

University of Cincinnati

Date: 3/25/2013

I, Da Shen, hereby submit this original work as part of the requirements for the degree of Master of Design in Design.

It is entitled:

Comparative Evaluation of Repurposing and Optimized Approaches in Web Application Design

Student's name: **Da Shen**

This work and its defense approved by:

Committee chair: Benjamin Meyer, MFA

Committee member: Heekyoung Jung, PhD



3266

Comparative Evaluation of Repurposing and Optimized Approaches in Web Application Design

A thesis submitted to the Graduate School
of the University of Cincinnati
in partial fulfillment of the requirements for the degree of
Master of Design
in the School of Design
of the College of Design, Architecture, Art and Planning
2013
by
Da Shen
Bachelor of Science, Beihang University, China, 2010

Thesis Committee:

Benjamin Meyer, MFA

Heekyoung Jung, PhD

Abstract

Given the emergence of mobile technology, the difference of devices and their adjunct operating systems have been progressively enlarged. On devices with varying screen sizes, user interaction and user experience become different. This makes web application design a more complicated task than before in order to meet various compatibility and user experience requirements. To fix this issue, web application design approaches have evolved into two categories: repurposing approach and optimized approach. In this study, I design and develop a cross-device web application by using these two approaches respectively. Usability testing is performed to collect data and user experience comments from respondents. Then analysis of the data shows which approach is more superior in specific situations.

Keywords: web application, repurposing design, optimized design, responsive design, usability testing

Acknowledgments

Thanks to my committee members, Ben and Heekyoung who gave me great advice on completing this thesis.

Thanks to my friend Yao Kong who provided me many insights on constructing the thesis application concept.

Thanks to my parents who always support me especially on those hard days.

Table of Contents

CHAPTER 1 INTRODUCTION	1
1.1 Web Application for varying screen size devices.....	1
1.2 Design Approaches for Cross-Device Web Applications.....	2
CHAPTER 2 METHODOLOGIES.....	5
2.1 Repurposing vs. Optimized Design Approaches	6
2.2 User-Centered Design.....	7
CHAPTER 3 CASE STUDY: DESIGN OF STOCK TRACKER WEB APPLICATION	9
3.1 Design Requirements.....	9
3.2 Approach 1: Repurposing Design.....	20
3.3 Approach 2: Optimized Design	23
CHAPTER 4 DEVELOPMENT	28
4.1 Introduction: Web Technology.....	28
4.2 Data Structure and Connection	30
CHAPTER 5 USABILITY TEST	33
5.1 Objectives	33
5.2 Planning	33
5.3 Execution	35
5.4 Analysis	36
CHAPTER 6 CONCLUSION	39
BIBLIOGRAPHY	44

List of Figures

Figure 1. Application design info structure.....	5
Figure 2. Initial wireframe	12
Figure 3. Final wireframe.....	12
Figure 4. Application structure	13
Figure 5. Main page of the smartphone version.....	14
Figure 6. “My List” page of the smartphone version.....	15
Figure 7. “My List” detailed information	16
Figure 8. Functions of the “Comparison Group” section.....	18
Figure 9. Functions in a default group page.....	19
Figure 10. Width standards of different device groups.....	20
Figure 11. Relocate the main menu for the tablet version	21
Figure 12. The application width is 80% of the browser width on the laptop or desktop version.....	22
Figure 13. Design thinking of optimized approach.....	24
Figure 14. Optimized interface design for tablets	25
Figure 15. Subordinate information of a stock.....	26
Figure 16. Subordinate information of a default group.....	26
Figure 17. All information on one screen for the laptop or desktop version	27
Figure 18. Four technical layers of the stock tracker	28
Figure 19. The data structure and connection of the application	31
Figure 20. The connection between application interface and database.....	32
Figure 21. The photo of usability test	35
Figure 22. Chart of the usability test analysis	38
Figure 23. A chart to illustrate the scores of all attributes for the four versions.....	40
Figure 24. A chart to illustrate the total scores and section scores of the four versions	41

Chapter 1 Introduction

1.1 Web Application for Varying Screen Size Devices

In the beginning of 21st century, user interaction with personal computers (PCs) is not only performed on devices like desktops and laptops but also on smartphones and tablets. Meanwhile, accompanied with these varying screen size devices, various operating systems were keeping released, like iOS, Android, Windows 8, Mac OS and Linux. With the support of these operating systems, applications are widely used by people to perform specific tasks and have become one of the most essential features of information technology market. One of the main platforms for applications is web.

The Web, with the features of high accessibility, compatibility and lightweight, has been the bridge to distribute applications on different operating systems. “The development of applications on the World Wide Web is not a recent topic. For more than 15 years, computer scientists and practitioners have discovered and promoted the use of the Web as a platform for application development” (Corral, Sillitti, Succi, Garibbo, & Ramella, 2012). Shubin and Meehan (1997) indicated “the induction of dynamic presentation layers and scripting languages enabled web systems to conduct simple tasks, such as filling in a form, and soon even more complex jobs like managing a database or execute complex functions”.

“The Web-based application paradigm works by utilizing the web browser as an intermediate level of abstraction that permits to implement the logic layer based on a

scripting language (e.g., JavaScript) and the presentation layer under HTML and CSS, making it easily portable to different web browsers” (Corral et al., 2012). No matter what device it is, it has to have a browser to access the internet and web applications can be accessed by browsers that connected to the internet. This access provides web applications great accessibility on different platforms and devices. After accessing, the next thing is compatibility, thanks for the consistent improvement of web technologies and the rapid evolution of web browsers, the compatibility of web applications is becoming stronger in different browsers and many advanced functions can be achieved regardless of the browsers brands. As Charland and LeRoux (2011) stated, “As even if a browser does not support a native compatibility, it’s not because it cannot or that it won’t; it just means it hasn’t been done yet”. It’s a matter of time for web to become a highly compatible platform and providing web applications a great place to work.

1.2 Design Approaches for Cross-device Web Applications

In web applications design and development, the main difference of devices is determined by screen size. In this thesis, I classified all screen sizes into three groups:

- Smartphones;
- Tablets;
- Laptops/desktops.

You might doubt that screen size of smartphones ranges from 2 inches to 4 inches; there are tablets of 7 inches and 9 inches; laptops and desktops can range from 11 inches to 27

inches. However, although screen sizes are still various in each of the above group, device features and interaction within each group are very similar. In addition, new devices with new screen sizes are continuing released, it might take forever to make a stable and comprehensive list of screen size, which might not make more sense than my rough list. Like Smus (2012) said, “This is only one of many possible breakdowns, but one that makes a lot of sense at the time of writing”. Taking these issues into consideration, a smartphone, a tablet and a laptop will be the three standard comparison groups in this thesis.

Although can be easily distributed on devices of different groups, web applications do not spontaneously fit the usability requirements of each group. It is designers and developers' obligation to optimize the user experience of an application on different devices by using certain user experience strategies. Based on Nielsen's (2012) study:

When it comes to user experience strategy, there are two opposing schools:

- Repurposing: make as few designs as possible — preferably only one — and reuse the same material across as many platforms as possible.
- Platform optimization: design different user interfaces for each main platform, integrating the user experience layers as tightly as possible.

It is necessary to clarify that repurposing design is not equal to responsive design but contains responsive design as one of its strategies. Responsive design is a popular design

strategy including several web technologies to make web pages adapt to different kinds of devices.

Either repurposing or platform optimization approaches is valid to accomplish the cross-device features of a web application. In Nielsen's (2012) opinion, “Repurposing has huge cost advantages. Most of the work must be done only once. But on most platforms, the outcome tends to be a substandard user experience”, and vice versa. It is a reasonable conclusion that optimized approach will give a better cross-device user experience because designers will "integrate the user experience layers as tightly as possible" (Nielsen, 2012) for each device group. Similarly, as Smus (2012) stated, “There is a fundamental tradeoff here: the more device categories you have, the better a user experience you can deliver, but the more work it will take to design, implement and maintain”. In this thesis, I will apply the two design approaches into cross-device web application development, following usability testing and analysis, and then comparing the analysis results of both approaches. The final analysis results will help people choosing the proper approach when they design and develop web applications. I assume my thesis providing the following contributions:

- Figure out and approve the features of repurposing and optimized design approaches.
- Provide detailed standards as a reference for people to choose the proper design approach when designing web applications.

Chapter 2 Methodologies

In this thesis, I used repurposing and optimized design approaches to design two cross-device applications respectively, named demo 1 and demo 2. Here “cross-device” means the application can run on smartphones, tablets and laptops/desktops and meet the user experience requirements of each kind of devices. User-centered design will be used as the primary research method to meet the design requirements of the application. Figure 1 shows the application design info-structure.

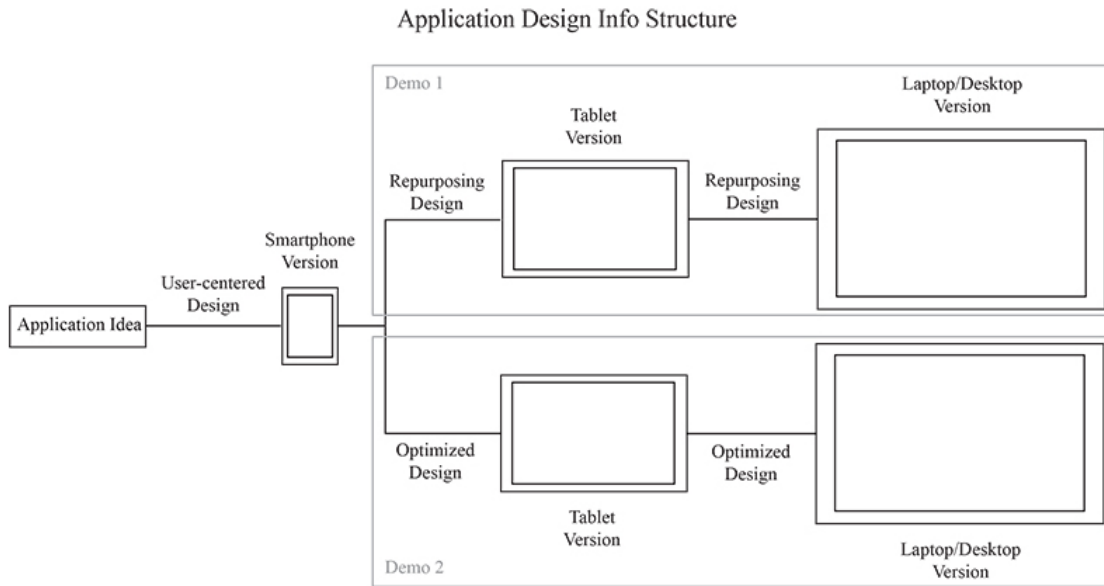


Figure 1. Application design info structure

As it shows in Figure 1, user-centered design was performed at the very beginning to get insights of users to construct the smartphone version of the application. This smartphone version is the fundamental version shared by both demo 1 and demo 2. Then, for demo 1, use repurposing approach to maximally reuse the content of its smartphone version,

transplanting it into the tablet version and the laptop version. On the other hand, to apply the optimized design approach on demo 2, I developed the optimized tablet version and the optimized laptop version. The point of optimized design is focus on deliver the corresponding user experience of specific device but not maximally reuse content.

Although each approach has its characterization, they are not exclusionary to each other.

For example, when doing repurposing design, I should also consider user experience of specific devices; when doing optimized design, I can reuse content as well. In the rest of this chapter, I will introduce these design methods and how will I apply them.

2.1 Repurposing vs. Optimized Design Approaches

Repurposing design and optimized design are the main methodologies of this thesis.

When applying repurposing design, a designer should first design for one screen size device, and then maximally transplant the design and content of this version into other screen size devices. I chose media queries, a technology of responsive design to perform this transformation.

World Wide Web Consortium (2012) defined media queries:

A media query consists of a media type and zero or more expressions that check for the conditions of particular *media features*. Among the media features that can be used in media queries are ‘width’, ‘height’, and ‘color’. By using media queries, presentations can be tailored to a specific range of output devices without changing the content itself.

Besides media queries, I think repurposing design is a collection of methods that can make an application more adaptive on different size screens. For example, according to Urban Dictionary, “fat finger” is “A specific kind of typo where one stroke presses two keys instead of the intended one” (“Fat Finger,” n.d.). To perform touch gestures correctly, a button should be big enough to prevent the "fat finger" problem. Another method called “content strategy” which means the content should be simpler on smaller screens than it is on larger screens. Because according to the smaller screen size, keeping all the same information as it is on a larger display can be inconvenient to read. Most importantly a designer should keep the concept of repurposing design in mind in order to grab even a little method to make a web application adaptive.

On the other hand, the concept of optimized design is pretty straightforward. As it literally means, optimized design focuses on delivering optimized user experience on a specific platform. Designers customize an application to fit the usability of specific platforms. But it does not mean they cannot reuse content between different platforms. Reusing content is just not the main point of optimized design. On demo 2, optimized approach will be performed in the design of both of the two device groups.

2.2 User-Centered Design

User-centered design is performed as the primary research method to get insights from users. As Wikipedia stated, “In broad terms, user-centered design is a type of user interface design and a process in which the needs, wants, and limitations of end users of a product are given extensive attention at each stage of the design process”

(“User-Centered Design,” n.d.). In fact, today this design method has become a lot broader. It is practiced in nearly all of the design fields especially in application design. Because in application design, “It is often very difficult for the designers of a product to understand intuitively what a first-time user of their design experiences, and what each user's learning curve may look like” (User-Center Design, n.d.).

The main point of user-centered method is designers ingratiate users but not users follow designers. Each application performs some kind of tasks. If it is confusing to understand and use at the first time, users might leave and never return. To attract users, designers have to look from a user's perspective and study user behavior to understand users and make user-friendly design. Thus I choose user-centered design as my fundamental design method to construct the application concept and build the smartphone version of the application.

Since this thesis just uses user-centered design as a tool to get user insights, but does not focus on how to perform user-centered design, I used the basic and necessary features of user-centered design, like interview, analysis and prototype but not a comprehensive collection of features of this design method. My design research process is:

- User interview
- Insight analysis
- Wireframe
- User interview
- Insight analysis

- Refine wireframe
- Design
- Implementation
- Usability test

After introducing the main methods. Research and design will be performed in next chapter.

Chapter 3 Case Study: Design of Stock Tracker Web Application

I applied repurposing design and optimized design approaches on a web application, named Stock Tracker. Users can use it to track stock information of specific companies.

3.1 Design Requirements

Stock tracker applications are not a new idea in the application industry. There are stock trackers with various functionalities for almost all of the operating systems. The objective of my stock tracker is to practice different design approaches on it. So I decided to keep it simple and functional.

To construct the application, I performed user-centered design methods to understand user needs. My target users should have medium or advanced knowledge of stock

investment. Field research was conducted with interviews and questionnaires. Here are some of my typical questions:

- Talk about your main investment activities?
- How do you track stock information?
- How often do you track specific stocks?
- If use a stock tracker, what functionalities do you need?
- On what kind of devices would you usually use a stock tracker application?

I interviewed several users for this study. Some of them graduated with degree in investment banking, and the others are casual investors. By analyzing their answers, I had the following insights:

1. Stock market has a long history. It has some traditions and rules, no matter visually or functionally, and investors have got used to these traditions. It is better to follow but not break the rules when designing a stock tracker, or users would feel uncomfortable with the new design and refuse to use it.
2. When tracking a stock, investors are not only satisfied with basic stock data, like current price and change, but need a more comprehensive information of the stock, including most of the detailed stock data, the graphic chart and related news. Although the information of a stock can be a lot more than this, these three parts are much enough for a lightweight application.
3. There are four kinds of stock charts in the industry. They are line charts, bar

- charts, candlestick charts, and point and figure charts. The target users of my study prefer candlestick charts for it can represent more comprehensive information than the others.
4. Search and store stocks should be the most fundamental function of a stock tracker.
 5. The related news of a company is very helpful for analyzing its stock.
 6. Comparing stocks from the same industry can provide insights for users.

Then, the following application structure was developed according to the above insights.

The final design includes the following features:

1. It has two main functions. The first function let users to search and store stocks as a list. The second one let users to make stock groups and compare stocks within each group.
2. When looking into a stock, users are able to see some detailed stock data, a candlestick chart of the stock and the related news of the stock.
3. The visual design of this application should follow the traditional features of stock market.

With these features as the guideline, I built the initial wireframes of the application. A second round of user interviews was conducted based on the initial wireframe. This iterative process of design and user feedback continued until the wireframe was satisfactory. Figure 2 and Figure 3 shows the draft of the initial wireframe and the final

wireframe respectively.

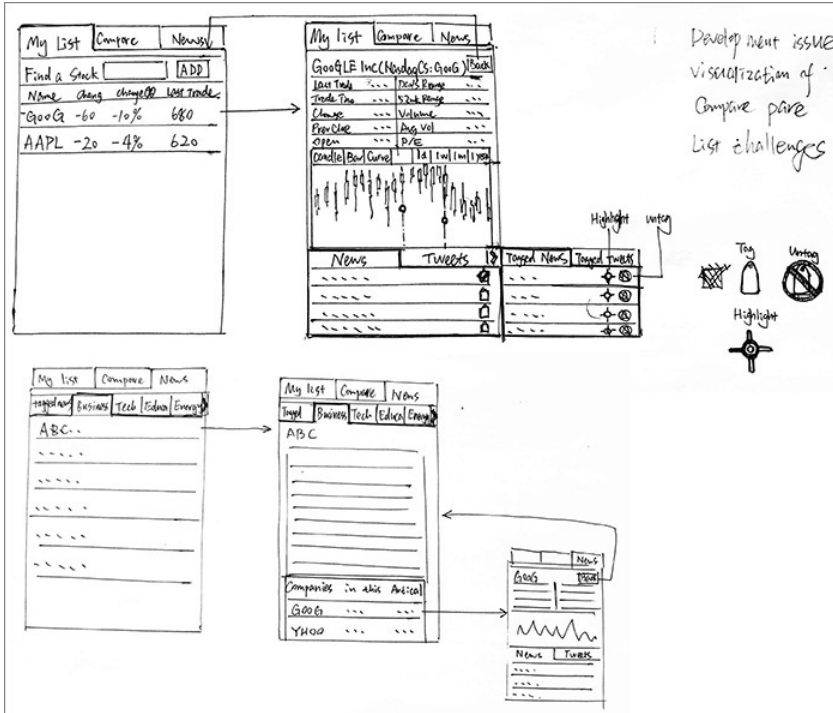


Figure 2. Initial wireframe

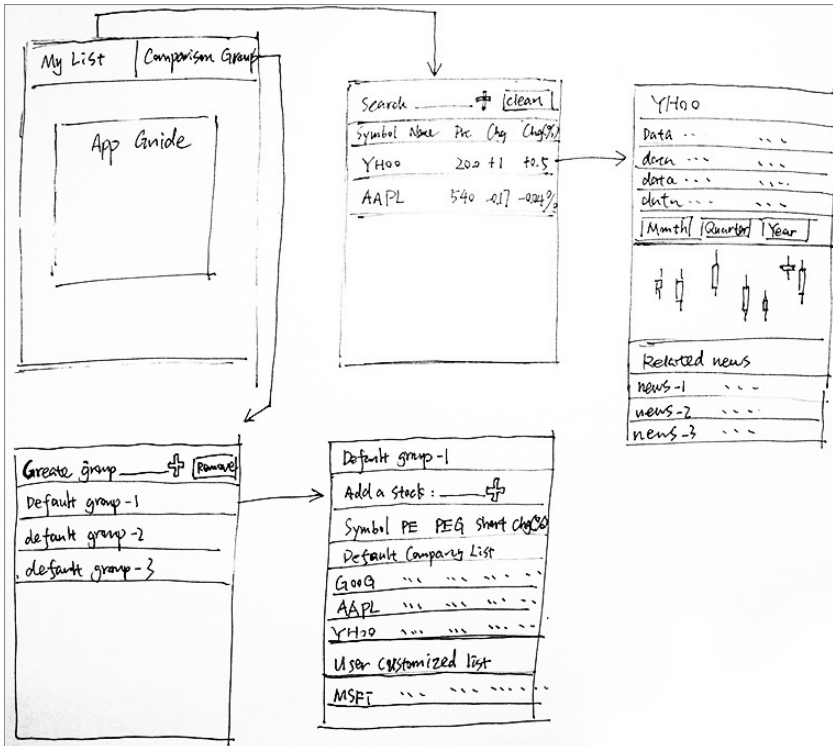


Figure 3. Final wireframe

Figure 4 illustrates the application structure:

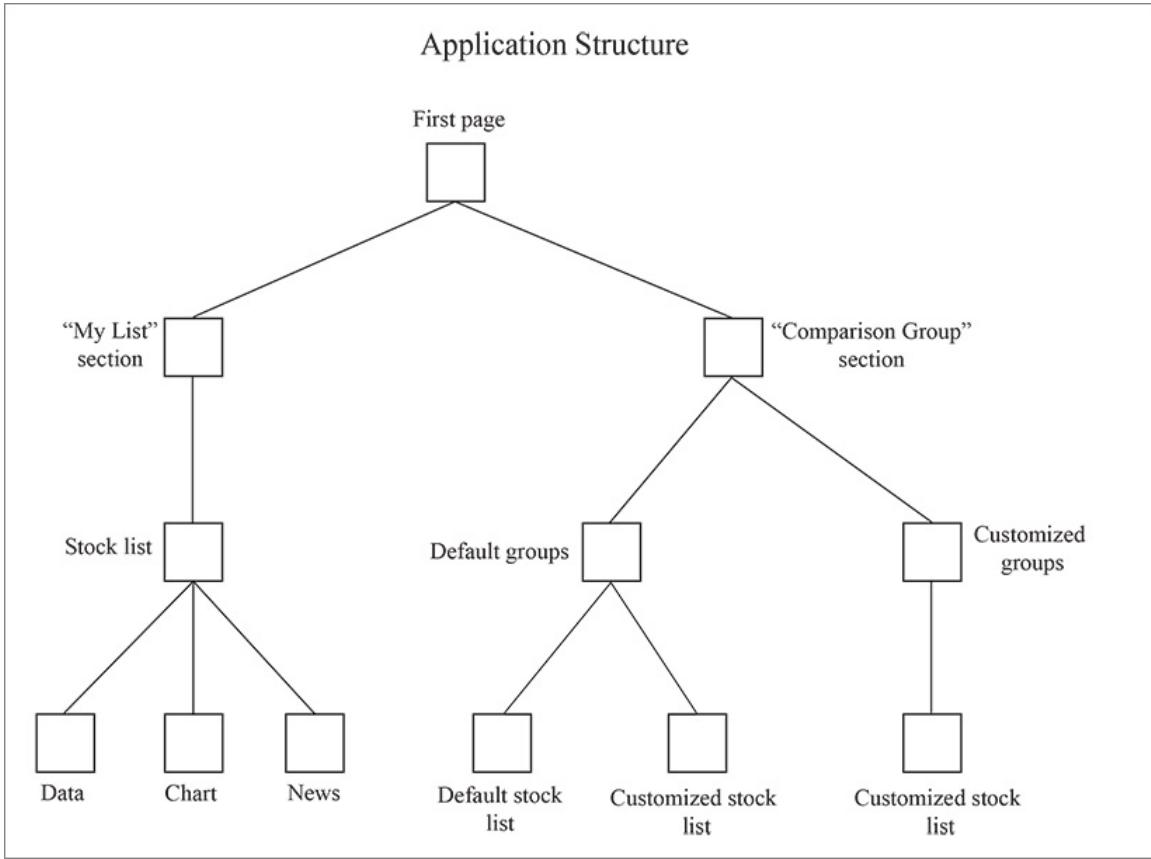


Figure 4. Application structure

On the first page, there is a main menu with two main functions, named “My List” and “Comparison Group”. Under the main menu is the application guides which indicate the functionalities of each section and how to use them. See Figure 5.

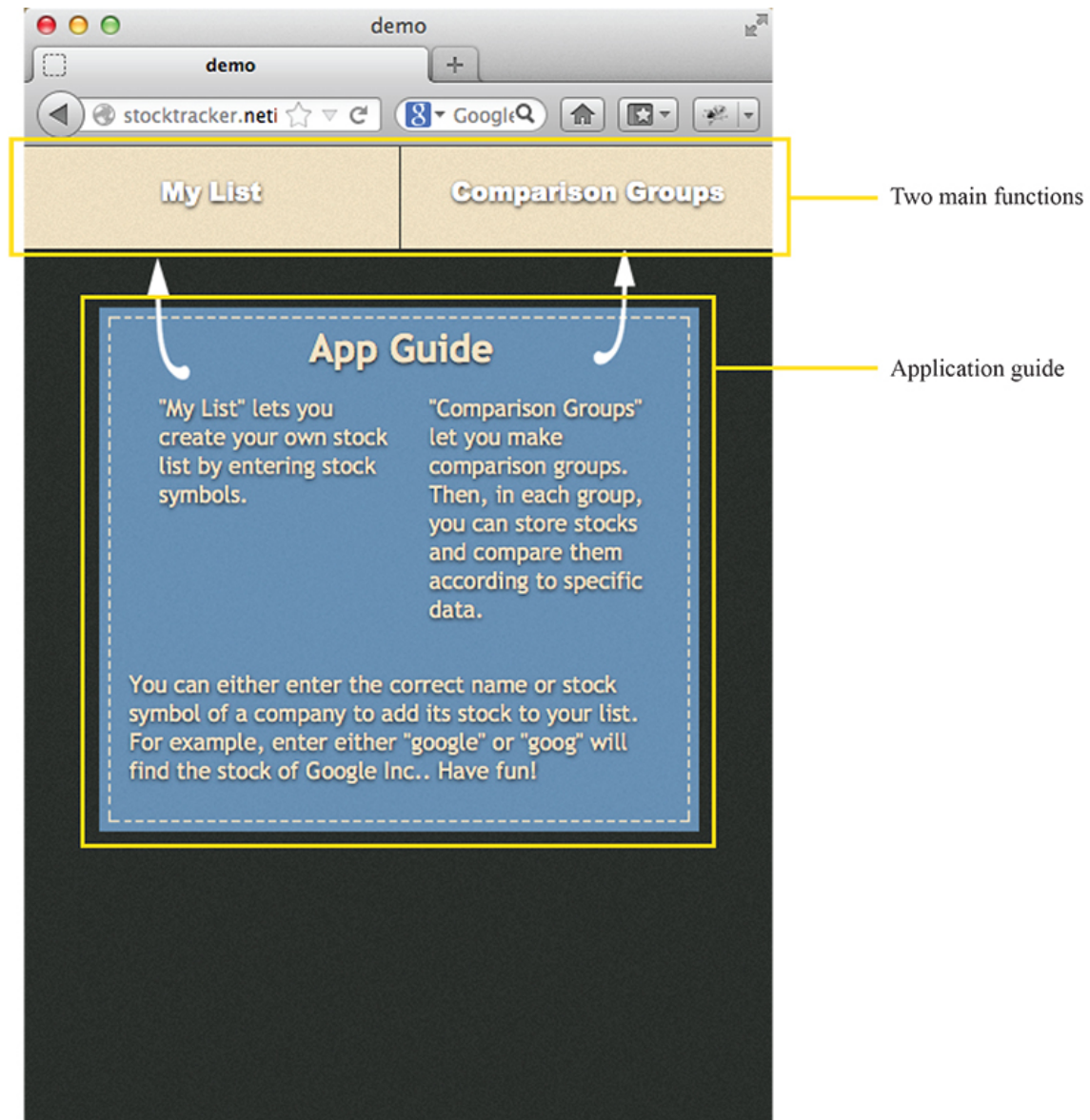


Figure 5. Main page of the smartphone version

After click into “My List” section, there is an input bar let users enter the name or symbol of a company. The company that users found will be listed in the dark area. On the right of the input area is a button called “clean my list” which can delete the current stock list. Below the input bar is another bar with different kinds of data names on it, including “Symbol”, “Name”, “Price”, “Change” and “Change%”, each one on the top of a column.

The corresponding data of each stock will be showed in each column, under the data names. See Figure 6.

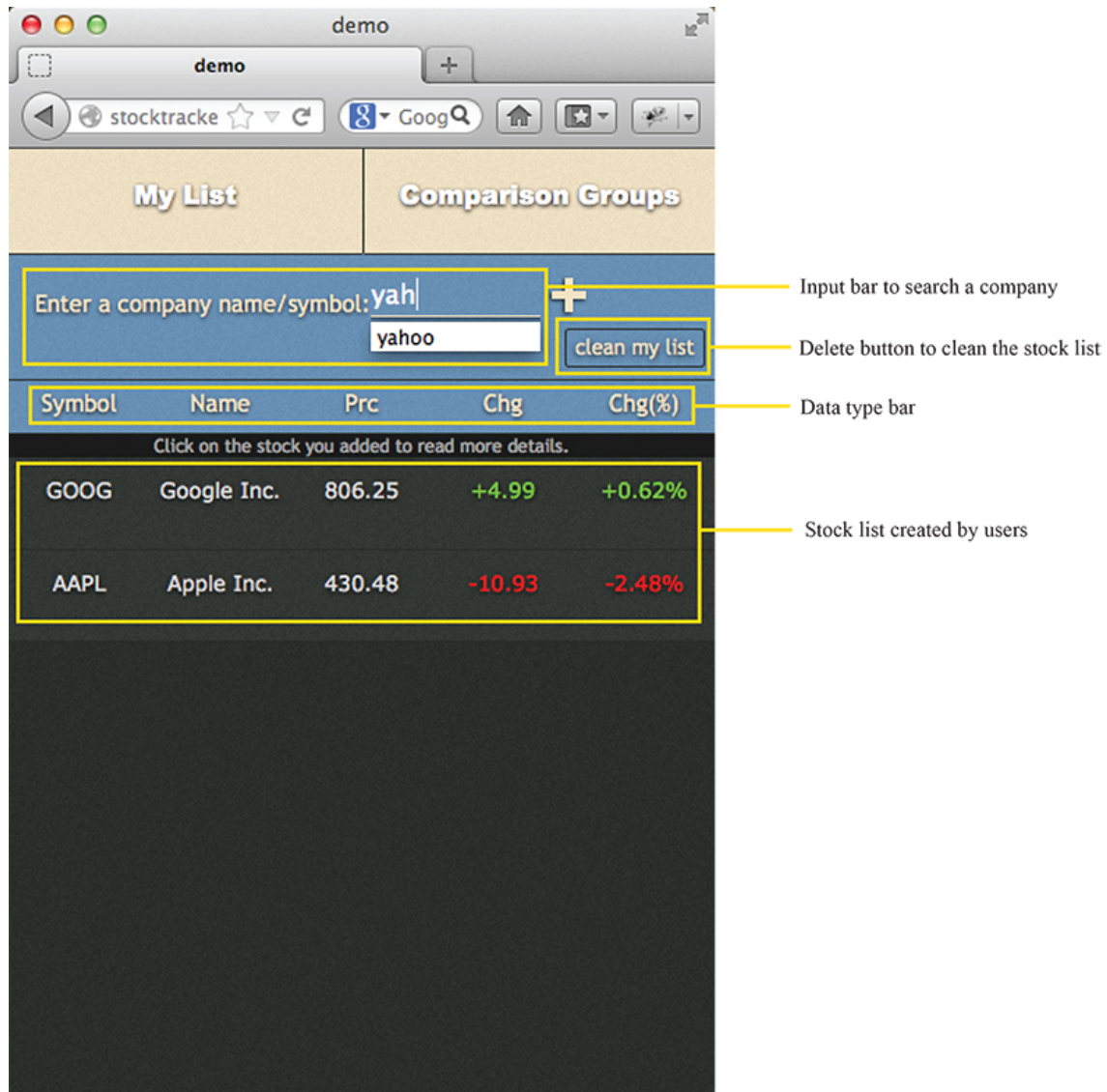


Figure 6. "My List" page of the smartphone version

Then by clicking on each stock the application goes to the stock detailed information page. The detailed information includes some comprehensive stock data, a candlestick chart based on timeline, and the news related to this stock. In the candlestick chart users

can choose time span from past one month to one year. By click on the related news, the page will be redirected to the original web page of the news. As Figure 7 shows:

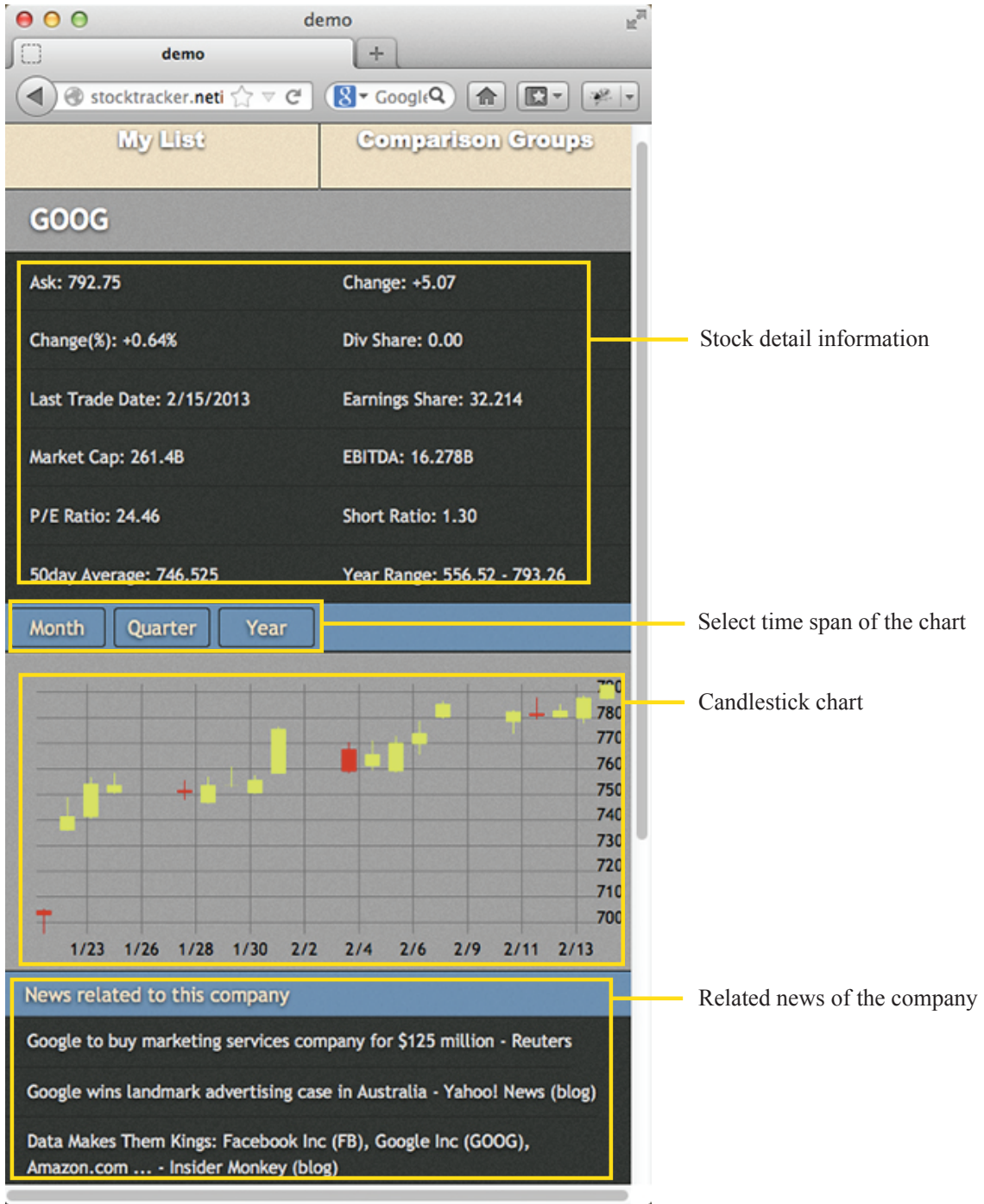


Figure 7. "My List" detailed information

Another main functionality is called “Comparison Group”. In “Comparison Group”, users can search some companies in the same industry and put them into a comparison group. Then open the group and compare these companies according to specific data. There are also some default groups. By click and go into each group, users can get a list of default companies. They are the famous companies in their fields and can be used as references when users comparing stocks. In addition, some stock data names (PE Ratio, PEG Ratio, Short Ratio, Change%) are been showed for users to compare stocks. According to the user research, these stock data is considered vital when comparing stocks. Besides the default company list, users can search and add companies, customizing their own company list. Also, if users do not want to use the default groups, they can create their own groups. Figure 8 and Figure 9 shows the functions in the “Comparison Group” section.

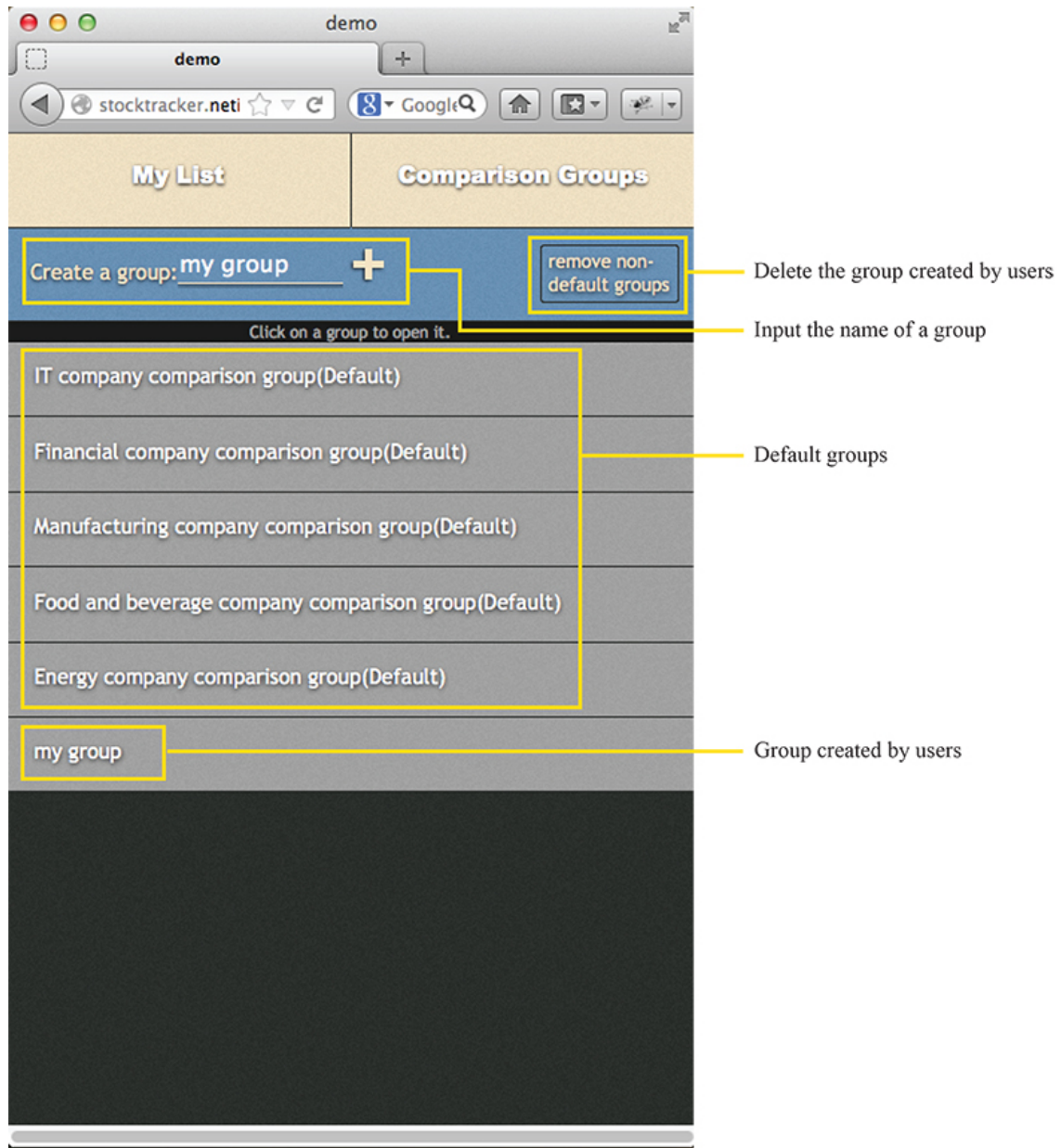


Figure 8. Functions of the “Comparison Group” section

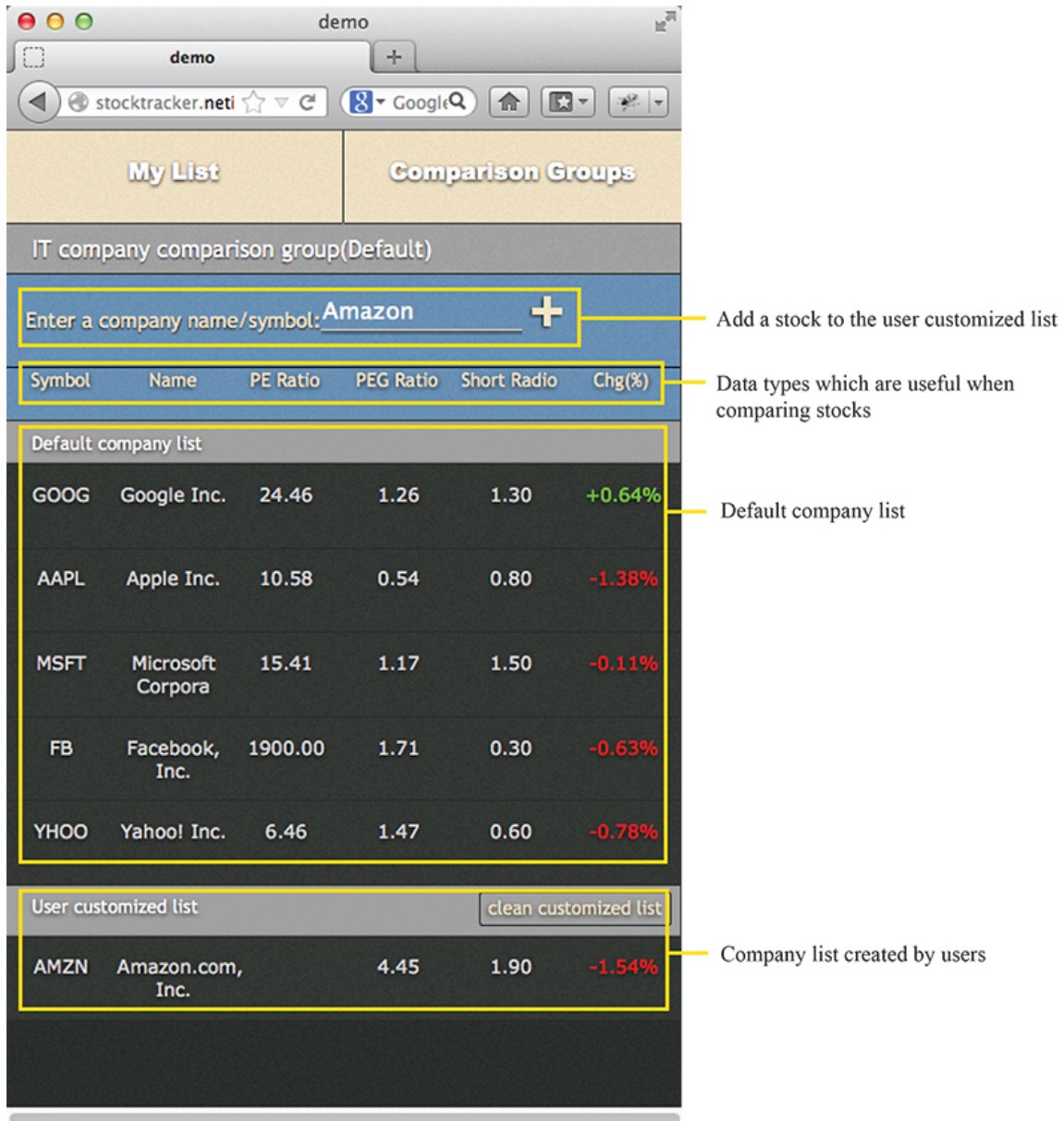


Figure 9. Functions in a default group page

In this application description, I went through all the design and functionalities of the stock tracker. The smartphone version of it is finalized. But as a cross-device application, how does it look and perform on tablets and laptops? The answer is implement repurposing design and optimized design on it.

3.2 Approach 1: Repurposing Design

Repurposing design involves developing a single fluid design that can be presented on multiple devices. According to this strategy, I decided to not change the elements but only the layout of the user interface components for the tablet and laptop versions.

Before dive into repurposing design, I need to set up the screen size standards of smartphones, tablets and laptops. As follow:

- Smartphone: $0 < \text{screen size} < 480\text{px}$;
- Tablet: $480\text{px} \leq \text{screen size} < 1024\text{px}$;
- Laptop: $\text{screen size} \geq 1024\text{px}$.

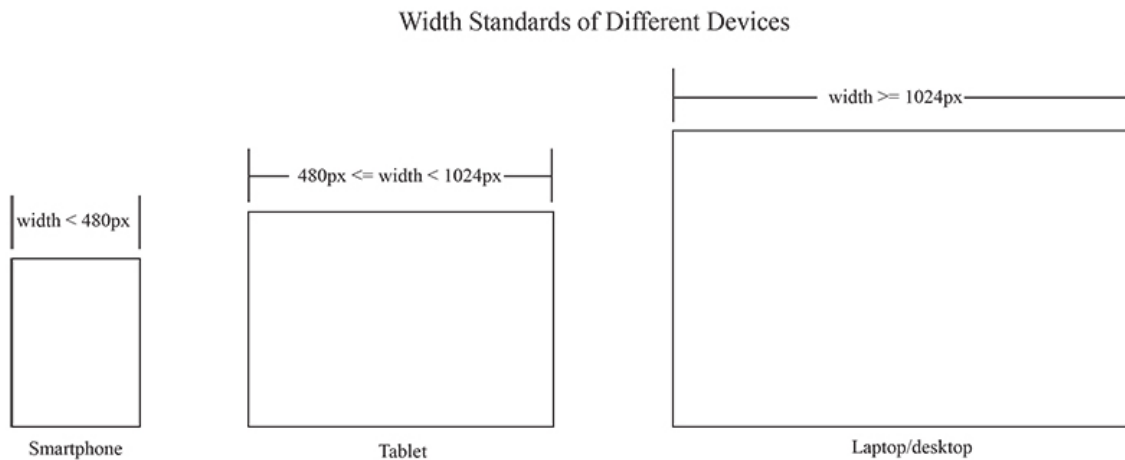


Figure 10. Width standards of different device groups

The boundary numbers of this standard are flexible and should only be considered as a reference because there is not an exact width standard for those devices. I choose these boundary numbers just because they are reasonable pixel widths for smartphones, tablets and laptops/desktops.

Then I used media queries, a web technology to change the layout according to specific platforms. “Media Queries is a CSS3 module allowing content rendering to adapt to conditions such as screen resolution” (“Media Queries,” n.d.). In another words, I wrote three CSS style sheets named style-small.css, style-middle.css and style-large.css. Then wrapped them by three different media queries. When the browser size matches the specific width standard, the corresponding media query will request browsers to pick the proper style sheet from the three style sheets and render it.

For tablet interfaces, my target display proportion is landscape. When the width of the screen exceeds 480px and smaller than 1024px, I used a media query to move the main menu from top to the left side, in order to keep the visual design stay in a good proportion. As it shows in Figure 11.

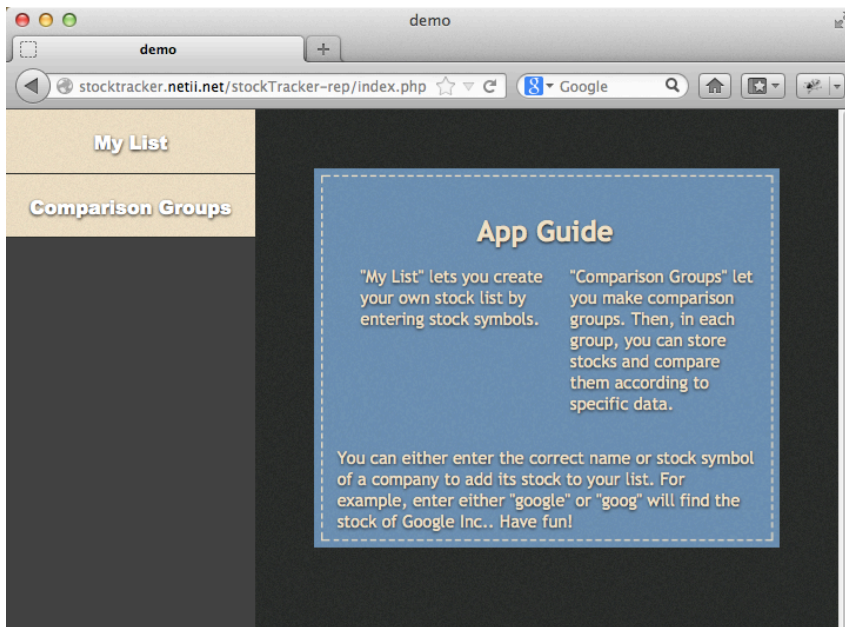


Figure 11. Relocate the main menu for the tablet version

When the width of the screen exceeds 1024px, the device is considered as laptops/desktops. There will be more space for me to show more information at one screen. But to keep following repurposing approach, I kept the same user interface components as much as possible. Hence I simply made the width of the application became 80% of the browser width, and made the design stays the same. See Figure 12.

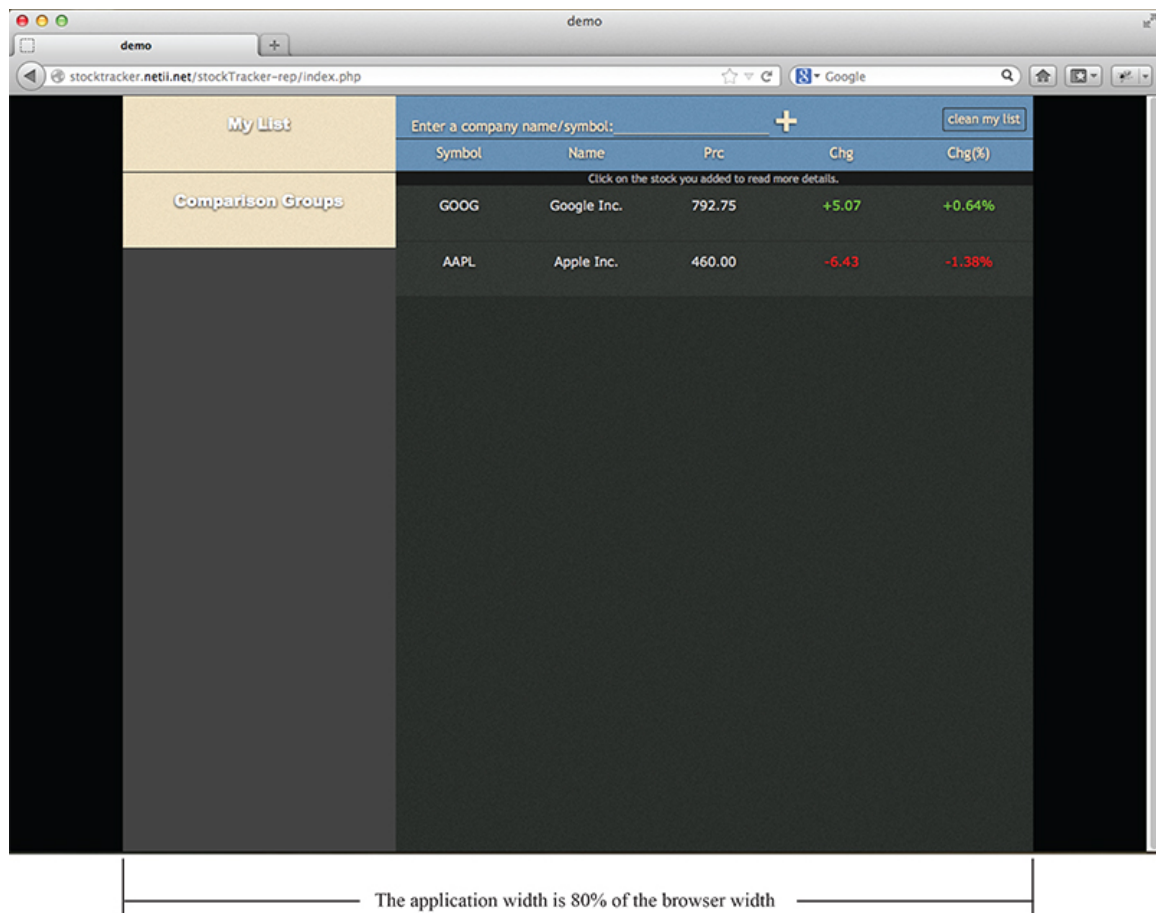


Figure-12. The application width is 80% of the browser width on the laptop or desktop version

As its definition, repurposing design keep most of the design components the same on different devices and adaptive to different devices. This design approach is easy to be

implemented.

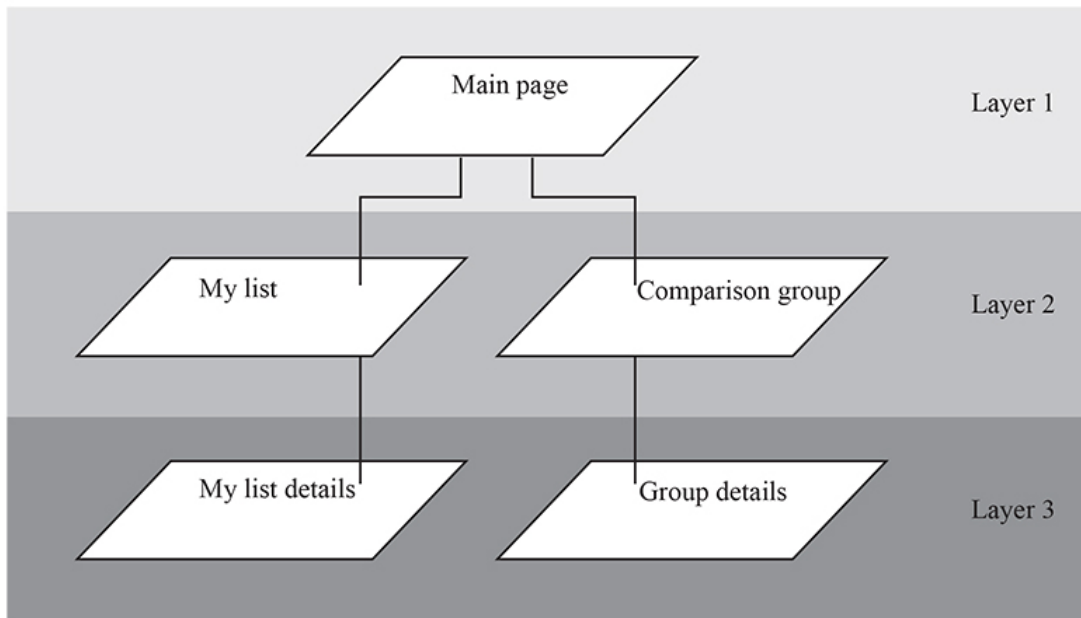
3.3 Approach 2: Optimized Design

Optimized design means “design different user interfaces for each main platform, integrating the user experience layers as tightly as possible” (Nielsen, 2012). According to the definition, I designed the application interface based on the unique features of each screen size group. The main difference between small screen devices and larger screen devices is screen space. When there is more space, the application can show more information at a time.

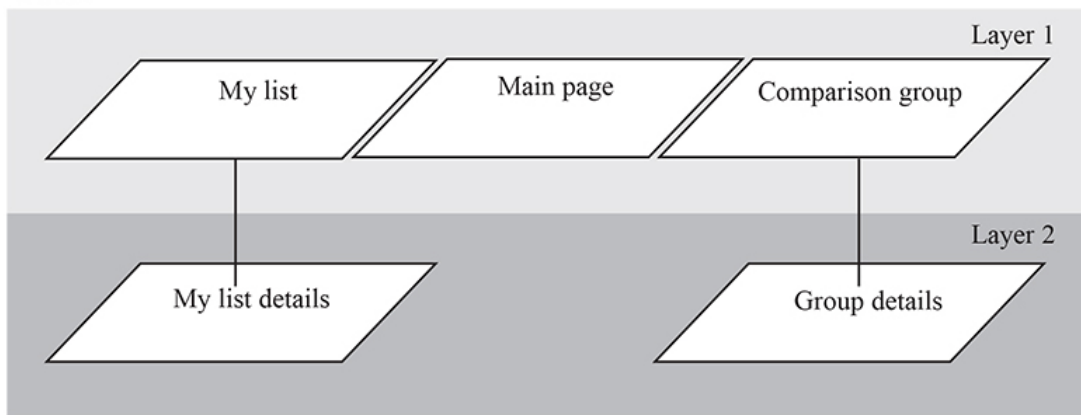
On the smartphone screen (screen width < 480px) I separated the information hierarchy into several layers. Users can access information by clicking through each layer. When the screen size become larger, I integrated the previous two layers information into one layer. Thus the information hierarchy becomes simpler, and users can get more information on each layer. Figure 13 illustrates the design thinking when I was performing optimized approach on the application.

Design Thinking of Optimized Approach

Smartphones



Tablets



Laptops/Desktops

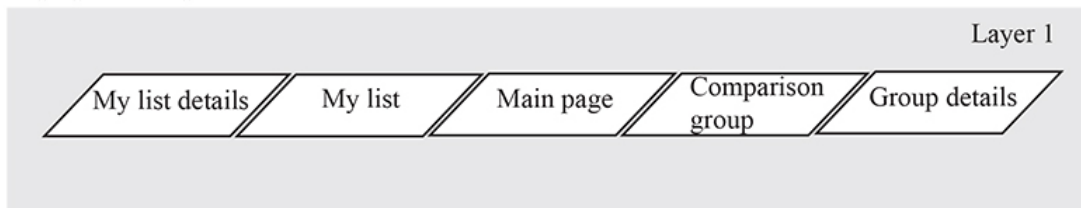


Figure 13. Design thinking of optimized approach

To participate the design thinking on tablets, when the screen width between 480px and 1024px, the application display the detailed stock information of “My List” and the detailed group information of “Comparison Group” in one screen. After users click on a stock in “My List”, the detailed information will be showed on the right side black area. The same as it is in “My List”. Thank for the bigger screens of tablets, more information can be integrated in one screen. Users can see stock list, group list and one of their subordinate information at the same time. See Figure 14, Figure 15 and Figure 16 for details.

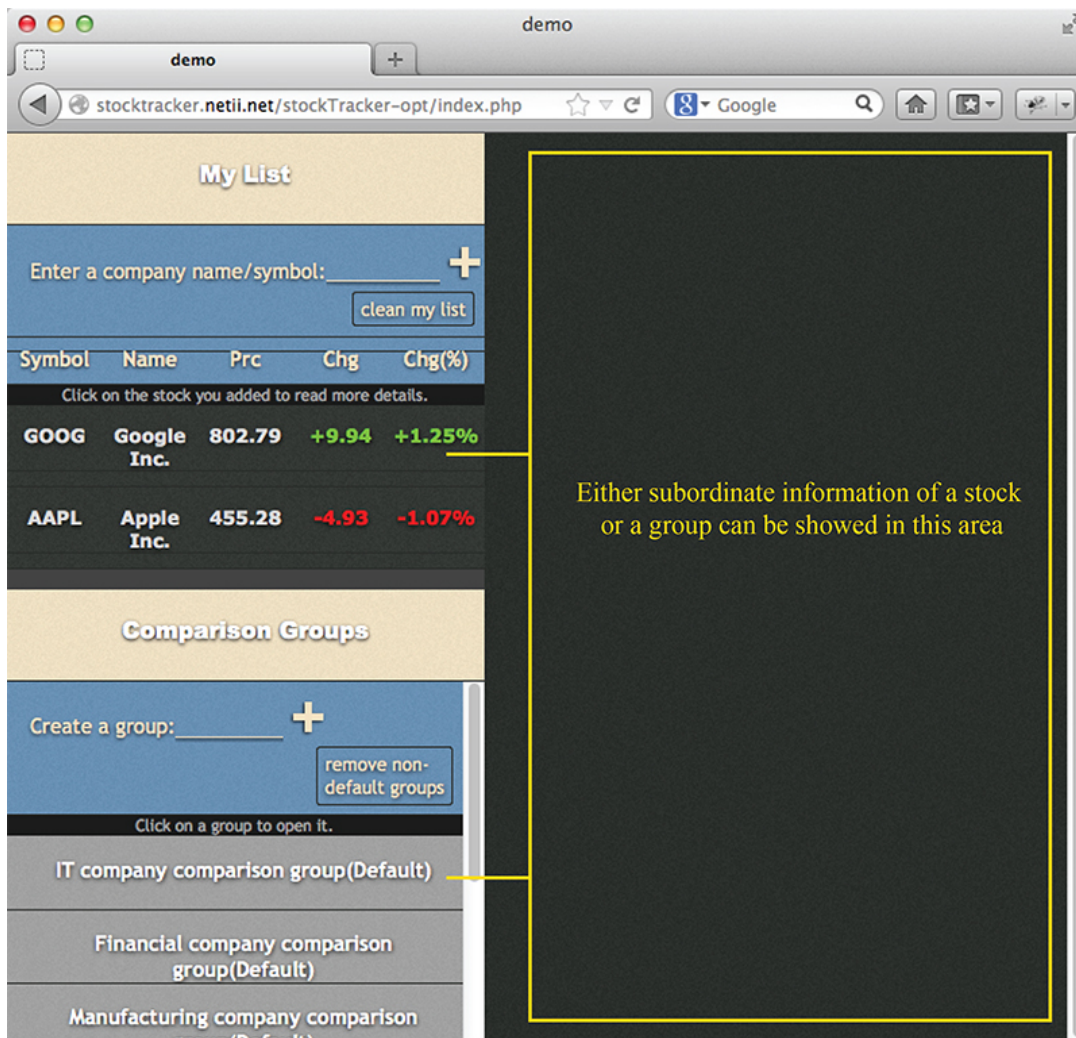


Figure 14. Optimized interface design for tablets

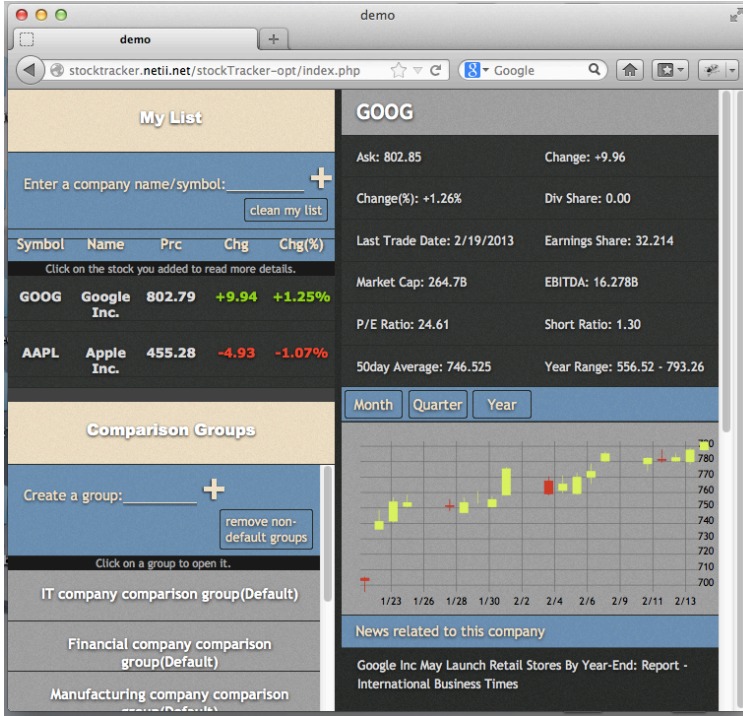


Figure 15. Subordinate information of a stock

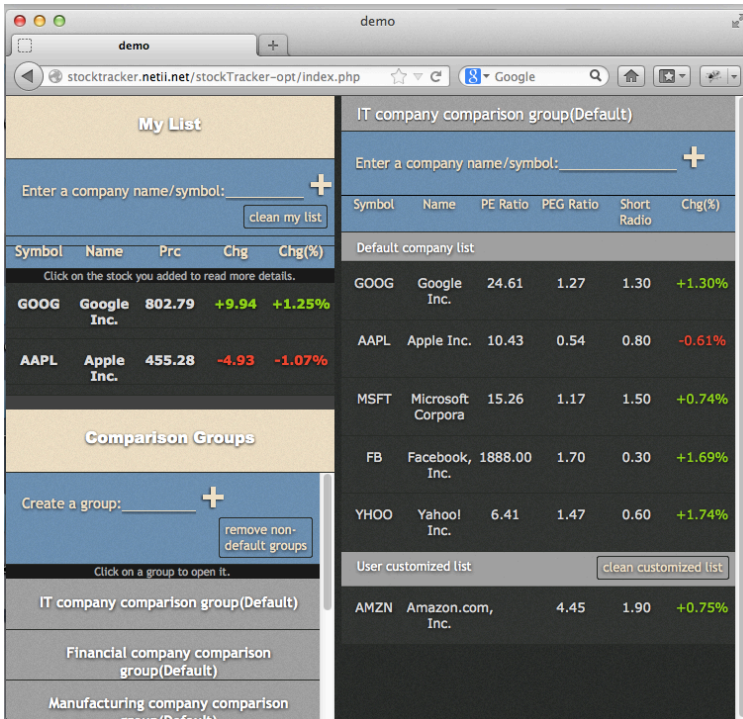


Figure 16. Subordinate information of a default group

When it comes to laptops (screen width ≥ 1024 px), screen sizes become even bigger. With a lot of space, I built all the information at one screen. “My List” section takes the left half part of the screen while “Comparison Group” section takes the right half part, and both of them show their subordinate information on the bottom half part of their own area. Users can see every thing on one page, as it is shown in Figure 17.

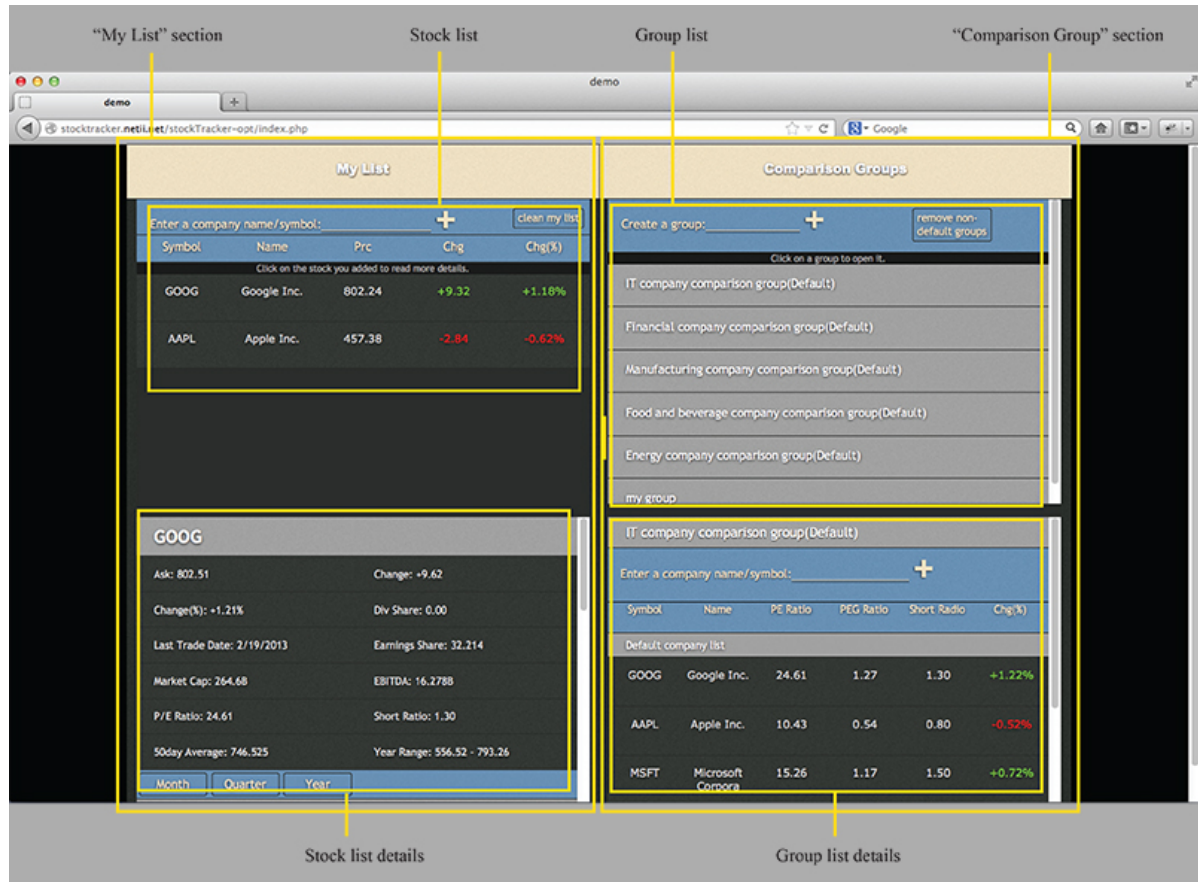


Figure 17. All information on one screen for the laptop or desktop version

By doing optimized design, I dug into the features of each device and customized the application interface for each one. It took more time to design this than design repurposing interfaces. By far, all the design work is done. In next chapter, I will introduce how I build the design into a real web application.

Chapter 4 Development

In this chapter, I will introduce the technology that I used to develop the stock tracker and the data structure of the application.

4.1 Introduction: Web Technology

The stock tracker is a web application that runs on browsers. It was purely developed by web languages and has four technical layers: *content layer*, *presentation layer*, *function layer* and *data layer*. See Figure 18.

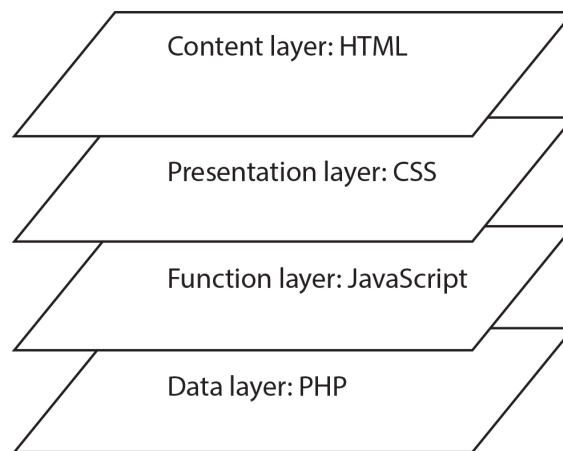


Figure 18. Four technical layers of the stock tracker

The *content layer* is the foundation of a web page. It contains all the user interface elements, including text, images and videos, and is written by hypertext markup language.

“HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.... The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages” (“HTML,” n.d.).

The *presentation layer* controls the layout and visual style of content elements and is built by cascading style sheets. “Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML” (“Cascading Style Sheets,” n.d.). For example, “Hello World” is a plain text on a web page. By using CSS designers are able to set the text with a new font family, make it red, resize it to 20px, and locate it to the center of the page.

The *function layer* is responsible for the interaction between web interfaces and users. This layer is wrote by JavaScript. “JavaScript (JS) is an interpreted computer programming language. It was originally implemented as part of web browsers so that client-side scripts could interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed” (Flanagan, 2006).

The *data layer* is used to connect the application to database and is controlled by PHP. “PHP is a server-side scripting language designed for Web development but also used as a general-purpose programming language.... PHP code is interpreted by a Web server with a PHP processor module which generates the resulting Web page” (“PHP,” n.d.).

With all of the four layers work together, the stock tracker becomes a fully functional web application. Then I will introduce the data structure and connection of the application.

4.2 Data Structure and Connection

The main idea of the stock tracker is to retrieve specific data from database according to user input. Thus database provides all the information used by this application.

I chose Yahoo! Finance as my database. Its application programming interface (API) let developers to interact with the database. In addition, Yahoo! Query Language (YQL) is a query platform developed by Yahoo! and it can help developers easily retrieve data from Yahoo! Finance API. The information retrieved from the database includes real time stock information and historical stock information.

Besides database, I used Google Feed API to filter news and get those related to a specific company. Furthermore, to make candlestick stock charts in the application, I used D3.js to dynamically bind historical stock data with interactive graphics. Figure 19 and Figure 20 illustrate the data structure and how data is connected to the user interface.

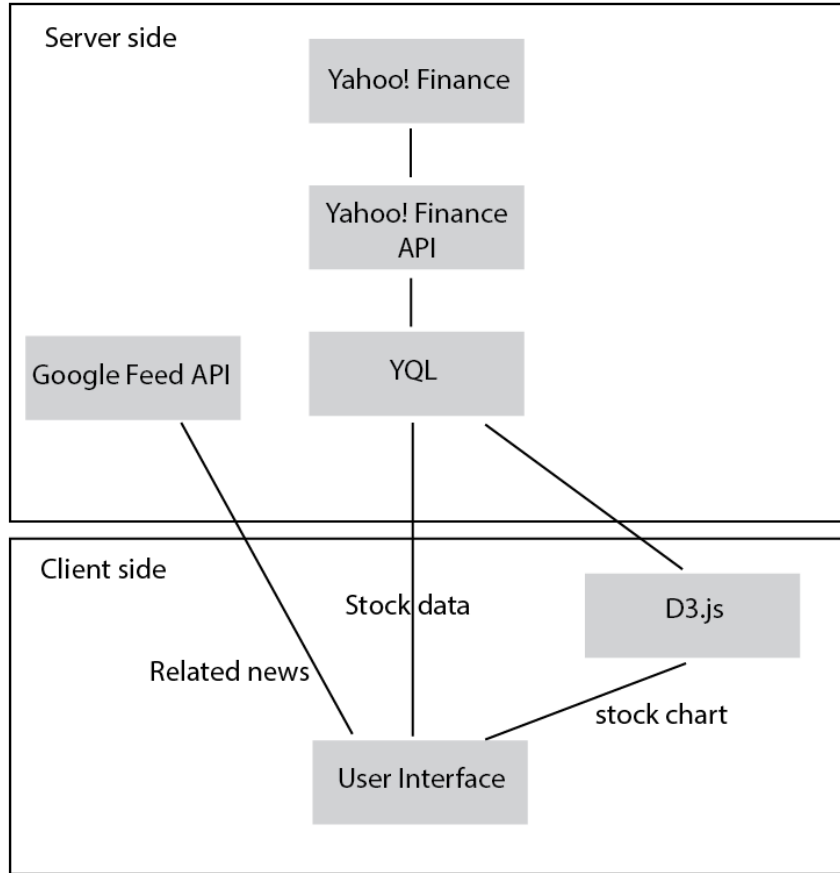


Figure 19. The data structure and connection of the application



Figure 20. The connection between application interface and database

In the next chapter, usability testing will be performed, and then I will analyze the test results and compare both design approaches.

Chapter 5 Usability Test

5.1 Objectives

In the previous chapters, I used repurposing approach and optimized approach in the design of stock tracker application. In this chapter, I performed usability test on this application, getting data and insights to analyze and compare both design approaches.

5.2 Planning

When planning the test, the first thing to consider is the number of users. According to Nielsen's (2000) study, "Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford". "His argument is that, once it is found that two or three people are totally confused by the home page, little is gained by watching more people suffer through the same flawed design" ("Usability testing," n.d.). According to Jakob Nielsen's comments and considering the time cost for this thesis I decided to find four users to test my stock tracker application. Some of my respondents are graduated with degree in investment banking and others do not have investment experience.

The test can be separated into three parts: tasks, questionnaire and discussion. For tasks, I set down twelve tasks to let respondents go through all the application functions step-by-step to perform varying missions. Below is the task list:

In "My List" section:

1. Add the stock of Google, Apple and Microsoft to the stock list.
2. Find the stock with the biggest absolute value change in the list.
3. Find the market cap and the dividend share of that stock.
4. Open the nearest quarter candlestick chart of this stock.
5. Read one of the related news of this stock.
6. Delete the stock list you just created.

In “Comparison Group” section:

7. Compare some IT stocks from default setting.
8. Figure out which company in the IT company comparison group has the highest PE ratio?
9. Add two stocks into the user customized list of the default IT company comparison group
10. Create a comparison group called “my group”.
11. In “my group”, add two stocks and find the one with the highest short ratio.
12. Delete “my group”.

Respondents are going to be asked to perform the tasks on most of the application versions, including the repurposing tablet version, the repurposing laptop version, the optimized tablet version, and the optimized laptop version.

After finishing all the tasks, they will be asked to complete questionnaires. The content for the questionnaire is based on user experiences from all of these versions. The analysis will be performed based on their answers. I numbered all the four test demos:

1. Repurposing tablet.
2. Repurposing laptop.
3. Optimized tablet.
4. Optimized laptop.

Then I will ask subjects to arrange these versions in the order of degree they prefer from best to worse according to four specific questions. Besides the four questions, there are also some other questions to get user comments on the application versions. In the end of the test, I will discuss with subjects in order to get the insights of their overall impression on the four versions of the stock tracker.

5.3 Execution

Since the design and development have been done, subjects were asked to do the tasks on the real application. I stayed with them, observing their actions, answering their questions and taking notes. At the same time, I recorded the whole process with audio and photography. Figure 21 shows the photo from a usability test.



Figure 21. The photo of a usability test

Some subjects could not come to test. So I asked them to do a remote user test via video chat software. Since it is not possible to observe their interaction with the user interface, I asked them to use think aloud protocol to do tasks. “Think-aloud protocols involve participants thinking aloud as they are performing a set of specified tasks. Users are asked to say whatever they are looking at, thinking, doing, and feeling as they go about their task. This enables observers to see first-hand the process of task completion” (“Think aloud protocol,” n.d.).

5.4 Analysis

To analyze the user test data, I set up two sections, “user experience” section and “design and develop” section, and a bunch of attributes to score the performance of each version. There are four attributes in user experience section: intuition, accuracy, efficiency and emotion. For design and development section, there are two attributes: time and difficulty. Then I made some questions asked users to arrange the demo versions according to specific attributes. For example, below are some user test questions and answers to score the versions for user experience attributes:

Please arrange these versions in the order of degree you prefer from best to worse:

1. Repurposing tablet.
2. Repurposing laptop.
3. Optimized tablet.
4. Optimized laptop.

1. Which demo version is more intuitive for you to finish the tasks? (For the “intuition” attribute.)

1>2>3>4

2. Which demo version did you make less “undo” or mistakes when performing tasks? (For the “accuracy” attribute.)

2>1>4>3

3. Which demo version did you take less time to complete the tasks than the others? (For the “efficiency” attribute.)

2>1>4>3

4. Which demo version do you like? (For the “emotion” attribute.)

1>2>3>4

The above four questions corresponds to the four attributes (intuition, accuracy, efficiency, emotion) respectively. In the answers, the first version will get 4 points and decrease to the last version which will get 1 point. In this specific test, the answer of the “accuracy” attribute is “2>1>4>3”, thus the scores for the “accuracy” attribute would be:

- 2. Repurposing laptop: 4 points,
- 1. Repurposing tablet: 3 points,
- 4. Optimized laptop: 2 points,
- 3. Optimized tablet: 1 point.

For design and develop attributes, I asked myself two questions and use the same way to score them:

Which version you use less time to design and develop than the others? (For the “time” attribute.)

Which version is easier to build? (For the “difficulty” attribute.)

After data processing and scoring, I made a chart to visualize the analysis results. I used

percentage scores for each attribute of each version. For instance, the total score of the “Intuition” attribute is 16 points (the maximum score for each test is 4 points and there are four user tests). The repurposing tablet version got 11 points in the “Intuition” attribute. 16 divided by 11 is 69%. The percentage score of the “Intuition” attribute of repurposing tablet is 69%. The same logic of percentage scores was applied to all the other attributes. See Figure 22:

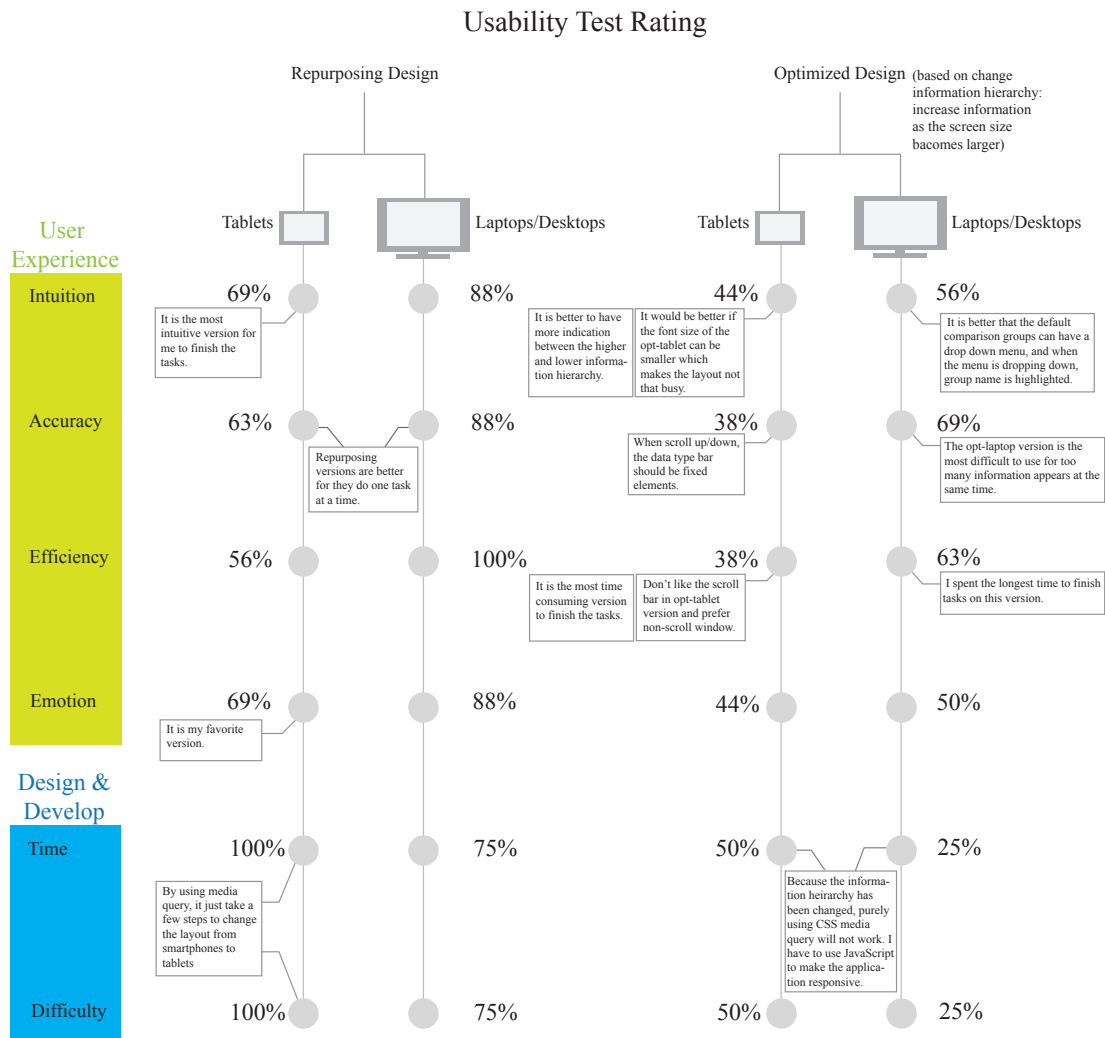


Figure 22. Chart of the usability test analysis

This chart illustrates how much scores each application version got on each attribute. By this chart people can get the big picture of how differently application versions perform in different attributes. In the next chapter, I will make the thesis conclusion based on all the previous usability test results.

Chapter 6 Conclusion

Based on the analysis in the previous chapter, the total scores of the four versions are: repurposing tablets: 49, repurposing laptops or desktops: 64, optimized tablets: 30, optimized laptops or desktops: 40. After calculated into percentage scores, they are: repurposing tablets: 68%, repurposing laptops or desktops: 89%, optimized tablets: 42%, optimized laptops or desktops: 56%.

For the user experience section, the scores of the four versions are: repurposing tablets: 41, repurposing laptops or desktops: 58, optimized tablets: 26, optimized laptops or desktops: 38. After calculated into percentage scores, they are: repurposing tablets: 64%, repurposing laptops or desktops: 91%, optimized tablets: 41%, optimized laptops or desktops: 59%.

For the design or develop section, the total scores of the four versions are: repurposing tablets: 8, repurposing laptops or desktops: 6, optimized tablets: 4, optimized laptops or

desktops: 2. After calculated into percentage scores, they are: repurposing tablets: 100%, repurposing laptops or desktops: 75%, optimized tablets: 50%, optimized laptops or desktops: 25%.

Figure 23 and Figure 24 visualize the relationship between scores, attributes and application versions:

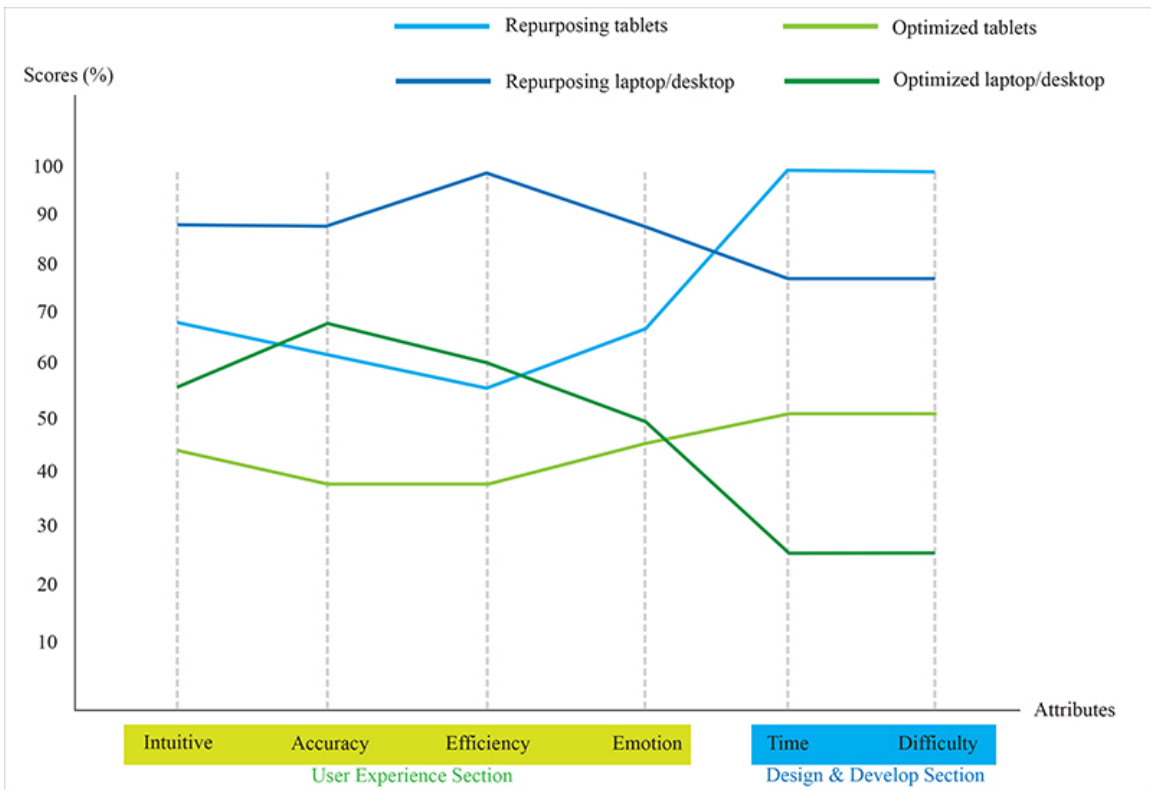


Figure 23. A chart to illustrate the scores of all attributes for the four versions

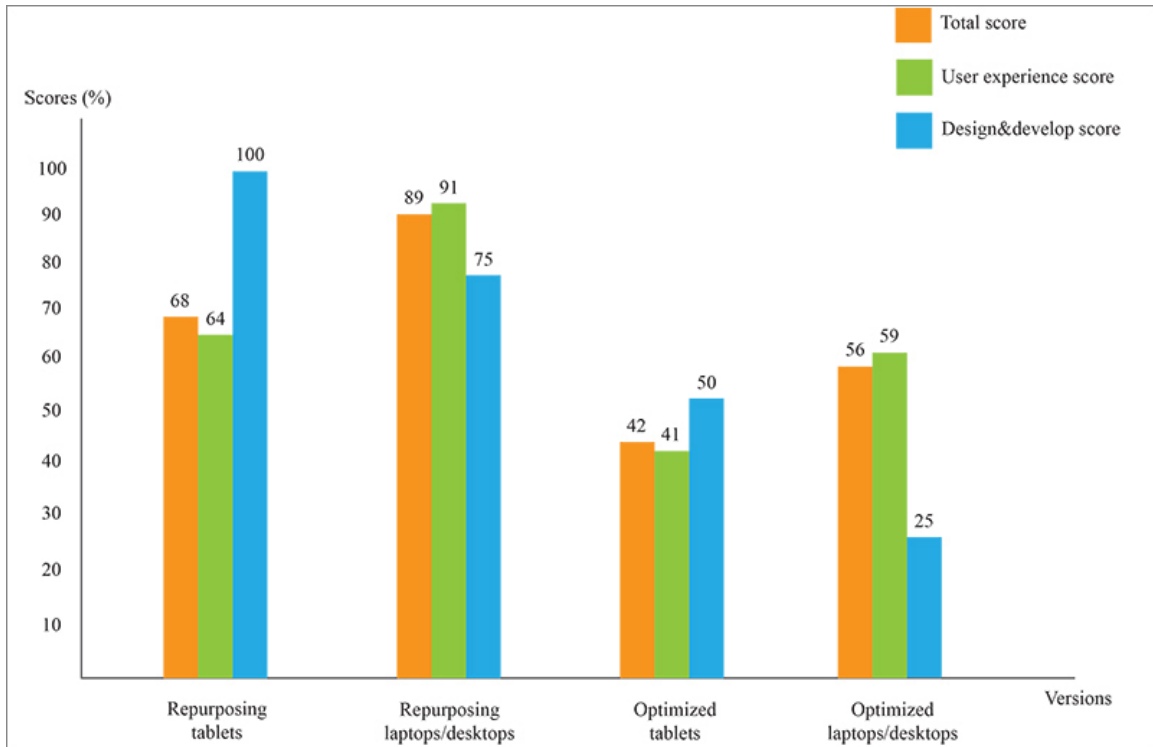


Figure 24. A chart to illustrate the total scores and section scores of the four versions

From the above charts, in most cases, no matter in total scores, user experience scores or design and develop scores, demos designed by repurposing approach got higher scores than demos designed by optimized approach.

One important thing to mention is the optimized design approach in this thesis is performed by changing the information hierarchy (display more information on one screen as the screen getting bigger), as it is shown in the Figure 13. Thus the analysis and conclusion of this thesis should all base on this specific feature. There are also some comments about the feature of changing information hierarchy in optimized versions, for examples:

- “In optimized tablet version, it is better to have more indication between the higher and lower information hierarchy.”
- “In optimized tablet version, when scroll up and down, the data type bar should be fixed elements.”
- “Don’t like the scroll bar in optimized tablet version and prefer non-scroll window.”
- “It is better that the default comparison group in optimized laptop version can have drop down menus, and when the menu is dropping down, group name is highlighted.”

The above comments indicate that after integrating more information to one screen in optimized tablet and optimized laptop or desktop versions, the application information hierarchy becomes fuzzy; as the application information getting more on tablets and laptop screens, there will be some scroll down menus to control the overflowed information, but users do not like to use many scroll down menus to navigate information; more information in one screen can make users feel the information hierarchy is busy and hard to use. On the other hand, repurposing versions are liked by most of the respondents for their information hierarchy is simple and users can do one task on a specific screen but not confused by several tasks on one screen. Although in large screens the simple information hierarchy of repurposing versions can be a waste of screen space, respondents think it is not important comparing to its advantages. Thus performing optimized design by adding more information while screen size becomes larger is risky, because normally it would be more time-consuming and more difficult to design and

develop the optimized versions, and the optimized versions are very likely to provide poorer user experience comparing to repurposing versions in the context of lightweight web application design.

According to all the previous works, repurposing approach is better than optimized approach in lightweight application design if optimized approach is performed by integrate more information to one screen while the screen size become larger. There might be a bias on the scores of design and develop attributes because I was the only respondent. Additional study with a larger designer and developer respondent group would resolve this issue. It is also important to note that the study is based on an application that demands a fluid environment. My study tests a stock tracker to work across multiple devices for the same user. So the scores of “intuition” and “efficiency” attributes will drop on optimized versions because of a learning curve: users need to rethink the functions of each optimized user interface of the specific device. On the other side, for repurposing versions, the user interfaces change very little and are likely to get higher scores in the “intuition” and “efficiency” attributes. For other types of applications with different user needs, the testing methods and results may be different.

Hope the analysis and conclusion of my thesis can be a helpful reference for people who plan to design cross-device web applications.

Bibliography

- Corral, L., Sillitti, A., Succi, G., Garibbo, A., & Ramella, P. (2012). Evolution of mobile software development from platform-specific to web-based multiplatform paradigm. *Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software*, 181-183. Retrieved from <http://dl.acm.org/citation.cfm?id=2048237&picked=prox&cfid=300128882&cftoken=78586268>.
- Charland, A., & LeRoux, B. (2011). Mobile application development: web vs. native. *Queue*, 9 (4), 20. doi: 10.1145/1966989.1968203.
- Cascading Style Sheets. (n.d.). In *Wikipedia*. Retrieved from http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- Fat Finger. (n.d.). In *Urban Dictionary*. Retrieved from <http://www.urbandictionary.com/define.php?term=fat+finger>
- Flanagan, D. (2006). *JavaScript: The Definitive Guide* (5th ed.). O'Reilly & Associates. ISBN 0-596-10199-6.
- HTML. (n.d.). In *Wikipedia*. Retrieved from <http://en.wikipedia.org/wiki/HTML>
- Media Queries. (n.d.). In *Wikipedia*. Retrieved from http://en.wikipedia.org/wiki/Media_queries
- Nielsen, J. (2000, March 19). Why you only need to test with 5 users. Retrieved from <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>

Nielsen, J. (2012, May 21). Repurposing vs. optimized design. Retrieved from

<http://www.nngroup.com/articles/repurposing-vs-optimized-design/>

PHP. (n.d.). In *Wikipedia*. Retrieved from

<http://en.wikipedia.org/wiki/Php>

Shubin, H., & Meehan, M. M. (1997). Navigation in web applications. *Interactions*, 4 (6), 13-17. doi:10.1145/267505.267508.

Smus, B. (2012). A non-responsive approach to building cross-device webapps.

Retrieved from <http://www.html5rocks.com/en/mobile/cross-device/>

Think aloud protocol. (n.d.). In *Wikipedia*. Retrieved from

http://en.wikipedia.org/wiki/Think_aloud_protocol

User-Centered Design. (n.d.). In *Wikipedia*. Retrieved from

http://en.wikipedia.org/wiki/User-centered_design

Usability testing. (n.d.). In *Wikipedia*. Retrieved from

http://en.wikipedia.org/wiki/Usability_testing#cite_note-useit-10

World Wide Web Consortium. (2012, June 19). Media queries. In *W3C Recommendation*.

Retrieved from <http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>