on, another is that certain elements can be hidden on smaller screens(Schade, 2014). To achieve this, breakpoints are set at certain screen sizes instructing the web page to alter its appearance depending on if the screen is larger or smaller than the size defined.

2.4 JavaScript

JavaScript is a scripting language that can be used to manipulate HTML-elements or CSS on a website during user interaction(Lane et al., 2012). Data handling can be used to retrieve, send and update information synchronously or asynchronously in real time while a user is active on a website. JavaScript can for example be used to save information a user types into a website form in a cookie, this information can then be retrieved and another form requiring the same information can automatically be refilled without the user having to retype the information. (Lane et al., 2012).

2.4.1 AJAX

AJAX, Asynchronous JavaScript And XML, is a technique used for updating and adding data to a website without having to reload the entire page(W3Schools, 2015; Lane et al., 2012). By avoiding a complete reload of a website, AJAX makes it is possible to update a small part of a page with data from an external source while simultaneously allowing a user to interact with other parts of the site(Lane et al., 2012).

2.4.2 Semantic Templates

Semantic templates can be used instead of building long HTML-strings in JavaScript. When using semantic templates, HTML-like syntax can be used to create templates that are easy to read and follow and JavaScript-objects can
be inserted into these templates by adding `{{}}` around the object. One way to build these semantic templates is by using Handlebars. (Brennan, 2012). A short example of a Handlebars template can be seen below, in this scenario a JSON-array `products` with an attribute `product` for each element is used. Each of the attributes in the products-array will be rendered in a new div-element with the class `item`.

```html
<div class="row">
  <div class="product-info">
    <p>Products</p>
  </div>
  <div class="data">
    {{#products}}
      <div class="item">
        <p>{{product}}</p>
      </div>
    {{/products}}
  </div>
</div>
```
3 Theory

A factor that fairly recently has had a large impact on how web pages are to be created is the use of mobile devices such as tablets and smartphones. These devices have screens that are a lot smaller than normal desktop computers which historically have been the target medium to show websites on. As the use of these devices have grown, the web has been required to adapt in order to provide mobile users with the option of surfing on smaller devices and platforms. With this new take on the web, developers have been required to learn new methods and techniques in order to satisfy the mobile crowd.

Problems have followed from introducing new techniques and one of the problems that still needs solving is how tables with large amounts of data are to be displayed on small screens without losing usability in the process. A few attempts have been made to overcome some of these problems, some of the solutions are techniques and tricks that developers use today. These often include removing data, only showing small portions of data at a time or transforming a table by redesigning the way data is presented to a user. Other solutions are suggested in different thesis and research projects. These solutions are often developed for a certain platform or in order to overcome one or a few of the problems that exist. All of the solutions present valuable insights and ideas for this project regarding how tables with large amounts of data can be shown on small screens, what each author state is lacking in their solution and what is still needed is a solution that provides the functionality for showing large tables on small screens together with a high usability.

3.1 Problems Today

Several problems exist when it comes to showing large tables on small screens. The naive approach that historically has been used is to show tables in their desktop layout on mobile devices as well, an example can be seen in figure 1. The upside of doing this is that all information in the table is displayed to the user, the downside is that the usability will suffer greatly since the information displayed becomes extremely small and hard to read(Fern et al., 2014).

A few different approaches have been brought forward to try and overcome this problem. One approach has been to show only portions of a table at a time and allowing the user to maneuver through the remaining parts(ZURB Studios, a; Fern et al., 2014), an example is shown in figure 2. While this approach solves the problem with data becoming too small to read, other problems arise as a result. The user will not be offered a good overview of all data within a table if only small portions of a table can be seen simultaneously(Maggie, 2011). Depending on how this is done, the user might also lose sight of what information he or she is watching. Understanding what information a cell contains can be impossible if column or row headers are not fixed when navigating through the table.

Other solutions include removing the data that is considered least important and reformattting data and displaying it in other ways(Covier, 2011, 2012; Maggie, 2011). While these methods might be considered to have a higher usability since less data is rendered in the same area than in the historically used solution, problems can follow from not showing all of the original data. Read more about this in section 3.2. An example of a solution like this can be seen in figure 4.
There is no universal solution for how to handle tables with large amounts of data on small screens today. Different situations call for different approaches and the content and design will determine what existing method to use. (Coyier, 2011; Fern et al., 2014).

3.2 Methods Used by Programmers Today

A single universal solution for how to display tables with large amounts of data on small screens does not exist today, but different tricks are available for programmers to use. Some of these tricks are (Fern et al., 2014):

1. To fix one column or row and make it possible to maneuver through the remaining table content.
2. To turn each row into an own table.
3. To convert a table into a chart.
4. To hide the table by default.
5. To hide the columns considered least important.
6. To let the user decide what columns or rows should be visible.
7. To turn tables in portrait mode into landscape mode.
8. To remove the table structure and replace it with, for example, only rows.

While each of these solutions offer a way to handle large tables on small screens, they all come with some downside. Some of these solutions, 1, 2, 7 and 8, do not necessarily result in loss of data, but can instead have a low usability due the large amounts of data still displayed. Another downside is that some of these solutions can result in a poor overview of the table data since all content can not be seen simultaneously (Coyier, 2012). An example displaying solution 1 with a fixed leftmost column is provided by ZURB Studios and can be seen in figure 3 (ZURB Studios, b).
Alternatives 3, 5 and 6 tackle the problems associated with showing large tables on small screens by removing some of the data in the tables. While this solution might increase some of the usability aspects by making the content displayed easier to view and read, the fact remains that data is being removed. In solutions 3 and 5 the developers together with the website owners can decide what data to show and not show to the user, but solution 6 requires the user to make a choice about what to see, without necessarily having enough information about which information is most important. In both scenarios the user will miss out on information present in the original table though. Solution 4 falls under the same category here since all data in the table will be lost to the user if the table is not showed in the first place. An example of solution 5 and 6 is provided by the Filament group and can be seen in figure 4(Filamentgroup). In this example the user is allowed to choose which columns to display. Figure 5 shows how the
user is allowed to choose columns from a predefined list.

Figure 4: A responsive table where the user is allowed to choose which columns to display, if the user does not choose any specific columns three default columns are displayed.
3.3 Existing Solutions

Except for the different tricks used by developers today, a few other solutions have been suggested to overcome the problems associated with showing large tables on small screens.

Two of these solutions have put a lot of energy into how to make interaction with HTML table elements easier, for example simplifying the process of copying content column or row wise. One of the solutions suggests three different modes for how to view tables on small screens, one mode for normal usage, one with focus on recording and one with focus on certain cells. These different modes all provide different useful functionality. In the normal mode a user is allowed to view tables in the same manner in which they are presented today, but with an extra option to collapse or open certain columns or rows to make it easier to
see relevant content simultaneously, see figure 6. The recording mode provides a user with an option to view one column or row in a table in a new view where the information is possible to see and copy more easily, see figure 7. The cell mode offers a user possibilities to view information in a specific cell in a table next to its headers, for cases where row or column headers are not fixed for small screens, see figure 8. (Tajima and Ohnishi, 2008).

Figure 6: A mode for normal usage of tables suggested by Tajima and Ohnishi with possibilities to collapse or open specific columns or rows.

Figure 7: A mode for recording table information suggested by Tajima and Ohnishi with possibilities to copy or manipulate specific columns or rows.
migure “a h mode for cell focus in tables suggested by –ajima and vhnisi with possibilities to view specific cell information together with column or row headers that are not fixed when scrolling through a table.

The other solution is developed for the Android 2.1 platform. This solution suggests a browsing mode and a manipulation mode for tables on small screens. The browsing mode provides a custom user interface which makes it possible for a user to retrieve and compare information between columns and rows in a simpler manner than what is normally provided. In the manipulation mode a user can interact with the table data more easily, it will for example make it possible for a user to copy certain chunks of data from a table(Xu and Shi, 2011). The two different modes can be seen in figure 9.

Figure 8: A mode for cell focus in tables suggested by Tajima and Ohnishi with possibilities to view specific cell information together with column or row headers that are not fixed when scrolling through a table.

Figure 9: A browsing mode, to the left, and manipulation mode, to the right, for large tables on small screens suggested by Xu and Shi.

Another example is a plug-in developed for Microsoft Internet Explorer where the user is presented with an option to view tables with large amounts of data in a customized interface. For users using this plug-in it will be possible to collapse or show certain columns or rows in a table, similar to the normal mode in the solution suggested by Tajima and Ohnishi described above, as well as sorting it based on certain columns or rows(Xu et al.). An example of an original table shown in the customized user interface together with the same table manipulated
can be seen in figure 10 and 11. Two drawbacks for this solution mentioned by
the creators are that firstly the plug-in only works for HTML table elements,
and secondly that it is only incorporated with Internet Explorer.

![Figure 10: Original table shown in the customized user interface developed by Xu et al.](image)

![Figure 11: Table with column Singer collapsed by a user in the customized user interface developed by Xu et al.](image)

What the authors and creators of the above mentioned solutions say is still
lacking in their developed prototypes is a high usability. The suggested solutions
provide ways to work with and manipulate tables with large amounts of data
on small screens, but due to the size of the projects they have been working
on neither of them have had time to investigate and revise the usability of
the developed solutions and they therefore suggest that this is possible and
recommended to do as future work(Tajima and Ohnishi, 2008; Xu et al.).
4 Method

In order to fulfill the goals associated with this project it was concluded that several approaches and techniques were going to be required. It was established that two development rounds, one for existing solutions and one for the new prototype, were needed. From this it followed that tests were going to be needed in order to evaluate said solutions from a usability perspective. Due to the limited amount of time available for this project, two rounds of usability tests were considered too time consuming. As a result, it was decided that one round of usability tests was appropriate to use for the developed prototype alongside already existing solutions. Since the developed prototype had to be done for this, the test round was considered to be most appropriate to perform during later stages of the project. For the first round of development where existing solutions were implemented it was instead decided that discussions with usability experts working at Valtech would take place. This was done in order to find positive and negative usability aspects present with the solutions chosen, these results were then used as a basis for the prototype that was developed during the second development round.

It was decided early on that a good approach would be to divide the project into a few different parts with different focus, this was done in order to set clear milestones and goals that in the end would help with reaching the final goal for the project. More about these parts and their respective goals can be read in section 4.1.

Collecting feedback and opinions from potential users was an important part of the project to measure the usability of the developed prototype. Different usability test methods were studied in order to establish a method that would generate meaningful results for the project as a whole. The purpose of the usability tests was to collect qualitative information about the developed solution’s usability compared to existing solutions’. Due to this, a qualitative approach rather than a quantitative one was used during the tests. Motivations for why a method like this was chosen and descriptions for how the usability tests were carried out can be found in section 4.2.

4.1 Division of Work

In the beginning of this project it was concluded that it would be beneficial to divide the work into smaller parts in order to reach the final goal. These smaller parts were set up in a manner that made it possible to use them as chronological milestones throughout the project. Four main parts of the project were identified and different focuses and goals were set up for each phase.

4.1.1 Introductory Phase

The first part of the project consisted of a literature study where existing solutions for how to display large tables on small screens were investigated. This was divided into two different parts, methods used by developers today and entire solutions suggested by others. The goal during this phase was to find and extract some of the most common and appropriate methods used by programmers today and to see what had been done within this area of research so far. The methods used by developers today that were extracted were the ones who were considered
closest to fulfilling the requirements set for the solution that would later be presented as the result of this project. These requirements included that it was not allowed to remove any data from the table and it was not allowed to convert the table into, for example, a pie chart or similar. This was done in order to make the existing solution and the new prototype that were to be usability tested in parallel as similar as possible when it came to table content.

The literature study was also done in order to reveal what method to use when performing usability tests later on during the project. The goal with these tests was to get qualitative feedback about how users experience the usability of the prototype developed during implementation phase two compared to other existing solutions developed during the first implementation phase.

4.1.2 Implementation Phase 1: Existing Tricks

The methods extracted during the introductory phase were developed during the second part of the project. These solutions were then discussed with usability experts working at Valtech to see if any usability problems existed with the solutions as well as to see if any positive aspects were to be taken into consideration when developing the new solution. More information about the solutions that were developed can be found in section 5.1.1 and 5.1.2. Results from the discussion with usability experts can be found in section 6.1.1 and 6.1.2.

4.1.3 Implementation Phase 2: The New Prototype

During the third part of the project the focus was to develop a new prototype based on the results received when talking to usability experts, see section 6.1.1 and 6.1.2. The new prototype was intended to show tables on small screens, exclusively mobile phones due to their small size, with a high usability and without removing any data, read more about this implementation in section 5.2. The hope was that this would separate the suggested solution from existing ones where usability still is a big issue for the cases where all content in a table is shown, read more about this in section 3.3. The hope was also that the solution would differ from some of the existing development tricks since all data was displayed in the new prototype and from others since the usability was believed to increase, read more about existing tricks in section 3.2.

4.1.4 Final Phase

The fourth and final part consisted of finishing the project. During this phase the report was finalized and the project was presented at the University.

4.2 Usability Testing

Usability testing is a method that can be used to determine the usability of a product by allowing prospective users to use it. Usability tests often reveal problems that may exist, and can show to what level the users are satisfied or dissatisfied with the tested product. Usability tests are often performed throughout development to find possible usability problems early on, by finding issues early on developers can redesign and fix problems that are detected before a product is released on the market. This often leads to products with an increased usability and lower costs for redesign. (usability.gov, 2013a).
Different methods for usability testing exist, all with different goals and purposes. Some common factors often considered important to receive answers for though are (usability.gov, 2013b):

- **Intuitive Design**: Can the user navigate and understand the architecture effortlessly?
- **Ease of Learning**: How fast can a new user understand the interface and perform basic tasks?
- **Efficiency of Use**: How fast can an experienced user perform tasks?
- **Memorability**: Can a returning visitor with ease use it effectively?
- **Error Frequency and Severity**: How common is it that users perform errors, how severe are these errors and how can users recover from them?
- **Subjective Satisfaction**: Does the user like the product?

Many different methods exist for retrieving answers for either some or all of these questions. These methods have different goals and tests are performed differently depending on what type of information is relevant to receive. The test method that is to be used should be targeted towards the answers that are most relevant for the investigation, therefore it is important to think about what kind of answers have meaning in different scenarios. Rohrer (2014) states in his article *When to Use Which User-Experience Research Methods* that it is beneficial when performing tests to think about which of the following factors that are most well-adjusted for the specific scenario:

- **Attitudinal or Behavioral**
- **Qualitative or Quantitative**

He also states that it is always important to think about:

- **Context of Use**

According to Rohrer, *attitudinal or behavioral* often refers to the difference between what people *say* versus what they *do*. *Attitudinal* can be helpful to understand how people think and what mental models they use when they interact with, for example, an interface. These test results can be found by asking test persons questions and have them answer how they think they would react or respond in certain situations. *Behavioral* on the other hand can be seen by studying how people actually act and use a system or product. Test persons can be asked to perform certain tasks and how they achieve the goal can be studied via, for example, eye tracking or pattern discovery. Other methods to get behavioral results is to use A/B testing where two different solutions for the same functionality are presented to different random users with the purpose of finding out which of the solutions that seem most successful while users interact with them.

Whether a usability test should be *qualitative or quantitative* depends on if the importance of the test is to investigate how some users experience a system or product, or how a large crowd acts and uses a system or interface. Rohrer
suggests that qualitative test methods are good to use when the tester can be present during the test. During a test session it is possible for the tester to view the test person during usage and ask questions or adjust the test depending on how the session is going. These types of tests answer questions like why and how to fix problems during discussions with test persons. Quantitative test methods on the other hand often collect information from many test persons via, for example, surveys or other analytics tools. These results are often interpreted by computers and can not give answers about how people behave when they interact with a system or interface, but rather answers questions regarding how many and how much when it comes to how users interact with a system or interface.

Lastly, Rohrer says that it is always important for a tester to take the context of use into consideration. If the test person uses the system or interface daily and the usage comes naturally, the behavior might vary compared to a test user who sees the test object for the first time and who may only use it sporadically. (Rohrer, 2014).

4.2.1 Discussion with Usability Experts

As an alternative to using usability tests to evaluate the usability of a system or interface, an option is to let experts do the evaluation. This evaluation is sometimes called usability review. This method is not as effective as usability testing since real users and their behavior is not tested, but it can be used as a compliment in certain cases. A usability review can reveal some usability problems early on during design without being as expensive or time consuming as a usability test suite can be. In order to be most effective and fair, a usability review can be performed together with a usability test suite. (usability.gov, 2013c).

Due to a limited amount of time for this project it was not possible to perform two usability test suites. Instead, it was decided that a variant of a user review where a discussion with usability experts regarding the existing solutions were to be held, read more about these in section 3.2, and usability tests were chosen for the implemented prototype.

During the discussion the usability experts were shown the first prototype with a fixed leftmost column to begin with, read more about this in section 5.1.1, and were asked to think about positive and negative aspects of the prototype in regards to usability. A discussion followed after this and the results from this can be found in section 6.1.1. The same procedure was then repeated for the second prototype where the user gets to decide which columns he or she wants to see, read more about this in section 5.1.2. The results from this discussion can be found in section 6.1.2.

The goal with this discussion was to find possible problems and positive aspects concerning usability with existing solutions today. Some of the positive aspects were then taken into consideration to increase the new prototype’s usability, and attempts to overcome some of the usability problems were made, read more about this in section 5.2.

4.2.2 User Tests

To get feedback about the difference in usability between an existing solution, more information about prototype 1 which was used for the test can be found in
section 5.1.1, and the developed prototype, usability tests were performed on the two solutions in parallel. All of the test persons tested both solutions, but half of them started with the existing solution and the other half started with the developed prototype. The test persons did not know which of the solutions they were starting with before the tests were done.

To not make the task easier during the second try, the same kind of task was used but with differing table content. In the existing solution tablets were used and in the new prototype laptops. These two kinds of products were considered similar in the sense that both of them require a lot of technical specifications to be present in a table and the products are similar to each other even though they are not precisely the same.

Before the tests were performed each of the test persons were asked to imagine a scenario where their computer broke down and they were required to buy a new one. The test persons were asked to describe how they would proceed in a scenario like this. For those who did not use websites where products can be compared when buying a computer, the test persons were asked if they had ever used a site like www.prisjakt.nu, www.pricerunner.se or similar and if they were familiar with how the sites work.

After this the test persons were asked to try and find a product they were interested in from the prototypes. They were asked to imagine a scenario where they were using a website for product comparisons and that they had chosen tablets, for the existing solution, or laptops, for the new prototype, and received a random selection from each category. The test persons were asked to speak their thoughts out loud during usage to collect spontaneous thoughts and impressions. During the tests the users were given a few qualitative questions about the solutions they were testing, for example “can you retrieve enough information from the table to make an informed decision?” followed by questions asking them to state how easy or hard this was. After the tests the users were asked to compare the solutions to each other.

After each test was finished the results were summarized into three main points that were important to remember from that specific test. The full results from the user tests can be found in appendix A.

4.2.3 Test Persons

When the twelve test persons were chosen the first criteria was that each user had to have extensive experience with normal usage of a smartphone. This criteria was set in order for the test results to best reflect the usability of the prototypes, not how well a test person is at handling a smartphone. By choosing only experienced smartphone users for the tests the results can be presumed to relate to the usability of the developed prototypes and not to the user’s interaction with a smartphone. Including people with little or no experience with using a smartphone would have great impact on the test results and would make them hard to interpret. Comparing test results between an experienced smartphone user and an inexperienced one would require extra knowledge on how to interpret the results in order to ignore the difference in experience between the different users groups. Due to this, a more fair result was believed to be received if all test persons would have experience with smartphone usage.
5 Implementation

During the project two different implementation phases were required, more information about what these phases included and their respective goals can be found in section 4.1.2 and 4.1.3.

All of the developed prototypes, both existing solutions and the new prototype, have been developed using a combination of HTML, CSS and JavaScript, more on this in section 2. The focus has been to find and develop prototypes with a high usability on small screens, the usability on large screens has therefore not been taken into consideration.

5.1 Existing Tricks

Different techniques and tricks used by programmers today were studied in the beginning of this project, the techniques and tricks relevant for this thesis can be found in section 3.2. Two of these techniques were developed, these two were chosen due to their similarity with the prototype that was going to be developed later on in the project.

5.1.1 Fixed Leftmost Column

The first prototype to be developed was a solution provided by ZURB Studios. In this solution the leftmost column in a table is fixed on small screens while the rest of the table content is scrollable. A picture of what this solution looks like can be seen in figure 3. ZURB Studios provides the code for this solution on their website and how they have implemented their solution could therefore be studied.

This solution works on a HTML table structure and the main idea is that how a table is displayed alters depending on whether the width of the window is smaller or larger than 767 pixels. When the window is larger than this, the normal desktop version of the table is shown. When the window is smaller than this, a more mobile friendly and adapted version of the table is displayed.

When the width of the window displaying the table is smaller than 767 pixels, a copy of the table is made to begin with. By using JavaScript, different CSS-classes are then added to and removed from the original table and the copy to achieve this solution. The copied version of the table is fixed and all columns except the leftmost are hidden, this becomes the fixed leftmost column. In the original table the first column is hidden and the rest of the table content receives a property that makes the content scrollable horizontally. (ZURB Studios, b).

For the user tests tablets were used in this prototype. A random selection of tablets were picked from www.prisjakt.nu (Prisjakt.nu, a) with varying brands, price ranges and attributes to make the products as different as possible and to make it easier for users to make a decision during the tests. If all products were very similar, more research is believed to be required by the users and a choice would be harder to make. The attributes shown for every product in the table during tests were Product, Price, Screen size, Screen resolution, Operating system, 4G LTE, Review, Ranking, Camera, Weight, Memory, Storage, Microphone, Full name, Stores, In stock and Comments. How users see the table can be seen in figure 12.
migure UVa – the existing prototype with a fixed leftmost column used for usability tests, the table content used is tablets.

5.1.2 User Decides Columns

For the second prototype the idea and inspiration for how this can be made is provided by the Filament group. This solution shows a normal table in desktop layout on large screens. On smaller screens the user can decide which columns he or she wants to see (Filament group), an example can be seen in figure 4 and 5.

The solution that was implemented consisted of several HTML div-elements which each represented a column in a table. These were placed next to each other using CSS and where given unique data-id attributes. A list where the user could choose which columns to see were then implemented with checkboxes representing each column in the table. The different checkboxes corresponded to the data-id attributes given to each column, resulting in a column being shown if the corresponding checkbox was checked and vice versa. See examples of the prototypes used during the discussion with usability experts in figures 13 and 14.

![Table Example](image-url)
5.2 The New Prototype

To help increase the new prototype’s usability on small screens, the solution was developed mobile first, see section 1. A solution for mobile devices was first developed, followed by a simple table representation for desktop viewing. Media queries were used to decide which of the two solutions that was to be shown.
depending on a screen’s size. The solutions were developed using HTML, CSS and JavaScript.

The data that is shown in the tables is product information, in the scenarios chosen for user tests computers were the products used. Each computer had the attributes Product, Price, Screen size, Weight, Review, Ranking, Memory, Storage, Processor model, Full name, In stock, Stores and Comments that could be used to compare different computers to each other. Among these attributes, Product represented the computer’s name, Stores represented how many stores the computer was sold in, and Comments were other users’ opinions. A picture of each product was also shown under the corresponding product’s name. In the test scenarios chosen, Comments are not written out or shown to users, but rather the number of comments that are available and an average rating from commenting users is displayed. In real desktop scenarios this information is shown in a similar manner but it is possible to read the available comments by clicking on them, in the developed solution it is not possible to click on the comments to read further information, this also goes for the attribute Stores which in real scenarios would be possible to click on to see which stores the product is sold in.

The information about each computer was retrieved as a random selection from www.prisjakt.nu (Prisjakt, b) and was then represented in JSON-format in the following manner:

```
products: [{
  product: 'Apple MacBook Air',
  picture: 'http://s3.pji.nu/product/standard/800/937232.jpg',
  price: 8999 + ':- ',
  in_stock: 'No',
  stores: 2,
  review: '9/10',
  ranking: 955,
  memory: 2 + ' GB',
  storage: 64 + 'GB SSD',
  processor_model: 'Intel Core i5 2467M',
  full_name: 'Apple MacBook Air – 1,6GHz DC 2GB 64GB 11,6”',
  weight: 1.08 + ' kg',
  screen_size: '11.6 + ''',
  comments: '8+ 3'
},

  product: 'Asus Zenbook',
  picture: 'http://s3.pji.nu/product/standard/800/2209863.jpg',
  price: 15001 + ':- ',
  in_stock: 'No',
  stores: 11,
  review: '8/10',
  ranking: 800,
  memory: 8 + ' GB',
  storage: 256 + 'GB SSD',
  processor_model: 'Intel Core i7 4500U',
```
full_name: 'Asus Zenbook UX302LG-C4014H'
weight: 1.5 + ' kg'
screen_size: 13.3 + ''
comments: 1
}
{
...
}
];

To build the developed solution, HTML *div-elements* were used to structure the table. A table container was created and CSS was used to give the container a *flexbox* layout. Within this container, rows were then created and items were rendered as column cells. Templates written using *Handlebars* were used to render each product’s information present in the JSON-array *products*. Every attribute, for example the price for every product, was rendered as a row, resulting in every unique attribute being rendered as a flexbox-item. As a result, every product was displayed as a column. Each of the elements in a column were given an *order* representing the index the product had in the JSON-array. A user has the option to move a column to the left or right in this solution. When a user clicks on the markers indicating if the column is to be moved left or right, the order-attribute is changed using JavaScript to show the columns in a new desired order. A picture of the developed solution can be seen in figure 15.

In the developed solution a user has the option to collapse specific rows directly in the table by clicking on a cross next to the row header, see figure 15. When a user clicks on this cross, JavaScript is used to hide the specific row by applying a CSS-class *hidden* with the attribute *display: none* to it. When a row is collapsed, a list appears above the table with the removed row’s name, see figure 16. If more rows are collapsed, the list grows with the removed attributes. When the attributes representing removed rows are clicked, the rows are shown in the table again and removed from the list. To match a list element to a specific row, matching *data-id*-attributes and JavaScript was used. When the list with removed attributes is shown for the first time and when list elements are added to it, the table is pushed down on the website. To not confuse the user by moving him or her around in the table when collapsing a row, JavaScript was used to scroll to the same place in the table that the user was located at before he or she collapsed the row.

A minimum width was set for the leftmost column containing headers for what each row represents to make sure that the column does not shrink too much and to make the content in the headers readable. To give the column an appropriate size in the surrounding container the attribute *flex: 0 0 20%* was assigned to it. By doing this, the column was set not to grow or shrink, and the initial width of the column should be 20% of the surrounding container.

In order to make the remaining columns scrollable from left to right and vice versa, the remaining columns were assigned a class where the CSS-attribute *overflow-x* was set to *scroll*. Since Handlebars were used to render all of the data row-wise, the flexbox-layout used made each of the rows scrollable individually when this was done. To overcome this and to make it possible to scroll through the remaining content and to see each column displayed correctly when doing so, JavaScript was used to bind the scroll of each row together. Since scrolling
was not added to the row headers, these were fixed on the screen while scrolling through the product information.

Figure 15: The new prototype developed with computer information used for user tests.

Figure 16: The new prototype developed with computer information used for user tests, rows removed.

5.2.1 Usability Aspects Considered

A lot of the feedback received during the discussion with usability experts, see section 6.1, was taken into consideration during the development of this new
prototype.

As a result from the feedback received for prototype 1, see section 6.1.1, the developed solution had every second row marked in light gray to make it easier for a user to separate rows from each other. Since the results from this discussion mentioned that showing specific columns next to each other could be preferable, it was also made possible to move columns left or right manually for the user in the developed solution. This meant that two products could be moved to be placed next to each other to make the process of comparing them easier.

From the results regarding prototype 2, see section 6.1.2, it was concluded that a fixed column or row with headers could be preferable when showing large tables on small screens. Due to this, the leftmost column in the new solution, including the headers Product, Price, Screen size, Weight, Review, Ranking, Memory, Storage, Processor model, Full name, In stock, Stores and Comments, was fixed while scrolling through the products. This was done in order to see what the information in each row represented regardless of where in the table the user is located.

In prototype 2, a good feature was that it was possible for a user to customize the table by adding or removing columns. In the developed solution this functionality was provided but since the table was inverted when shown on small screens, rows can be added or removed instead of columns. The information that was possible to add or remove were the fixed headers described above. The only attributes that can not be removed are the product’s name and picture, which together are shown in the table as a column header. This was done to avoid a situation where a user removes a product name by mistake, and then finds it impossible to know which product every cell belongs to. In the existing solution discussed with usability experts a menu was present in which a user could make choices about what columns to see in the table. To minimize the number of clicks needed, and therefore the required user interaction, the menu was removed in the new prototype, see figure 15, and replaced by small crosses on each row in the table, showing a user that the row can be collapsed and removed directly in the table. The removed information can then be added again by clicking on the row’s name, present in a list appearing above the table, see figure 16. By default, all of the information is shown to minimize the risk of users missing important information.

The general feedback received regarding both of the existing prototypes, described in section 6.1.3, described that a big problem with showing large tables on small screens is that column headers can sometimes be so long that very few columns, and therefore little content, is displayed simultaneously on the screen. To try and avoid this, the layout flexbox was used in the new prototype. In this layout each column was set to have the CSS-attribute flex: 1 which resulted in each column taking up an equal amount of space in the containing table. Each column was also set to have a minimum size to avoid them from being too small to read. In the rendered solution this resulted in approximately three different columns plus the header column being visible on a small screen simultaneously.

The general feedback also suggested that it is important for developers to think about whether a large number of rows or a large number of columns is important to see simultaneously. Since more rows than columns will be visible on a smartphone screen in normal portrait mode, it can be beneficial to place data that is to be compared in columns instead of rows. By comparing two columns instead of two rows, more cells can be compared simultaneously.
When product data is shown in desktop mode, each product is represented by a row and different rows can be compared to each other. A normal scenario might be that a user is interested in a few different products and wants to compare as many features as possible between them. Due to this, the developed prototype showed each product as a column instead of a row, making it possible to compare more features for a few products simultaneously than comparing few attributes for a larger number of products.

As a result of the feedback received during the discussion with usability experts, the developed solution became a combination of the two existing prototypes regarding some aspects, and new features were added to hopefully improve the usability further.
6 Results

During the four phases of the project different results have been received. The literature study resulted in a lot of relevant information being found, read more about this in section 2 and 3. How the prototypes were developed can be found in section 5.

Results regarding how the prototypes have been received by usability experts and users can be found in this section.

6.1 Discussion with Usability Experts at Valtech

Two different existing solutions were chosen for the discussion with usability experts at Valtech, the two solutions can be found in section 5.1.1 and 5.1.2. These two solutions were considered most relevant in comparison with the prototype that later was to be developed during the project. The solutions do not necessarily remove an data from tables and they try to solve some of the problems associated with showing large tables on small screens. The two solutions were discussed separately and then compared to each other in order to find positive and negative aspects regarding their usability.

6.1.1 Prototype 1: Fixed Leftmost Column

One of the positive aspects regarding usability found for prototype 1 was that every second row in the table was marked with white and gray respectively, making it easier for users to see which row is which when scrolling left and right through the table’s columns.

Normally, the scroll direction for content on a website is up and down and since this behavior has been learned by many, users are rarely required to think about how to scroll in order to see more content on a website. In scenarios where this prototype is used the scroll direction for the table content becomes left and right, while the scroll direction for the rest of the website is likely to remain up and down. Due to this changed scroll direction, an indication could be required in order to show users that more table content is present outside of the screen’s visible area.

A downside mentioned with this solution is that users are not able to display specific columns next to each other. If a user wants to, for example, compare the content in two columns placed far from each other, he or she will be forced to scroll left and right through the table and remember one of the column’s values while looking at the other’s.

This was the prototype chosen as most appropriate to use together with the new prototype during user tests performed later on in the project.

6.1.2 Prototype 2: User Decides

A drawback for this prototype compared to prototype 1 with a fixed leftmost column was that no headers, neither row or column, were fixed. In prototype 1 the row headers, the values in the leftmost column, are fixed, making it easier for the user to see what each row is representing when scrolling through the remaining table content.

A possible usability problem that was mentioned during this discussion was how to show the user in the easiest manner possible that he or she can choose
which columns to see. In this prototype, the list displaying which columns a
user can choose to see is hidden by default and the list appears when a button
is clicked. Minimizing the number of clicks a user has to go through to reach a
goal is a good method to help users perform tasks with more ease. Showing the
user that it is possible to choose columns and to make it possible for him or her
to choose these columns without having to reveal a specific menu with a click,
can be beneficial to increase the probability that users will use this service.

A few problems follow from letting users decide what columns should be
displayed. The main question will be if users actually know which columns they
want to see and whether or not they know which information is important. This
also leads to the problem of what users will do if they do not know this. Showing
some content by default is one solution to make sure uncertain users will see the
information considered most important from the developers’ point of view.

Even if this solution offers a user a good chance to choose the columns he or
she wants to see, it is important to think about making this action as easy as
possible. If choosing columns require a lot of energy and effort from the user it
is likely that he or she will avoid doing it, even if the result would be beneficial.
Displaying checkboxes straight up instead of in a menu hidden behind a button
is one solution to achieve less required user interaction. The danger with this
is that the area with checkboxes will take up a lot of space and push the table
and its content further down. Being able to see the table and its content while
choosing columns can be beneficial to make it easier for the user. Having to
scroll in order to see the table after column choices are made is another example
of user interaction that preferably should be minimized.

One thing to think about is whether the majority of users will choose columns
in a customized manner or if they will benefit from having a certain number
of columns displayed by default, and if so, which columns. If the majority of
users are likely to be interested in the same information it might be beneficial
to show these columns by default, but also provide more advanced users with
an option to customize the table more. This might help with minimizing the
user interaction needed by the majority of users while maintaining the option to
choose columns freely for those who want.

Another solution for this could be to provide an option for users to collapse
columns directly in the table instead of having a certain area on the website that
shows which columns are displayed and which are possible to choose from. A
problem that will require a solution in this scenario is how to show the user that
he or she can add columns that are collapsed in the table. Making it possible to
collapse columns in the table and also adding the collapsed columns to a menu
or specific area that shows which columns are hidden could be one alternative to
minimize the number of clicks and scrolling needed from the user outside of the
table area.

6.1.3 Both Prototypes

One potential usability problems that came up which is important to think about
generally when implementing a solution for showing large tables on small screens
is the length of column headers. In the examples used for the discussion the
headers were very short, but in many cases these headers are a lot longer and
more descriptive and will therefore require more space. If the headers are too
long the danger is that very few columns, and therefore little content, will be
Another possible usability problem that was discussed was what will happen if many rows are present in the table. Certain scenarios might require several rows to be visible at the same time. A possible solution discussed was that if all rows can not be visible at the same time the column headers should be fixed and always visible on the screen when scrolling down in the table. By doing this, information in table cells will be easier to understand.

From a usability perspective a possible refinement was suggested for both of the prototypes. This enhancement was that developers should think carefully about whether the most important part is to see many rows or many columns from the original table. If tables are displayed on a smartphone screen in portrait mode, more rows than columns will be visible. A thought could be to invert rows and columns in cases where the number of columns a user must see is larger than the number of rows. An example discussed was showing fund information over time to a user. Often in a scenario like this the important thing is to show data over time for each fund, requiring a time span that is long enough to be visible on the screen. If each fund is displayed in a separate row with different dates in each column the result will be that only a short time span will be visible on a small screen. Instead of forcing the user to flip his or her phone to landscape mode to see a longer time span the table could be inverted to make use of the phone’s height in portrait mode. A fund would then be represented by a column and more data can be displayed on the screen simultaneously.

An option that was discussed during this meeting was to remove the table structure and instead replace it with chunks of data to increase the readability of the content, see example of how information can be displayed in chunks in figure 17(Google Play ASOS). To achieve this an option is to place important data that must be comparable between different columns in the same location within all chunks, preferably on a straight vertical line between the chunks to make comparisons possible. The remaining data can be placed as flowing text underneath a headline, this headline can be chosen from one of the columns that best represent all of the data, for example a heading or title. This solution will require a lot of restructuring of data and will therefore need careful consideration to make the table content easy to read. The intention with this suggestion was to make it easier to read all content originally present in one row simultaneously on a small screen. Comparing two different rows, in this case transformed to chunks, might be harder if this solution is used. Placing content that should be comparable in the manner described above might increase the possibility to compare different entries to each other.

During the discussion a potentially good solution was suggested as a combination of the two existing solutions. If a combination of the two solutions was implemented with the added option to invert rows and columns and taking the possible usability problems discussed into consideration, a prototype with a high usability might be possible to develop.
6.2 User Tests

During the twelve user tests, each individual result was summarized into three main points to remember from each respective test. Below, these summarized points can be seen and the full results from the user tests can been found in appendix A. In table 1 a summary of the opinions expressed by many of the test persons can be seen.

The results from the first user test were:

- The user wants to be able to remove excess information and reorganize content according to his or her preferences.
- The user does not understand what the arrows indicating that columns can be moved do to begin with.
- The user wants more information about what each attribute represents.

The results from the second user test were:

- The user thinks that both of the solutions are better than solutions previously seen, and likes that it is possible to customize information in the new solution.
- The user thinks that the direction of data depends on what he or she wants to see. In cases where the user wants to compare several products he or she thinks it is better to have products placed in rows. When he or she has narrowed it down to a few products, it is preferable to place the products in columns and compare more attributes among the remaining products.
• The user would like a better overview of the data that is present in the
table and would like to be able to sort products based on attributes

The results from the third user test were:
• The user likes the new solution better when comparing a few products.
• The user wants to be able to customize the table in many ways, removing
attributes was considered as a good function but sorting products based on
attributes and removing columns was wished for as well. The user thinks
that the arrows indicating that columns can be moved should be made
clearer so that the functionality is apparent.
• The user wants to be able to filter the products more to have fewer products
to compare in the table. This might be done in an overview shown before
the table solution where each product is shown. Wants to have as little
user interaction as possible, not like in the existing solution where a lot of
scrolling is needed.

The results from the fourth user test were:
• The user likes to be able to add and remove information.
• The user wants to see the information for each product in rows rather than
in columns.
• The user thinks that the prototypes are best fit for different scenarios.
  If the selection is narrowed down the existing solution is considered best
  fit. In cases where many products are compared the new prototype is
  considered best fit.

The results from the fifth user test were:
• The user wants to read information horizontally instead of vertically.
• The user wants to be able to have more information in the table. He or
  she wants to be able to add and remove information in the table.
• The user wants to be able to sort products based on attributes, for example
  price, as an addition to moving columns manually.

The results from the sixth user test were:
• The user likes the new prototype better because it is easier to see and
  read the information in the table and thinks that it is easier to compare
  products in this prototype.
• The user likes that it is possible to remove attributes that are not considered
  important in the new prototype, but wants the functionality to move
  columns to be clarified.
• The user would like to be able to sort products based on attributes, for
  example price or name.
The results from the seventh user test were:

- The user thinks that showing products in columns is better when comparing a small number of products, for example two or three products. He or she prefers that products are shown in rows if more than three products are to be compared.

- The user thinks that crosses and arrows, and their function, should be clarified. Likes that it is possible to customize the new table.

- The user thinks that both of the table solutions are easy to understand, and that they are both preferred over table solution previously seen on mobile devices.

The results from the eight user test were:

- The user thinks that pictures are good to have.

- The user likes that products are shown in columns instead of rows. Likes that it is possible to customize the table but would prefer it if drag and drop was used instead of arrows.

- The user thinks that a lot of technical specifications makes the table cluttered, and would prefer if fewer attributes were shown. It might be better if the user could add extra information if wanted. Would like to be able to sort products based on attributes.

The results from the ninth user test were:

- The user thinks that a picture of each product is good to have, it makes it easier to retrieve information about a product. He or she also thinks that design is considered important for every product.

- The user likes that it is possible to place products next to each other to make the comparison easier in the new prototype.

- The user wants crosses and arrows to be made clearer so that their function is more apparent.

The results from the tenth user test were:

- The user wants to see products placed in rows.

- The user wants to be able to both add and remove attributes, and thinks that it is good that a user can customize a table to his or her liking. Wants to be able to sort products based on attributes.

- The user wants to be able to remove products directly in the table as well. By doing this, a user can remove products that are not of interest and can easily get a smaller selection of products to choose from, which will make the decision easier to make.
The results from the eleventh user test were:

- The user thinks that the new prototype provides a better overview of every product.
- The user thinks that pictures are not needed but would like it if products could be sorted based on attributes.
- The user does not understand what the crosses and arrows to remove and move content are for.

The results from the twelfth user test were:

- The user likes the functionality provided by the arrows, but would like the fact that columns can be moved clarified.
- The user wants to be able to sort products based on attributes.
- The user likes that it is possible to customize the table in the new solution, wants to be able to do this in a solution viewed on a smartphone with a small screen.
Wants to be able to customize the table content by, for example, adding or removing attributes or moving products next to each other.  
Prefers the new prototype.  
Wants to be able to sort products based on attributes, for example price.  
Arrows, crosses and their functionality must be clarified, should be apparent.  
Thinks that pictures of products are good to have in the table.  
Likes that every second row is light gray because it makes it easier to see which row is which.  
Thinks that the best direction of data depends on what stage of a purchase you are at or what you are interested in. Products placed in rows are preferable when many products are to be seen while products placed in columns are better when comparing a few products more thoroughly.  
Wants more information about what some attributes actually mean.  
Prefers the existing prototype.  
Both prototypes are better than previously seen solutions.  
A better overview of all the information present in the table is needed.

<table>
<thead>
<tr>
<th>Opinion expressed by users</th>
<th>Number of test users expressing opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wants to be able to customize the table content by, for example, adding or removing attributes or moving products next to each other.</td>
<td>11(^1)</td>
</tr>
<tr>
<td>Prefers the new prototype.</td>
<td>9(^2)</td>
</tr>
<tr>
<td>Wants to be able to sort products based on attributes, for example price.</td>
<td>8</td>
</tr>
<tr>
<td>Arrows, crosses and their functionality must be clarified, should be apparent.</td>
<td>7</td>
</tr>
<tr>
<td>Thinks that pictures of products are good to have in the table.</td>
<td>7</td>
</tr>
<tr>
<td>Likes that every second row is light gray because it makes it easier to see which row is which.</td>
<td>5</td>
</tr>
<tr>
<td>Thinks that the best direction of data depends on what stage of a purchase you are at or what you are interested in. Products placed in rows are preferable when many products are to be seen while products placed in columns are better when comparing a few products more thoroughly.</td>
<td>4 (+ 3(^3))</td>
</tr>
<tr>
<td>Wants more information about what some attributes actually mean.</td>
<td>3</td>
</tr>
<tr>
<td>Prefers the existing prototype.</td>
<td>3(^4)</td>
</tr>
<tr>
<td>Both prototypes are better than previously seen solutions.</td>
<td>2</td>
</tr>
<tr>
<td>A better overview of all the information present in the table is needed.</td>
<td>2</td>
</tr>
</tbody>
</table>

\(^1\)Out of these eleven, three users wanted to be able to remove products as well as attributes directly in the table.  
\(^2\)Seven of these nine were women.  
\(^3\)Three users also expressed that they might not prefer different table structures depending on what information is shown, but they wanted a list or similar with the products shown first. In this list a better overview of products that exist could be seen and the products of interest could be chosen and compared more thoroughly in the new prototype.  
\(^4\)Two of these three were men.
7 Discussion

In order to make a table easy to use on a small screen, the table content must be restructured and rearranged to make it possible to read as much data as possible simultaneously. To find a single solution that works for showing all types of large tables on small screens is extremely hard and was never the intention with this project. Different tables will have different information that is important to display properly on a small screen, what the data is for is of great importance when deciding how to display a table to a user. If the data in different columns is to be compared to each other one solution might be good, while scenarios where one column’s or row’s data is to be displayed all at once requires another solution to make the content readable. In order to achieve the highest usability possible, the information in a table and how it is to be used must be taken into consideration when deciding how the table should be altered.

7.1 Method

Tests were performed during this project to get qualitative feedback regarding the usability of the existing and custom prototypes developed. From the beginning the optimal scenario included two rounds of usability tests, one for the existing solutions and one for the custom prototype. Due to the limited amount of time available it was concluded that two test rounds would be too time consuming and would take up time that could be used to optimize the custom made prototype. It was decided that the most important solution to try with real users was the custom made prototype. This solution was new and not yet evaluated or tried by real users, it was concluded that testing this would return more relevant test results for the project. Since the goal with the project was to see if a prototype that has a higher usability than existing solutions do could be implemented, it still seemed relevant to test the solution compared to existing techniques as well. Due to this, the existing solutions and the custom made prototype were tested in parallel to see which of the solutions the test persons found had a higher usability. To not color the results of the tests, the test persons did not know which of the solutions they were trying during the tests. A danger during the tests was that users could learn and get used to maneuvering through tables after testing the first prototype. If this happened, the users might always think that the second prototype would have a higher usability. Alternatively, users could prefer the first solution they tried since more energy might be put into reaching a goal during attempt one, making the second attempt feel repetitive and of little interest. To make the order in which the prototypes were tested have as little effect as possible on the test results, half of the test persons started with the existing solution and the other half with the custom made prototype.

Qualitative feedback regarding the existing solutions was still needed as a first step during the development of the new prototype. To implement a prototype with a, hopefully, higher usability, the existing solutions had to be tested in some way before the new prototype was developed. A discussion with usability experts was decided to be used as a compliment to the usability tests that were going to be performed later on in the project. As mentioned in section 4.2.1, this discussion was not considered as a replacement for usability tests but was done to show usability flaws and positive aspects with the existing solutions that could be taken into consideration during further development. One of the existing
solutions was then tested on real users in parallel with the new prototype to evaluate the difference in usability between them.

7.1.1 Discussion with Usability Experts at Valtech

During the discussion with usability experts at Valtech, one of the most important conclusions drawn was how important the information in a table and how it is to be used is when redesigning a table to fit on a small screen, more on this in section 7.2. Some tables will require comparisons between different columns and in these scenarios, comparisons must be easy to do even when the table is shown on a small screen. To begin with, the data must be visible simultaneously in order to be compared. If the table structure is modified or in some cases removed, the data must be rearranged in a manner that still makes column or row data comparable without being placed in a pure table structure. Content can for example be placed in a vertical line between different chunks of data, read more about this in section 6.1.3 and see an example of what chunks might look like in figure 17.

Removing the table structure can in some cases increase the readability of the table content even if the normal information flow is altered. Grasping the meaning of table content might be easier if the data is displayed in a customized way on a small screen, compared to showing a shrunken table. An example of data being displayed in chunks can be seen in figure 17.

7.1.2 Usability Tests

In order to get meaningful results from the usability tests, realistic scenarios for the targeted users were important. The targeted users were people who are very used to using smartphones. People who often use smartphones are likely to perform certain tasks on their mobile devices, checking their bank account, monitoring social media and reading newspapers are examples of tasks people often use their smartphone for.

Since the purpose of the project is to try and find a solution for how to show tables on small screens with a high usability, an appropriate task seemed to be to find a scenario that people are used to performing on desktop computers and applying this to mobile devices. Combining experience with mobile usage and familiar desktop scenarios was believed to decrease the effect the tasks would have on the test results, and show results more closely related to usability rather than unfamiliarity with the test scenarios.

Different test scenarios and data types were considered for the project, read more about the different data types considered that can be used to create realistic test scenarios in section 7.2.1. Comparing products was a scenario that many of the targeted users were believed to be familiar with, this combined with product comparisons being appropriate table content resulted in the chosen data and test scenarios.

An important factor to think about when evaluating the user tests is that the table structure and how information is presented in the table is what is being tested. When users are imagining the scenario used for the tests they are asked to act in the manner they would have if they were to purchase a product. In real scenarios like this, the tables used will include more information than what the tables used during the tests did. For example, in real scenarios Stores
and Comments can normally be clicked on to read more information about each product. Stores will normally show which stores the product is sold in, and comments will show other users’ opinions about the product. In the test scenarios these attributes are not possible to click on, and users might therefore not act entirely as they normally would. If one of these attributes normally would be used as a main reason for buying a product, the test persons can not see all of the information they want to make an informed decision. In the tests the users do not have to buy a specific product but instead they are asked to reason and think aloud about how they would act if they were to purchase an item for real. Due to this, it was not considered a problem that certain information was not possible to view since users might still say that they would use, for example, comments as a ground for buying a product during the test. It was not necessary for users to read every comment since the main point of the user tests was not to evaluate if users could perform a purchase, but rather to see how well the information in the table was displayed and how easy it was for users to read and understand said information.

7.2 Table Content

The data that is represented in a table will have great effect on what a modified table representation can look like on small screens. As a result, different solutions and variations will be required for different data types to achieve a table representation that has a usability that is as high as possible. Some cases, for example stock information or mobile subscription information, will require table entries to be comparable to each other to allow a user to make an informed decision or to see differences that might be of interest. Other scenarios, such as fund information over time, bank account information or statistical data, might make it necessary to have a lot of history or different alternatives that should be visible simultaneously.

The data types described above are often complex and very informative, resulting in summarized versions presented in tables to make the content easier to grasp. To not pass over this purpose, it is important to find solutions for how large tables can be displayed with a high usability on small screens.

The table content chosen for this project was product comparisons. This data was chosen partly because each entry in a table of this kind requires many columns or rows with important data. Depending on who the user is, different fields will be considered most important and the need for users to customize the data exists. For example, in cases where a user wants to buy a computer, or any other type of technical device, a lot of technical and cosmetical specifications will be present which will render a large table. Some users might focus on technical details for the computer, while others might put more value into the price, weight or screen size of the computer. This means that all of the fields are required for users to be able to make a valuable comparison, the developers will not be able to choose which columns or rows that are considered most important.

The table content used was also chosen due to the effectiveness with which realistic scenarios can be created for user tests. It can be assumed that many of the targeted users have compared products before a purchase, and that many of them have used websites when doing so. Due to this, many will be familiar with how the tables and websites work on desktop computers. So even in scenarios where the users have not used the tables on small screens, they are likely to be
aware of how the information can be used. Realistic scenarios that users can relate to, for example “You are to buy a new computer with a 13” screen, which of the ones available would you buy based on the existing information?”, can be created. Since many users are likely to be familiar with how the information present in a table is to be used, their focus can be aimed at assessing how well the information is presented rather than trying to figure out what the task they are performing actually is.

Product comparisons was also chosen since e-commerce is a growing market, and the use of mobile devices when shopping online is increasing (eMarketer, 2015; Tobias Lütke, 2014). Some areas of commerce have already been adapted to work well on mobile devices, and some are in progress of becoming more mobile friendly. Technical devices is a market that is hard to adapt to small screens due to the large number of technical specifications present for each product. If several products are compared to each other, a realistic scenario when a technical device is purchased, many of these technical details will be compared between products. To increase the possibility that technical devices can be purchased from a mobile device, these comparisons between products could benefit from being more mobile friendly.

7.2.1 Other Data Types

Other data types that were considered for this project were statistical data and fund or bank account information over time. Both of these data types are often presented in large tables and require much data to be visible at the same time.

Statistical data is often viewed on desktop screens, a number of reasons exist for why this is. One reason could be that statistical data often is used in work situations, for example by agencies or public institutions. In scenarios like these, it is likely that a very large amount of data must be viewed simultaneously and that it might be a better alternative to look at it on a desktop screen where the work requiring analysis of the statistical data is performed.

Another argument for why statistical data was not chosen was that test scenarios would be harder to create and a smaller crowd of people would be the targeted users.

The reason for why fund or bank account information over time was not chosen was that from this data it is relatively easy to strip away information to create a user friendly mobile version. Even if a lot of information is present in a desktop version, presenting fund or bank account information on a mobile screen can be done by modifying the table. Bank account information can for example be stripped down to only include transaction information such as amount for the transaction, from or to whom or what the transaction is, and when the transaction took place. Similar minimization can be performed for fund information over time. In cases like this it could be a preferable alternative to modify the data and to show it in, for example, chunks instead of using a normal table structure, see example of information being displayed in chunks in figure 17.

Another data type considered was mobile subscription plans. When comparing different plans for subscriptions, many different attributes must be considered. This was not chosen due to the fact that the solution that would be needed could be the same as the one provided, different products are being compared but instead of technical devices the product is mobile subscription plans. If the
solutions needed are believed to be the same, the thing that will differ is the data in the table and the test scenarios. Between these two data types, test scenarios where technical devices were to be purchased were believed to be easier for a larger number of users to relate to.

7.3 The New Prototype

Before the new solution was developed, two existing prototypes were discussed with usability experts working at Valtech. The feedback from this discussion can be found in section 6.1. When the new prototype was developed, as much of this feedback as possible was incorporated into the design, see section 5.2.1.

One thing that was not implemented in the new prototype, but that might have increased the usability further, was to fix column headers in a similar manner as row headers. In the developed prototype, row headers stay on the screen when a user scrolls through the products in the table. Column headers, where each header consists of a product name and a picture of the product, could in some cases benefit from being fixed as well. In the scenarios used for user tests each row represents an attribute that can be used to compare products with, for example price or screen size for a computer. In the test scenarios, enough attributes are present to take up more space than what a smartphone screen provides in vertical direction. Due to this, a user has to scroll down to see all rows present in the table and by doing this he or she will no longer be able to see the column headers. Fixing column headers might make it easier for users to remember which of the table cells that belong to which product. This removes the necessity for a user to remember which columns he or she is looking at, and the usability might therefore increase further.

A downside, and one of the reasons for why this was not implemented in the current prototype, is that if both column and row headers are fixed on the screen, very little screen area is left for cells with data. If both column and row headers are short, this solution might work very well since much screen area will be left for cells with data that is to be compared. In the current test scenarios computer information was used and each computer’s product name together with a picture of the product was long enough to take up several lines within the header cell. When the prototype was developed two options existed for how to solve this problem. One alternative was to increase the width of the column and let each product’s name take up only a few lines and to place the picture next to the name. Another alternative was to decrease the column width and allow the names to take up several lines, increasing this specific row’s height. A picture of each product could then be placed underneath the product’s name instead of next to it. By doing this, more columns could fit on the screen simultaneously and more products could therefore be compared without having to scroll. Since most of the remaining attributes did not include as much text as the names together with the picture did, they were not harmed by shrinking the column width. Because of this, the second alternative was chosen. Based on the row height caused by each product’s name and picture together with the relatively few number of attributes present in the table, column headers were not fixed to save screen area and to make it possible to compare a larger number of attributes on the screen simultaneously. If a user is only interested in a few attributes and removes the rest, a possible scenario is that the attributes a user is interested in will fit on the screen.
Since each product’s name was very long, for example *Apple MacBook Air - 1.6GHz DC 2GB 64GB 11.6’’, the product names were shortened to only include the actual name and not the technical specifications in the title, in the mentioned scenario this resulted in *Apple MacBook Air*. An attribute *Full name* was instead added to include the full title for those interested, the technical specifications removed could be found in the table as well.

### 7.3.1 Design Choices

In the developed solution, columns and rows were inverted compared to the desktop version of the table. Data that requires more space and that should be viewed simultaneously can benefit from being shown in columns, taking advantage of the screen area offered by a smartphone in portrait mode. The belief was that in a scenario where a user wants to buy a new computer, he or she is likely to quickly narrow the selection down to a few products depending on, for example price or screen size. When the products that are left are then compared many attributes might be necessary to compare more thoroughly.

When the new prototype was implemented, the CSS-layout flexbox was used for showing tables. The table content was structured using div-elements and styling was then applied to them. During the implementation phase different techniques and and methods were tested to see which alternative was most appropriate. When using flexboxes, an attribute *flex-direction* can be set to indicate in which direction the items within the flexbox should be rendered (CSS Tricks, 2015). For the elements in a table to take up as little of the screen’s space as possible but still be readable, the best solution found was to set each row’s height to that of the element with the largest height within the row. By using the flex-direction *column* for the entire table container, elements are laid out from top to bottom within the container producing appropriate table rows. Each row was then made into a flexbox-container with the flex-direction *row* to place every element in the row from left to right as table cells. With this design a result was that when horizontal scrolling was applied to the rows, every row scrolled individually. This was overcome by binding all of the rows’ scroll together with JavaScript.

This design was chosen because it was considered appropriate for the data being represented. Having products in columns and allowing the user to collapse rows and move columns to be placed next to each other was an appropriate table solution for products. Other data types might require other solutions to have a high usability, read more about this in section 7.2.1.

Displaying product information using chunks, for example, was not considered appropriate. As mentioned in section 7.2.1 chunks might be preferable compared to a normal table structure when showing, for example, bank account information, see example of what chunks might look like in figure 17. Why this solution is considered appropriate for account information is that few elements within each chunk must be rendered, possibly only the transaction date, amount and recipient or source must be shown. Another reason for why this solution is considered appropriate for bank account information is that fewer comparisons between elements are believed to be needed. While product information’s main purpose might be to compare the different products to each other, a user’s interest when it comes to bank account information is more likely to be checking account balance or specific transactions. Each chunk of data can render data
with a high usability efficiently, but comparisons between different chunks will be a lot harder to perform. Each piece of information must be placed in the same location within a chunk and instead of comparing two columns or rows where the data is placed close together, these two locations separated by a lot of other data must be compared. The main purpose with using chunks is to display all of the data for one item simultaneously to the user, and if much data is present for each item a possible scenario is that a chunk will fill an entire smartphone screen. The specific chunk studied will be very easy to interpret, but comparing different chunks might become impossible.

7.3.2 Possible Extensions

Indications on a website that it is possible to scroll up and down are seldomly needed since this is the normal scroll direction that users are familiar with. In certain scenarios or specific contexts indications that it is possible to scroll might be needed, but often users are so used to this behavior that hints might be considered excessive. When the scroll direction alters from this normal behavior, indications might be necessary to alert the user of how and where content is available. Since the scroll direction in the developed solution is left and right, indications for where users can find more content are needed. In the prototype provided the rightmost column that is visible on the screen is, in most cases, slightly cut to show that the content is partly hidden and that more of the content is possible to look at if the table is dragged horizontally.

Some might argue that this is not enough to show users that the table is scrollable. An alternative that could have been implemented is to slightly fade the right edge of the table on the screen, provided that the user has not reached the rightmost column. By doing this, the user can see that content is faded out and that there is likely more to see. Another alternative could have been to always, or only on the first page load, show the scrollbar. A scrollbar is a graphical element that users are familiar with and seeing one together with its direction will show users how the content is laid out and how to see more. Other alternatives to show the user how content is laid out on the page could be to show a minimized version of the table in a corner of the screen, marking where in the table the user is located and at the same time showing where the remaining content is present. In this solution though, this alternative was not considered appropriate since the minimized table would take up space on the screen that the real table could benefit from using.

Many alternatives for how to show a user that a scroll direction going from left to right and vice versa exists, but the most important thing is that developers think about this and show users that scrolling is possible.

In the developed solution a user can move columns to the left or right, making it possible to move specific columns next to each other. An option that is not provided is for users to collapse columns, in a similar manner as rows can be collapsed. One could argue that since the option to relocate columns exists and since closing or moving a column one step requires the same amount of clicks, this functionality is not needed. Scenarios where a user wants to study more than two columns next to each other and where these columns to begin with have several other columns that are not of interest between them might benefit from having the option to close columns though.

To minimize the number of clicks needed and to make it possible for users
to still both close and move columns, the solution that is believed to be the most optimal is to make it possible for the user to collapse columns directly in the table with crosses on each column, and to be able to move columns using drag and drop. Drag and drop is a function that many smartphone users are familiar with, and if each column has an indication showing the user that it is possible to move the column this solution will require only one motion for moving a column instead of clicking once for each step the column is to move in the table. Showing users that the column can be moved using drag and drop can be done by, for example, adding a textured area in a visible corner of each column, an indication often used to show this functionality.

This was not implemented in the current prototype due to a lack of time and due to the fact that in scenarios with tables containing technical data like the tables in this project does, a filtering option is often provided before this table is shown. This filtering option is used to remove alternatives that the user is not interested in and to show the products that have the attributes the user is looking for. If one assumes that a filter like this has been applied before the table is rendered, the resulting table is believed to not include an extremely large number of products that the user has absolutely no interest in. Even so, for the solution to be applicable to a larger number of data types it would be beneficial to add both the option to close columns and also to implement drag and drop for each column.

An extension that could make the developed solution applicable to even more scenarios is to provide and option for the user to invert the table directly on the screen. In the developed prototype the data is inverted compared to the table’s desktop layout since this was considered a more appropriate alternative for the data used. Other scenarios where elements must be compared row-wise instead of column-wise could benefit from not being inverted when shown on a smaller screen. By adding an option for the user to invert the table on the fly, both of these scenarios can be covered by the developed solution instead of only scenarios where the table should be inverted when shown on smaller screens. This would also make it possible for users to change the direction depending on whether the table is shown on a mobile screen in portrait or landscape mode. The developed solution works best provided that the smartphone screen is shown in portrait mode.

One feature that could have been implemented is to provide an option for users to sort products based on attributes, for example price or name. By doing this, users can see products in order based on the attribute they find most important. On websites like www.prisjakt.nu, www.pricerunner.se or similar sites this functionality is often provided and appreciated by many. To make it possible to add this functionality to the developed prototype, a JSON-array was used to store every product together with its attributes. By doing this, functionality can be implemented to retrieve data from the array based on how the user wants information to be displayed.

### 7.4 Results

W3C, The World Wide Web Consortium, defines usability as “The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments”. In this definition W3C state that effectiveness refers to “the accuracy and completeness with which specified users can achieve specified
goals in particular environments”, efficiency refers to “the resources expended in relation to the accuracy and completeness of goals achieved” and satisfaction refers to “the comfort and acceptability of the work system to its users and other people affected by its use” (The World Wide Web Consortium, 2002).

As mentioned in section 4.2.2, qualitative questions were asked to users during the tests, read more about usability testing in section 4.2. Due to the relatively low number of test users qualitative feedback was sought for, in contrast to quantitative. This qualitative feedback was sought for in order to see why users like or dislike a prototype, why they find information easy or hard to retrieve from a prototype and how possible problems can be fixed in order to increase the usability of the prototype. An important thing to remember though is that even if the results show an indication about positive and negative usability aspects for the developed prototype, the thoughts expressed during each test are subjective and all details might not be true from a more objective usability perspective.

Besides these qualitative questions, users were also studied during the tests to see that they understood the prototype they used. This was done in order to see if what the users said matched what he or she did which indicated that they understood what they were doing during usage, for example if the user said “I am removing attributes” it was also studied to see that this was in fact done.

When the developed prototype was evaluated, it was done in relation to the definition of usability mentioned above. The results can be seen as qualitative information about how a group of people perceive the developed prototype’s usability and how they think it might be improved further.

7.4.1 User Test Results

When the new prototype was tested on real users together with an existing solution, the users were unanimous on some points and their opinions differed widely on others. The summarized results from these user tests can be seen in section 6.2 and table 1, all of the notes taken during the twelve user tests can be found in appendix A.

Something that was great to hear was that two of the test users stated that the solutions used in the tests were both better than any other solution they had previously seen. Since the existing prototype used was considered as a good one for some data types by usability experts, it can be concluded that some solutions that are pretty good already exist out there. The new prototype was developed with the hope of increasing the usability when looking at large tables on small screens even further. If both of these solutions are considered better than many others seen, one can hope that the development of solutions to show large tables on small screens is going in the right direction. The existing prototype is already used and surely the new prototype can be used as inspiration to others for how data can be shown. Hopefully these two solutions can be developed further and new solutions can be brought forward to try and fight the problem of showing large tables on small screens.

What almost every user thought was nice, eleven out of twelve, with the new prototype was that it was possible to customize the table to your own liking. The fact that attributes shown in rows could be removed seemed to be appreciated by most of the test persons and that products could be placed next to each other to perform a comparison more easily also seemed to be liked. Three of these eleven users wanted to be able to remove products as well, not just attributes.
As stated by one of the tests users, “removing products could be good in order to narrow the selection of products down and to make the remaining products easier to compare on a small screen”. The possibility to remove products should be implemented in the future to satisfy these users.

What seven of the users also thought was that the indicators showing that columns can be moved and rows removed must be redesigned to make their function more apparent. At the moment some of the users realized what the crosses and arrows were for after a while, the crosses were more easily understood than the arrows, but the functionality should be apparent from the start. To try and address this problem these elements should be styled in order to show users what they are for immediately and to make the prototype easier to use more efficiently. One user suggested that the crosses could be made red since this color is often used to indicate that elements can be removed or closed. A few other users suggested that drag and drop might be a better alternative for moving columns, compared to the arrows in the current solution. It can be concluded though that the functionality was appreciated by the majority of test users, eleven out of twelve, and that it might be good to provide options for users to customize large tables when shown on small screens. What is necessary though is to make sure that the functionality is clearly shown so that users understand what it is for and how to use it.

What eight of the test users also wanted in both of the solutions was an option to sort products based on attributes, for example name or price. As mentioned in section 7.3.2, the new prototype was implemented in a way that makes it possible to implement this in the future. Generally speaking though, this might be good to think about when large tables are shown on small screens. When www.prisjakt.nu, www.pricerunner.se or similar websites are shown on desktop screens this option is often provided. It can be concluded from the test results that users would like to have this functionality on mobile devices as well. This could be done by making each attribute clickable and if an attribute is clicked once, for example price, the products are sorted from lowest to highest price. If the attribute is clicked again, the products will be sorted from highest to lowest price. An arrow pointing upwards or downwards can be used to show users which of the attributes the products are currently sorted by and in which direction.

An interesting thing that also came up during the user tests was that some users thought that products should be placed horizontally, others thought products should be placed vertically and some users thought that the direction depended on what you are looking for or at what stage of a purchase you are. The three users who wanted to see products placed in rows preferred the existing solution, and those who preferred products being displayed in columns liked the new prototype. What some also said was that the existing solution was preferred when many products were to be compared, for example in the beginning phase when you are looking to buy a product and have not narrowed your selection down. Among these users, some said that they would have liked it if you from the existing prototype could choose a few products that then would be displayed in the new prototype for a more thorough comparison. Others thought that the new prototype was appropriate if you, in some other way, had narrowed your selection down to a few products that you wanted to compare more thoroughly. These users wanted to have a better overview of products first, not necessarily in the form of the existing solution, but in a list or similar. From this list they
wanted to choose products that could be compared more thoroughly in the new prototype. As a conclusion, it can be said that the new prototype was generally well received and appreciated when it came to comparing a few different products more thoroughly, but that other solutions might be necessary to find in order to provide users with a better overview of data before this comparison can be made.

An idea that occurs from the above mentioned phenomena is that product data might benefit from being shown differently during different stages of a purchase. More studying and tests are needed before anything conclusive can be said, but it is possible that during different stages of a purchase users would like information to be presented in different ways. If this is true, e-commerce could benefit from redesigning product websites and profits might even increase if these sites have different layouts depending on what stage of a purchase a user is located at.

As stated above, four users (alternatively seven, see table 1 footnote 3) thought that a different layout was preferable before a thorough comparison is to be made. Aside from these users, two others seek a better overview before the comparisons are made, mainly to see which products exist and how many products there are. Providing a good overview is one of the hardest problems to solve when it comes to showing large tables on small screens. If the entire table is shown in its desktop layout, all of the data is displayed and the user gets an idea of how much data there is. The downside of this is that the usability of the table is very poor, even if all of the information is shown the data will not be readable without zooming in a majority of cases. A solution to this, specifically appropriate when products are compared but that could be appropriate in other scenarios as well, is to provide the user with only a part of the table data to begin with and then provide the user with all of the data when only a few elements are chosen. An option could be to let the user decide one or two attributes that are considered most important, for example price and screen size. When the user has made this choice, he or she will see a list of all the products, placed in rows, with only these two attributes shown. The user can sort products based on these two attributes and choose which products to compare more thoroughly. The user might click on specific products to compare, or choose to compare all products with a specific screen size under a certain price. By doing this users will get to see an overview of all the products that exist to begin with. After this, users will be able to choose the attributes considered most important for the purchase, choose the products that are most interesting and compare these more thoroughly.

This could also be good since some of the test users stated that they wanted better filtering options in the table. Many wanted to exclude products with a price over a certain limit or certain brand. If the above explained solution was implemented, this could be achieved. If other solutions were used to provide users with a better overview, it is still considered to be a good idea to let users filter the data based on attributes they consider important. This is often possible to do on websites were products are compared, and it should therefore also be possible to do when looking at a website on a mobile device.

Another interesting thing found in the results is that among the three users who preferred seeing products placed in rows, two of them were male. Vice versa, a majority of the users who preferred the new prototype, seven out of nine, were female. Due to the relatively small sample of test users no definite conclusions
can be drawn from this but it would be interesting to test this theory further. If men prefer to compare information row wise and women column wise, this can be used in e-commerce and advertising. Products that are targeting women could benefit from being shown column wise if women think that comparisons are easier to make with this layout and products for men could be shown in rows if this is preferred.

What seven users also said was that pictures of the products sold were greatly appreciated. Many were more willing to buy a product if they could see what it looked like and not just see the technical specifications. This is something that is worth thinking about when selling products in general.

7.4.2 Taking the Results a Step Further

As mentioned in section 7.4, it is important to remember that the opinions expressed during the user tests are to be considered subjective from the users who tried the prototypes. Even so, the opinions can be considered as valuable qualitative feedback about the prototypes’ usability and can help identify usability problems and show an indication about how these problems can be fixed.

A question that should be asked is whether or not a usability that is as high as it is on a desktop computer can be achieved on a small screen with the data types used during this project, or if the small screen comes with a cost. One possible cost could be that the least important data is removed in order to increase the readability. Another possible cost is that all data will be present but that a poor overview of it will be available, that is if the data is to be readable. Further possible costs are that all three variables in W3C’s definition of usability might not be fulfilled. An example of where only two of these variables were fulfilled during the user tests was when users tried to customize the table in order to compare products that are of interest more easily, users effectively achieved the goals and with a great satisfaction, but the efficiency with which they could do it was poor. A better solution for moving columns might increase the usability while performing this specific action, increasing the efficiency with which it can be performed, but the overall usability of the solution still lacks a good overview of all the data which means that many of the users miss seeing a large number of products with a high satisfaction.

This theory about cost being necessary is also strengthened by the existing solutions described in section 3.3. Each of the three solutions discussed either require user interaction to see data of interest on the screen simultaneously, or offer different modes with different features for the user depending on what he or she is trying to achieve. The thing that these three solutions have in common is that they require user interaction, reinforcing that a cost might be necessary to pay in order to provide a high usability on a small screen. If this theory is true, a start towards finding a way to show large tables on small screens with a high usability is to identify which cost is least ruinous for different data types.

In this project, the cost was that the overview of the data suffered in some scenarios. Users identified that the new prototype was easy to use when a small number of products were to be compared more thoroughly, but also said that it was hard to use if a large number of products were to be compared since an

---

5The variables mentioned are the ones with which a specified goal in a particular environment is to be achieved, effectiveness, efficiency and satisfaction. Read more about this in section 7.4.
overview was missing. The existing prototype on the other hand provided a better overview of the number of products that existed, displaying them in rows which could be perceived as a list like structure of the products. The majority of the users thought that this made the comparison between different products harder to perform in this solution though, indicating that this could be the cost for the existing solution. Since two different goals exist, seeing an overview of a large amount of products and comparing a small number of products more thoroughly, two different solutions might be necessary to achieve a high usability throughout an entire product comparison process. Providing a solution with a list like structure, described in section 7.4.1, where a user is required to make active choices leading them up to a state where a more effective comparison can be made might be the solution with the highest usability for product comparisons. The upside to this solution is that both of the negative usability aspects stated for each individual solution can be overcome by combining them into two steps instead. The cost will be that more user interaction is required, but with the possible upside of a high usability in the two steps.

To begin with, many users want to compare a large number of products on a high level. The answers showed that many might prefer a list with less information to get a good overview. In this first step the cost of providing a good overview might be that information is lost in the process. Later, when comparing a few products more thoroughly, many preferred removing products that are not of interest from the original table and seeing only interesting products in a column layout in order to see more attributes on the screen simultaneously. On a desktop screen all of these steps are possible to fulfill in the same table since all of the information fit on the screen. A price that might be necessary to pay to fulfill all of the three variables in the usability definition is to require more user interaction than on desktop computers. If a minimal amount of user interaction is sought for and all data must be presented at once, the cost might instead be that only one or two of the three variables needed to achieve a task with high usability can be implemented.

A majority, nine out of twelve, of the test users found the new prototype easier to use and thought that it was easier to get the information needed from it compared to the already existing solution. The goal with the project, finding a solution that has a higher usability than existing solutions, can in some sense be considered achieved. Even so, many aspects still remain before the goal can be said to be completely reached. The developed solution is to be considered a solution on a conceptual level, more styling and tweaks will be needed depending on the data displayed and the scenario in which the table is used. What can also be said is that finding a global solution for how large tables are to be shown on small screens is not possible. Different data types will require different solutions and different scenarios call for different alterations of data in order to achieve a high usability. Generally speaking, it can be said that some data types allow developers to remove data from the table when the table is shown on a small screen. These data types are generally easier to develop solutions for due to the decreased amount of table content. Other data types can benefit from being shown with the normal table structure removed. These data types require a lot of innovative thinking from designers, usability experts and developers in order to not lose valuable information and functionality provided by the original table, while at the same time increasing the usability on mobile devices. The last category, table data that can not be stripped of information and that is hard
to show in a new way without losing valuable information, are the ones that have been investigated in this project. In some cases these tables are believed to benefit from being shown in the manner provided by the prototype developed during this project, especially the data types and scenarios used during this project.

More research is needed before any definite conclusions can be drawn about other data types and scenarios and whether or not they can benefit from the solution provided in this project. To get some answers about this, the prototype should be improved in the ways described in this section and possibly in the ways described in section 7.3.2 as well. After these improvements are implemented, the prototype should be retested to see if users find the new and improved prototype and its functions even better. In order to determine if the prototype has a higher usability than all other solutions out there, the prototype should be tested against other solutions as well. In the test scenarios during this project, the solution used was chosen since it was considered as the most appropriate existing alternative for the data type used. In order to see if the prototype has a higher usability in other scenarios as well, other data types should also be tested.

### 7.5 Possible Improvements

If the project was redone now, or if more time was available, a few changes would be made. One of the main things that would have been done differently is how and when the user tests were carried out. During the project only one round of user tests was carried out due to the limited amount of time. These user tests provided very good and valuable feedback, and after the tests were carried out some changes would have been good to implement for the new prototype. If tests could have been carried out throughout the project during several sprint like rounds, improvements could have been implemented and tested again. If this would have been done, an even better prototype could hopefully have been implemented.

It might also have been preferable if more people could have tested the prototypes. If several test rounds would have been carried out fewer people could have been tested during the first rounds to get qualitative feedback to improve the prototype. During a few test rounds at the end of the project, when the prototype approaches a ready state, more people could have been used for testing to get more feedback about the prototype after improvements had been implemented.
8 Conclusion

The goal with this project has been to produce a prototype that should help overcome some of the issues associated with showing large tables on small screens. During the project, a prototype has been developed for this purpose where features suggested as missed by both possible users and usability experts are implemented. This new prototype has then been tested in parallel with an existing solution to see which of the solutions is more appreciated by users.

The responses given by test users are to be considered subjective qualitative feedback to help develop the solution further. The majority of users thought that the new prototype had a higher usability, but many mentioned that they preferred the prototype in specific scenarios. Product comparisons were used for the tests and users hinted that having a better overview of information during the phase of a purchase were you are very uncertain would be good. The developed prototype was preferred during a stage where you have chosen a few products and want to compare their technical details more thoroughly.

If the results are to be interpreted on a higher level, a theory that emerges from the results is that a high usability on small screens might come with a cost. Fulfilling all of the variables present in W3C’s definition of usability, see section 7.4, might only be possible at the cost of losing something else. Whether the loss is an overview of information or detailed comparisons will depend on the data being presented and the scenario where it is shown. What can be said about the prototype developed in this project is that an overview is what was mostly sought for. If instead a solution like the one suggested in section 7.4.1 would be implemented, the belief is that a better overview can be provided and the usability might increase further for a majority of users. If this solution is implemented the cost might instead be that more user interaction is needed, but as stated by some users, this might be a better alternative to get both an overview of information as well as a good detailed comparison.

A possible future solution should be implemented and tested to strengthen the theories put forward during this project. More solutions should be looked for, for other data types and scenarios, and tested to see if users find them better than existing solutions. Hopefully this report and this project is an eyeopener for many to see that this problem might have solution that is yet to be found.
9 Future Work

While the developed prototype is considered appropriate for products or similar items that are to be compared to each other, solutions for how large tables with other data types can be displayed on small screens should also be investigated in the future. Some of the data types mentioned in section 7.2 and 7.2.1, for example statistical data, bank account or fund information, could also benefit from being displayed in a way that is more user friendly. Other data types could benefit from being displayed in other ways than the data used during this project. Some data types could benefit from being displayed without a normal table structure, read more about this in section 7.2.1, while others might need more table like solutions to convey information easily to the user. In both of the scenarios, good solutions must be sought for in order to provide users with an easy way to retrieve information from large tables on small screens.

Besides needing better solutions for how other data types can be displayed to people in a user friendly manner, more functionality can be added to the developed solution. Many ideas for how the developed solution can be improved and developed further can be found in section 7.3.2.

The theories brought forward in section 7.4.2 should also be implemented to see if mobile screens actually come with a cost when it comes to usability. If the suggested solution works better than the prototype developed during this project and existing solutions, a possibility is that usability regarding product comparisons on a small screen might benefit from requiring more user interaction.
10 References


W. Xu, X. Yang, and Y. Shi, A New Mode of Browsing Web Tables on Small Screens.


A User Test Results

Before the user tests were carried out, each test person was asked to talk about how he or she would act if forced to buy a new laptop. Many of the test persons had used sites where products can be compared for this, but for those who had not they were also asked to talk about how they would use such a site.

A.0.1 Test Person 1

Below the results from user test 1 can be found.

General Thoughts Before the Test: Test person 1 mentioned that if forced to buy a new computer he or she would:

- Go to a physical store and ask someone working in the store with good knowledge about computers or ask his or her brother who is knowledgeable about computers.
- Prioritize a computer’s performance in relation to its price.
- A site to compare products has been used before, but not specifically for computers.

New Prototype, Comparing Laptops: When trying to find a product to purchase in the table containing laptops the test person:

- Excluded products costing more than 10 000 kr mentally, he or she did not move any columns around to do so.
- Mentally removes some attributes, for example screen size and weight.
- Does not understand the difference between memory and storage.
- Removes some computers with a certain processor model based on hearsay.
- Finds comments from others uninteresting since they are subjective.
- Think that full name is uninteresting.
- Wants to see all the attributes he or she is interested in on the screen simultaneously. Does not realize rows can be removed and columns moved.
- Is asked if it is possible to see the sought for attributes simultaneously on the screen, realizes that rows can be removed and columns moved. The test person removes attributes that are not of interest and moves products of interest next to each other so that they are visible on the screen at the same time. Very satisfied with the result.
- Wants to be able to move a column farthest to the left and not just one step at a time.
- Thinks the overview of information becomes much better when attributes are removed and columns moved. Also thinks that the information is easier to grasp when doing this.
Existing Prototype, Comparing Tablets: When trying to find a product to purchase in the table containing tablets the test person:

- Finds the table solution to be more basic than the first one.
- Thinks it is harder to realize that he or she can scroll from left to right than in the first solution.
- Thinks it is bad that information can not be moved or removed, says that the first solution was better in this way.
- Thinks that other attributes are more important when looking at tablets instead of computers, screen size is mentioned as an example.
- Would choose an Android tablet since he or she is used to an Android smartphone.
- Says that camera and storage are important attributes to look at when buying a tablet.
- Does not like that a lot of scrolling is needed to compare to products to each other. Says that a lot of remembering is needed to compare two or more products.
- Says that the first solution was better.
- Thinks that pictures, present in the first solution, were excessive.
- Wants to be able to remove rows and columns.

Comparing Both Prototypes After the Test:

- Thinks that the second prototype, the existing solution, is easier to use since only scrolling is needed.
- Thinks that the first prototype, the new solution, is easier to retrieve information from. Also thinks that the prototype is more usable since information can be customized and organized more according to the user's needs.
- Would have liked a box with information or similar to explain what the attributes mean, for example memory and storage.

Three Important Things to Remember from the Test:

- The user wants to be able to remove excess information and reorganize content according to his or her preferences.
- The user does not understand what the arrows indicating that columns can be moved do to begin with.
- The user wants more information about what each attribute represents.
A.0.2 Test Person 2

Below the results from user test 2 can be found.

General Thoughts Before the Test: Test person 2 mentioned that if forced to buy a new computer he or she would:

- Start by going to a store with people who can help show him or her computers that can be appropriate.
- After getting information from a store, he or she would look online on, for example, www.prisjakt.nu or www.pricerunner.se.
- After comparing products on a site like the above mentioned, he or she would buy the computer from a store that is familiar.

Existing Prototype, Comparing Tablets: When trying to find a product to purchase in the table containing tablets the test person:

- Does not like that the product name is long enough to require scrolling to see the entire name, would have preferred it if the name was split to be on two lines instead.
- Generally likes the order of the attributes, but would like to rearrange some of them.
- Likes that the product name is fixed on the screen when scrolling.
- Thinks that the full name of the product is uninteresting. Can be removed or put last in the list of attributes.
- Likes that every second line is light gray to separate the lines from each other. Thinks that white and light gray is better than having black lines around each row since this can make the table look cluttered.

New Prototype, Comparing Laptops: When trying to find a product to purchase in the table containing laptops the test person:

- Thinks that review and ranking could be placed lower, and that screen size and memory are important things to see. In stock feels very important.
- Thinks that it is very good that the headers in the leftmost column are fixed when scrolling through the remaining content in the table. Thinks that is often missing when tables are shown on mobile screens.
- Likes that a picture of the product is present.
- Thinks that it is very easy to assimilate the information in the table compared to solutions previously seen.
- Likes that it is possible to remove rows directly in the table.
• After removing information from the table, the user does not realize that the information can be found above the table and added again. After a while the list where it is possible to add information into the table again is found, the user thinks that it must be made clearer that information can be added again.

• Thinks that it is a good solution for users who are good at reading technical specifications and know computers.

• Would like to be able to sort products based on attributes. For example only show computers with an SSD disk.

Comparing Both Prototypes After the Test:

• Thinks that it is easier to compare many products when they are placed in rows in the first prototype, the existing solution.

• Thinks that the second prototype, the new solution, is better when you have narrowed it down to a few products and want to see a lot of information about those products.

• Thinks that the arrows to move products are good, especially when you have narrowed the selection down to two or three products and can place them next to each other. Thought it was hard to understand what the arrows were for in the beginning.

• Would like an overview of the products that exist to see how many there are before the large table is shown. In the overview the user thinks that all the technical details are excessive. Would like to be able to choose information to see from the overview, for example only showing laptops with SSD disks in the table.

• Would like to move rows around, not just columns.

• Thinks that both of the prototypes are better than solutions previously seen.

Three Important Things to Remember from the Test:

• The user thinks that both of the solutions are better than solutions previously seen, and likes that it is possible to customize information in the new solution.

• The user thinks that the direction of data depends on what he or she wants to see. In cases where the user wants to compare several products he or she thinks it is better to have products placed in rows. When he or she has narrowed it down to a few products, it is preferable to place the products in columns and compare more attributes among the remaining products.

• The user would like a better overview of the data that is present in the table and would like to be able to sort products based on attributes.
A.0.3 Test Person 3

Below the results from user test 3 can be found.

**General Thoughts Before the Test:** Test person 3 mentioned that if forced to buy a new computer he or she would:

- Begin by looking online on sites comparing computers to find a good one.
- When the user has found one or a few models that seems interesting, he or she looks at [www.prisjakt.nu](http://www.prisjakt.nu), [www.pricerunner.se](http://www.pricerunner.se) or a similar site to compare the products and find a place to buy the computer. After this, the user wants to go to a physical store and see the computer and how it behaves during user interaction.

**New Prototype, Comparing Laptops:** When trying to find a product to purchase in the table containing laptops the test person:

- Starts by looking at the site to see which alternatives exist. Wants to be able to see more products than two or three products at a time.
- Wants to be able to filter the choices more, for example only choosing specific brands or similar.
- Realizes what the cross on each row does, likes that it is possible to remove rows and that the attributes appear in a list and are possible to add back into the table.
- Would like to have a landscape mode as well.
- Wants the leftmost column with headers to be made smaller to fit more columns on the screen simultaneously. Thinks that it is better to fit more products and fewer attributes on the screen (in cases where shrinking the headers results in them ending up on two rows, and therefore pushing the content down outside the screen).
- Thinks that the table offers a good overview and that it is clear how the table is to be used. Good way to display the data and the table contains relevant information.
- The user wants to start by looking at the hardware in the computer, removes other attributes and all of the information considered relevant is then displayed. Can then add comments or similar to get more information and compare products more thoroughly.
- Likes that scrollbars are visible when scrolling left or right, thinks that it gives the user an idea of how much information is left to see and where among this info the current information is placed. Likes that scrollbars are visible on each row instead of only one for the entire table.
- Wants to be able to read more detailed information, for example read comments.
Wishes that the arrows for moving columns would be clearer, for example by removing the left arrow on the leftmost column and the right arrow on the rightmost column. By doing this, arrows are only present and clickable in the direction columns can move and all buttons do something. Clicking left on the leftmost column without anything happening was considered confusing since no function existed.

Existing Prototype, Comparing Tablets: When trying to find a product to purchase in the table containing tablets the test person:

- Does not like that attributes can not be removed.
- Wants to be able to filter products more thoroughly, for example removing all products from a certain brand.
- Likes the first prototype, the new solution, better.
- Does not like that a lot of scrolling is needed to compare different products.
- Rather wants to see several attributes for a few products than seeing the same attribute for several products, for example seeing the attribute microphone for all products was not considered as important as comparing several attributes for two specific products.
- Wants all of the heading to be visible in the leftmost column.

Comparing Both Prototypes After the Test:

- Thinks that the only positive thing about the second prototype, the existing solution, is that more products are visible on the screen simultaneously. Likes the first prototype, the new solution, better when products are to be compared. Wants a solution where products are visible in a list in an overview first, and where it is possible to choose products from the list that are to be compared. The comparisons can be made in the first, new, table solution.
- Wants to be able to remove products too, not just attributes.
- Prefers the first, new, solution because it is easier to retrieve information from the table. Does not like that the second, existing, prototype requires a lot of scrolling.
- Initially thought that the arrows to move each column showed where a column’s edges are.
- Likes that it is possible to move columns, wants to be able to place columns in the order he or she wants. Sometimes hard to see that columns actually move when only two are shown in the screen simultaneously.
- Thinks that another solution for moving columns is needed, but that drag and drop would not be a good solution. The user thinks that arrows are clearer, one way could be to remove the left arrow from the leftmost column and the right arrow from the rightmost column. Another solution could be to have a heading sort or similar on the row with arrows to clarify what the arrows are for.
Three Important Things to Remember from the Test:

- The user likes the new solution better when comparing a few products.
- The user wants to be able to customize the table in many ways, removing attributes was considered as a good function but sorting products based on attributes and removing columns was wished for as well. The user thinks that the arrows indicating that columns can be moved should be made clearer so that the functionality is apparent.
- The user wants to be able to filter the products more to have fewer products to compare in the table. This might be done in an overview shown before the table solution where each product is shown. Wants to have as little user interaction as possible, not like in the existing solution where a lot of scrolling is needed.
### A.0.4 Test Person 4

Below the results from user test 4 can be found.

**General Thoughts Before the Test:** Test person 4 mentioned that if forced to buy a new computer he or she would:

- Ask his or her father who is knowledgeable about computers about what computer to buy.
- After getting an idea about what computer to buy, the user would go to a physical store often visited for computer purchases in the past and buy the computer there.
- The user has not used a site to compare technical devices before, but has used a similar site to compare other kinds of products.

**Existing Prototype, Comparing Tablets:** When trying to find a product to purchase in the table containing tablets the test person:

- Likes the order of the attributes.
- Likes that it is possible to scroll by touching the entire table. Likes that it is possible to see the same attribute for several products simultaneously, makes it easy to compare.
- Easy to read information in the table.
- Good that every second line is marked with light gray to separate lines from each other.

**New Prototype, Comparing Laptops:** When trying to find a product to purchase in the table containing laptops the test person:

- Thinks it is nice that it is possible to scroll sideways.
- Wants more information in the product’s name, for example the full name even if it makes the heading very long.
- Thinks that it is harder to separate products from each other compared to the first, existing, prototype. Would like more space between each column to separate them from each other more easily. Likes to have the products on the left and attributes in columns, the user generally prefers reading the content horizontally compared to column wise.
- Thinks that it is easier in the first, existing solution, to compare one attribute for several products.
- Liked that the header in the leftmost column had a black line separating it from the remaining table data. Thinks that the cross on each line is in the way.
- Likes the order the attributes are placed in. Review and ranking could be pushed further down.
• Initially thought that the arrows on each column was to browse to see more pictures of each product. The user tried doing it and realized the arrows moved the columns, and really liked that it was possible to move products and place interesting ones next to each other. Wants to be able to see at least three products next to each other on the screen.

• Realizes what the crosses are for, likes the functionality they provide and that it is possible to remove information from the table and then add it back. Really likes that it is possible to customize a table to look like the user wants it to.

Comparing Both Prototypes After the Test:

• Likes that products are presented horizontally like in the first, existing, prototype.

• Likes the second prototype, the new solution, better since it is possible to customize the table.

• Thinks that the first, existing, solution is better if you have narrowed it down to a few products and want to compare them. Likes the second, new, prototype better if you want to compare several products and find a potential product to buy.

• Thinks that the second, new, prototype is best fit for technical devices, especially computers, with a lot of technical specifications. Thinks that every attribute is clearly displayed. Likes that it is possible to remove information from the second, new, prototype.

Three Important Things to Remember from the Test:

• The user likes to be able to add and remove information.

• The user wants to see the information for each product in rows rather than in columns.

• The user thinks that the prototypes are best fit for different scenarios. If the selection is narrowed down the existing solution is considered best fit. In cases where many products are compared the new prototype is considered best fit.
A.0.5 Test Person 5

Below the results from user test 5 can be found.

General Thoughts Before the Test: Test person 5 mentioned that if forced to buy a new computer he or she would:

- Generally start by reviewing his or her budget to see how much money is possible to spend on a computer, this is then used as a base for which computer to buy.

- After doing so, the user looks around online on different store websites to find a product that might be appropriate.

- The choice is based on what he or she is going to use the computer for and the user tries to find a product that matches these needs.

- After finding a product he or she is interested in and that seems appropriate, the user looks at a site for comparing products, for example www.prisjakt.nu or www.pricerunner.se. Based on the information found on said website, the user buys the product from the place selling it with the lowest price.

New Prototype, Comparing Laptops: When trying to find a product to purchase in the table containing laptops the test person:

- Likes that there are pictures showing each product.

- Likes the order of the attributes, price first since this is often prioritized by buyers.

- Likes that it is possible to remove rows directly in the table, and that it is possible to add the information back afterwards.

- Likes that the leftmost column with attributes is fixed when scrolling, thinks that generally in tables shown on small screens the headers scroll with the rest of the content and you end up having no idea what you are looking at.

- Would like to have more information present in the table, optionally that more information can be added by the user if he or she wants to. For example, attributes like operating system or similar.

- Would like to be able sort products based on attributes, for example based on price or similar.

- Likes that it is possible to manually move columns.
Existing Prototype, Comparing Tablets:  When trying to find a product to purchase in the table containing tablets the test person:

- Wants to have a picture for each product.
- Likes that the products are placed in rows rather than in columns. Makes it easier to read and more products are displayed simultaneously.
- Thinks that columns are better distinguished compared to the first, new, prototype.
- Wants to be able to add and remove information like in the first, new, prototype to customize the table.
- Wants to be able to move products and attributes around.
- Likes that other information is present in the table, for example operating system which was wanted in the first, new, prototype. Wants access to more thorough technical specifications.

Comparing Both Prototypes After the Test:

- Thinks that the second, existing, prototype was easier to read information from. Prefers having products and read the information horizontally instead of vertically.
- Likes the second, new, prototype better since pictures for all of the products were present.
- Likes the second, new, prototype better because comparing products is easier. Easier to see comparisons column wise.

Three Important Things to Remember from the Test:

- The user wants to read information horizontally instead of vertically.
- The user wants to be able to have more information in the table. He or she wants to be able to add and remove information in the table.
- The user wants to be able to sort products based on attributes, for example price, as an addition to moving columns manually.
A.0.6 Test Person 6

Below the results from user test 6 can be found.

General Thoughts Before the Test: Test person 6 mentioned that if forced to buy a new computer he or she would:

- Go to a physical store and ask people with knowledge about computers.
- After doing so, the user would check computers suggested online.
- The user has never used www.prisjakt.nu, www.pricerunner.se or similar sites to compare products.

Existing Prototype, Comparing Tablets: When trying to find a product to purchase in the table containing tablets the test person:

- Would like to be able to sort products based on attributes, for example name, price or similar.
- Would like to see specific products next to each other.
- Thinks that it is very unclear that it is possible to scroll sideways in the table. He or she initially thought that the part of the table seen on the screen was the entire table, more indication that the table continues is needed to not miss information by mistake.
- Thinks that there is so much information shown that it is hard to get an overview.

New Prototype, Comparing Laptops: When trying to find a product to purchase in the table containing laptops the test person:

- States that this prototype is preferred.
- Thinks that this solution provides a better overview of each product and its attributes.
- Thinks that it is easier to check all attributes for a specific product in this solution, but that it was easier in the first, existing, prototype to compare attributes for several products, for example the price of several products.
- Does not understand what the attribute ranking means, also thinks that it is unclear what the difference between memory and storage is, would like an explanation of the information.
- Does not initially understand what the arrows are for. When asked if products can be compared easily the user sees the arrows and understands what they do. He or she likes that it is possible to move products and place items next to each other to perform a better comparison. More clarification or hints needed to see that columns can be moved.
- Thinks that too many attributes are shown, but sees that rows can be removed and is happily surprised. Thinks that it is good that attributes can be removed and that a comparison between attributes considered important by each user can be performed.
Comparing Both Prototypes After the Test:

- Likes the second, new, prototype better because it is easy to retrieve and understand the information shown in the table.

- Thinks that it is much more clear in the second, new, prototype that it is possible to scroll sideways. In the first solution the table was believed to be shown in its entirety from the beginning.

Three Important Things to Remember from the Test:

- The user likes the new prototype better because it is easier to see and read the information in the table and thinks that it is easier to compare products in this prototype.

- The user likes that it is possible to remove attributes that are not considered important in the new prototype, but wants the functionality to move columns to be clarified.

- The user would like to be able to sort products based on attributes, for example price or name.
A.0.7 Test Person 7

Below the results from user test 7 can be found.

**General Thoughts Before the Test:** Test person 7 mentioned that if forced to buy a new computer he or she would:

- Go to a physical store and ask people with knowledge about computers, usually buys a computer there and then.
- Has not used www.prisjakt.nu, www.pricerunner.se or a similar site for purchases in the past but knows how the sites work.

**New Prototype, Comparing Laptops:** When trying to find a product to purchase in the table containing laptops the test person:

- Thinks that the width of the leftmost column with headers can be shrunk so that more products will fit on the screen simultaneously.
- Likes the table format and thinks that it is easy to read and retrieve information from the table.
- The information he or she wants to see is present in the table.
- Initially does not understand what the arrows to move columns and crosses to remove rows are for.

**Existing Prototype, Comparing Tablets:** When trying to find a product to purchase in the table containing tablets the test person:

- Likes that it is easier to compare one attribute between several products simultaneously. Likes that the attributes are scrolled and not the products.
- Thinks that the solution is easy to use and the information you want is present in the table.
- Thinks that the solution is easy to understand.

**Comparing Both Prototypes After the Test:**

- Easier to understand the information in the second, existing, prototype.
- Likes that pictures are shown in the first, new, prototype.
- After being asked if information can be compared easily the user realizes that this is possible in the first, new, prototype. Likes that information can be removed or moved.
- Initially disliked the first, new, prototype since comparing two products was hard if the products are placed far apart. After realizing that products can be moved to be placed next to each other the user thinks that it is much easier to compare products and likes the solutions more, but still thinks that it is easier to compare a larger number of products in the second, existing, prototype.
After trying the functionality to move or remove information offered in the first, new, prototype the user likes the solution better if only two or three products are to be compared. If more products were visible simultaneously the solution would be preferred for more products as well. The second, existing, prototype is still preferred if a larger number of products is to be compared.

Wants the crosses that indicate that rows can be removed to be made clearer so that a user realizes what they are for immediately. This also goes for the arrows indicating that columns can be moved. A suggestion from the user is that the crosses can be red instead of gray, since this color often indicates that removal is possible. Also thinks that the arrows can have some other color, or explaining text.

Three Important Things to Remember from the Test:

- The user thinks that showing products in columns is better when comparing a small number of products, for example two or three products. He or she prefers that products are shown in rows if more than three products are to be compared.
- The user thinks that crosses and arrows, and their function, should be clarified. Likes that it is possible to customize the new table.
- The user thinks that both of the table solutions are easy to understand, and that they are both preferred over table solution previously seen on mobile devices.
A.0.8  Test Person 8

Below the results from user test 8 can be found.

**General Thoughts Before the Test:** Test person 8 mentioned that if forced to buy a new computer he or she would:

- Ask friends for recommendations.
- Go to a store and ask personnel for recommendations based on the budget available.
- The user has not used www.prisjakt.nu, www.pricerunner.se or similar sites when buying a computer, but the sites have been used for other types of products.

**Existing Prototype, Comparing Tablets:** When trying to find a product to purchase in the table containing tablets the test person:

- Would like to see pictures of each product.
- Wants to be able to sort products based on attributes, for example price.
- Would like to see more attributes, for example if the tablets are best adapted for reading books, taking pictures etc.
- Likes that scrolling sideways is possible.
- Wants to see explanations of some attributes, for example what 4G LTE actually means, or longer titles that are more explanatory.

**New Prototype, Comparing Laptops:** When trying to find a product to purchase in the table containing laptops the test person:

- Likes that pictures are present, makes every product description better.
- Prefers that products are shown in columns instead of rows, likes that pictures are displayed at the top.
- Prefers that all attributes are shown in the leftmost column instead of in the top row.
- Prefers this solution.
- Thinks that this solution might be hard to use if a very large number of products is to be compared.
- Wants to be able to sort products based on attributes, for example name or price. In this way it would be possible to have all products from a specific brand next to each other.
Comparing Both Prototypes After the Test:

• Likes the second, new, prototype better.

• Thinks that it is easier to read and understand how the information is to be used in the second, new, solution.

• Would like to be able to place specific products next to each other, likes that about the second, new, prototype. Would prefer if the columns could be moved with, for example, drag and drop that can be shown with a textured surface or similar.

• Wants to be able to remove products.

• Likes that it is possible to remove attributes in the second, new, prototype. Makes it possible to customize the comparison to what you are interested in, and if you are lucky all attributes can fit on the mobile screen simultaneously and therefore scrolling up and down will not be needed.

• Thinks that it is unclear what ranking means among the attributes, wants to be able to click on each attribute and get an explanation of what it means. Ranking might be excessive, review more important.

Three Important Things to Remember from the Test:

• The user thinks that pictures are good to have.

• The user likes that products are shown in columns instead of rows. Likes that it is possible to customize the table but would prefer it if drag and drop was used instead of arrows.

• The user thinks that a lot of technical specifications makes the table cluttered, and would prefer if fewer attributes were shown. It might be better if the user could add extra information if wanted. Would like to be able to sort products based on attributes.
A.0.9  Test Person 9

Below the results from user test 9 can be found.

General Thoughts Before the Test:  Test person 9 mentioned that if forced to buy a new computer he or she would:

- Look at what he or she needs a computer for.
- Thinks that weight is an important attribute, a computer can not be too heavy.
- After this, the user would look online to find a computer that fits said needs and has a good price.
- The user has used www.prisjakt.nu, www.pricerunner.se or similar sites when purchased products, but not when buying a computer.

New Prototype, Comparing Laptops:  When trying to find a product to purchase in the table containing laptops the test person:

- Thinks that it is hard to compare two computers on a mobile screen, easy that the products end up very far apart and you end up not knowing where to look and how to compare the products.
- Likes that every second row is marked in light gray, makes it easy to see which row is which.
- Likes that scrollbars appear when scrolling sideways, makes it clear how much information is present and where you are currently placed in the table.
- Initially did not understand what the arrows in each column are for, but when trying them out realizes that they move columns. Likes the function and wants it in tables shown on mobile screens, but would like it if the function could be shown more clearly.
- Likes the order of the attributes.

Existing Prototype, Comparing Tablets:  When trying to find a product to purchase in the table containing tablets the test person:

- Wants to be able to see a picture of each product.
- Likes the order of the attributes.
- Wants to be able to change the order of products, which was possible in the first, new, prototype. This is especially needed when enough products are present to take up more height than what the screen offers.
- Would like to be able to mark favorites and compare these more thoroughly.
Comparing Both Prototypes After the Test:

- Wants the crosses to remove rows to be marked more clearly.
- Thinks that it was easier to understand and read information from the first, new, prototype. Liked that pictures were present in the first, new, prototype.

Three Important Things to Remember from the Test:

- The user thinks that a picture of each product is good to have, it makes it easier to retrieve information about a product. He or she also thinks that design is considered important for every product.
- The user likes that it is possible to place products next to each other to make the comparison easier in the new prototype.
- The user wants crosses and arrows to be made clearer so that their function is more apparent.
A.0.10 Test Person 10

Below the results from user test 10 can be found.

General Thoughts Before the Test: Test person 10 mentioned that if forced to buy a new computer he or she would:

- Ask friends who have knowledge about computer for help.
- Go to a physical store and ask for help.
- Check online. If the user has gotten valuable help from friends, he or she uses www.pricerunner.se or a similar website to find the best price for that product. If no valuable help has been received, the user looks online for products matching his or her budget and needs.

Existing Prototype, Comparing Tablets: When trying to find a product to purchase in the table containing tablets the test person:

- Likes that it is possible to scroll sideways and that each product’s name stays on the screen when scrolling, thinks that normally the name follows when you scroll and then you forget which product is which.
- Likes the order of the attributes.
- Thinks that it is possible to get the information you need from the table to make an informed decision.
- Thinks that the ranking attribute is unclear, would like explanations for some of the attributes.
- Wants to be able to sort products based on attributes, for example price.

New Prototype, Comparing Laptops: When trying to find a product to purchase in the table containing laptops the test person:

- Likes the first, existing, prototype better.
- Likes that this prototype includes pictures of each product.
- Understands what the arrows to move columns do, but prefers having products placed in rows instead of columns because the comparison is easier to make then. Prefers comparing one attribute for many products instead of few products and many attributes.
- Thinks that comparisons are easier to make in the first, existing, prototype. Can only see two or three products at a time here and wants to be able to compare more products than this.
- Thinks that the layout in this product is well-made.
- Likes that every second row is light gray, makes it easy to see which line is which.
• Likes that it is possible to remove attributes, understands what the crosses on each line is for immediately.

• Thinks that it is good that attributes can be put back into the table after being removed, for example if you accidentally remove the wrong attribute.

• Likes that it is possible to remove all of the attributes and then just put the ones you are interested in back.

Comparing Both Prototypes After the Test:

• The user prefers to have products placed in rows instead of columns.

• The user wants to be able to remove products as well as attributes. Too time consuming to move the objects you are not interested in to the farthest right, would like to remove them directly in the table instead. Thinks that if products could be removed, a possible scenario is that you remove all products that are not of interest and that you might end up with only one or a few products left that are relevant to you. If this happens, it is easier to compare the products in the second, new, solution and you can decide which product fits you best more easily.

• Wants to be able to remove products and attributes in the first, existing, solution as well.

Three Important Things to Remember from the Test:

• The user wants to see products placed in rows.

• The user wants to be able to both add and remove attributes, and thinks that it is good that a user can customize a table to his or her liking. Wants to be able to sort products based on attributes.

• The user wants to be able to remove products directly in the table as well. By doing this, a user can remove products that are not of interest and can easily get a smaller selection of products to choose from, which will make the decision easier to make.
A.0.11 Test Person 11

Below the results from user test 11 can be found.

**General Thoughts Before the Test:** Test person 11 mentioned that if forced to buy a new computer he or she would:

- Ask his or her brother who knows a lot about computers for help.
- Look online for a computer that was considered appropriate.
- Has used www.pricerunner.se and similar sites to compare computers in the past, often when a specific product has been chosen and a place to buy it from is sought for.

**New Prototype, Comparing Laptops:** When trying to find a product to purchase in the table containing laptops the test person:

- Thinks that it is hard to get an overview of all the information present in the table.
- Wants to be able to sort products based on attributes, for example price or user rating.
- Thinks that every product is displayed very nicely, gets all of the information he or she wants about a product from the table.
- Does not see the arrows and crosses and does not realize what they are for.

**Existing Prototype, Comparing Tablets:** When trying to find a product to purchase in the table containing tablets the test person:

- Wants to be able to sort products based on attributes, for example price.
- Thinks that the first, new, solution provides a better overview of the attributes that can be used for comparisons.
- Does not care about the product’s name and does not think that the names must stay on the screen when you scroll through the remaining table content.
- Comments on the fact that this solution does not contain any pictures, the user thinks that pictures are not needed in any of the solutions.

**Comparing Both Prototypes After the Test:**

- The user likes the first, new, solution better.
- Thinks that the first, new, prototype provides a better overview of every product.
Three Important Things to Remember from the Test:

- The user thinks that the new prototype provides a better overview of every product.

- The user thinks that pictures are not needed but would like it if products could be sorted based on attributes.

- The user does not understand what the crosses and arrows to remove and move content are for.
A.0.12 Test Person 12

Below the results from user test 12 can be found.

**General Thoughts Before the Test:** Test person 12 mentioned that if forced to buy a new computer he or she would:

- Find out what he or she needs the computer for and then look online to try and find reviews and rankings for computers that match these needs.
- Look at forums and similar to find what other users have thought about computers. After finding an appropriate computer, [www.pricerunner.se](http://www.pricerunner.se) is used to find a good price for the product.

**Existing Prototype, Comparing Tablets:** When trying to find a product to purchase in the table containing tablets the test person:

- Wants to be able to sort products based on attributes, for example price. Thinks that comparing products will be easier if this is possible.
- Thinks that the solution is easy to understand.
- Likes that it is possible to see if the products are in stock and in how many stores.

**New Prototype, Comparing Laptops:** When trying to find a product to purchase in the table containing laptops the test person:

- Likes that it is possible to see pictures.
- Initially thinks that a lot of information is presented and that it becomes hard to see due to this.
- Wants to be able to sort based on attributes.
- Thinks that the layout differs from how similar information is usually displayed, products in columns instead of rows is not familiar.
- Likes that every second row is light gray, would like it if columns were more clearly marked as well.
- Does not initially understand what the arrows to move columns are for. Realizes after a while and really likes that it is possible to place items next to each other, but thinks that the arrows’ functionality must be clarified.
- Wants to be able to choose specific products into a shopping basket or similar and compare the chosen products more carefully.
- Understands what the crosses to remove rows are for immediately, really likes that it is possible to remove attributes that are not of interest.
- Likes this solution more when realizing that attributes are removable and that products can be placed next to each other.
Comparing Both Prototypes After the Test:

- The user likes the second, new, prototype better when realizing that it is possible to customize the solution a lot by only viewing attributes of interest and placing interesting products next to each other.

- The user likes the second, new, solution much better when realizing that products can be placed next to each other. By doing this, a lot less scrolling is needed to compare products of interest.

- If the user would use a solution for comparing products on a smartphone, he or she would use the second, new, prototype. If the first, existing, solution was used the user would prefer to look at the table on a desktop computer where more attributes can be seen simultaneously.

Three Important Things to Remember from the Test:

- The user likes the functionality provided by the arrows, but would like the fact that columns can be moved clarified.

- The user wants to be able to sort products based on attributes.

- The user likes that it is possible to customize the table in the new solution, wants to be able to do this in a solution viewed on a smartphone with a small screen.